INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Quantitative Kinetic Analyses of Adsorption and Desorption Processes at the Liquid - Solid Interface with Surface Plasmon Resonance

by

Linda Suna Jung

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Washington

1999

Program Authorized to Offer Degree: Chemistry

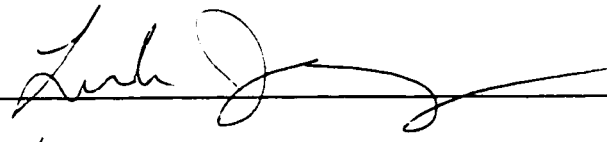
UMI Number: 9952841

UMI Microform 9952841

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Doctoral Dissertation

In presenting this thesis in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to UMI Dissertation Services, 300 North Zeeb Road, P.O. Box 1346, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature 

Date 11-23-99

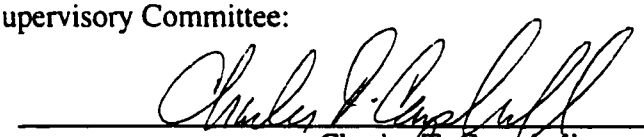
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Linda Suna Jung

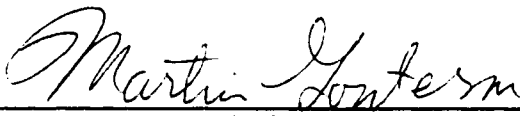
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

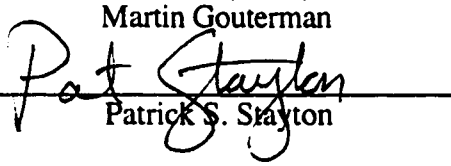


Charles T. Campbell

Reading Committee:



Martin Gouterman



Patrick S. Stayton

Date: 11-23-99

University of Washington

Abstract

Quantitative Kinetic Analyses of Adsorption and Desorption Processes at the Liquid - Solid Interface with Surface Plasmon Resonance

by Linda S. Jung

Chairperson of the Supervisory Committee: Professor Charles T. Campbell
Department of Chemistry

Surface Plasmon Resonance (SPR) spectroscopy is one of the few techniques that can directly detect adsorption onto a surface immersed in liquid with high time resolution and sensitivity to <5% of a monolayer. By modifying the sensing surface with a proper terminal receptor group, SPR provides a surface analytical tool for studying the kinetics of surface adsorption and desorption reactions in situ. Methods for extracting quantitative information as well as kinetic parameters from such experiments are presented here. First, a simple but quantitative mathematical formalism for interpretation of SPR signals from adsorbed films is presented. This provides a way of estimating adsorbed film thicknesses, surface coverages or surface concentrations from the SPR response over the entire range of film thicknesses without relying on calibration curves of response versus known thicknesses or surface concentrations. Next, a method for estimating sticking probabilities (rate per collision) and intrinsic rate constants for adsorption from liquid solutions is described, where a simple numerical solution to Fick's Law is used to calculate the adsorbate concentrations at incremental distances from the surface as adsorption occurs. The resulting concentration nearest the surface is needed to calculate both the intrinsic rate constant for adsorption and the collision frequency of the adsorbate with the surface versus time or coverage. These methods are applied to alkylthiol adsorption on the Au SPR surface and streptavidin (SA) adsorption on mixed

biotin-containing alkylthiolate monolayers. The sticking probability of alkylthiols on gold shows that its transition state is stabilized by ~0.7 kJ/mol per CH_2 group, or about half of that of the product thiolate. The kinetics of the competitive desorption of SA and SA mutants from the biotin-containing monolayers reveal the nature of the SA monolayer. Finally, a method for quantifying binding of proteins and other ligands to the surfaces of phospholipid bilayered vesicles (attached intact to the SPR surface via the SA layer) is demonstrated, and applied to determine the dissociation equilibrium constant for phospholipase A_2 binding to membranes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BAT. Biotin-terminated alkylthiol, $HS(CH_2)_{15}$-biotin

Biotin-DOPE. 1,2-Dioleoyl-*sn*-glycero-3-phosphoethanolamine with a Biotin$CONH(CH_2)_6CO$ group amide linked to the ethanolamine amino group

BSA. Bovine serum albumin

BTM. Biotin-containing alkylthiolate monolayer

$\mathbf{C_{bulk}}$. Bulk solution concentration

$\mathbf{C_s}$. Solution in nearest contact to the surface

$\mathbf{C_2}$. Ethanethiol

$\mathbf{C_6}$. Hexanethiol

$\mathbf{C_7}$. Heptanethiol

$\mathbf{C_8}$. Octanethiol

$\mathbf{C_{12}}$. Dodecanethiol

$\mathbf{C_{16}}$. Hexadecanethiol

$\mathbf{C_{18}}$. Octadecanethiol

$\mathbf{C_{15}COOH}$. $HS(CH_2)_{15}COOH$

$\mathbf{C_{11}PEO}$. Poly(ethylene oxide) terminated alkylthiol, $HS(CH_2)_{11}(OCH_2)_4OH$

D. Diffusion constant

$\mathbf{d_{diff}}$. Characteristic diffusion distance

$\mathbf{DO_{et}PC}$. 1,2-Dioleoyl-*sn*-glycero-3-phosphocholine

HBM. Hybrid bilayer membrane

$\mathbf{J_s}$. Collision frequency with the surface

MHD. Hexadecanethiol

PBS. Phosphate buffered saline

PEO. Poly(ethylene oxide) terminated alkylthiol, $HS(CH_2)_{11}(OCH_2)_4OH$

PLA_2. Phospholipase A_2

RI. Refractive index

RIU. Refractive index unit

S. Sticking probability

S_0. Initial sticking probability

SA. Streptavidin

SPR. Surface plasmon resonance

SPW. Surface plasmon wave

τ_{diff}. Characteristic diffusion time

$t_{1/2}$. Half-life of adsorption

WT. Wild-type

X_{BAT}. Surface biotin concentration

ACKNOWLEDGMENTS

The author wishes to acknowledge and thank C. T. Campbell for all his guidance and help throughout the years (and especially in these last few months). I also wish to acknowledge the following people who also made significant contributions to the work presented in this dissertation: J. S. Shumaker-Parry, M.H. Gelb, S. S. Yee, M.N. Mar, C.C. Jung, K.S. Johnston, T.M. Chinowsky, K.E. Nelson, P.S. Stayton, D.G. Castner, L. Gamble, S.C. Parker, J.M. Schurr, M. Boeckl, T. Sasaki, E. Naeemi, B.D. Ratner, W.A. Ciridon, J. W. Rogers, J. Ruzicka, D. Holman, B.P. Holm, J. Heutink and N. Ward.

I wish to thank the Campbell group members (past and present) for helpful discussions and support, J. S. Shumaker-Parry, H. Lu and Mack Carter for assistance, editing and helpful advice. In addition, I would like to thank M.J. Sailor (and the Sailor group), J.D. Simon, D.A. Roise without whom I would have never made it to graduate school, B.H. Robinson and J.M. Schurr whose help and encouragement helped me survive the required coursework.

DEDICATION

The author wishes to dedicate this dissertation to Nathan A. Frei (1972-1996), classmate, co-worker and friend.

"I don't want the world, I just want your half" - They Might be Giants

CHAPTER 1: INTRODUCTION

INTRODUCTION

Surface science has made much progress in the last 30 years at clarifying the structure of adsorbates and the mechanisms of chemical reactions occurring on the surfaces of solids in ultrahigh vacuum or low-pressure gas environments. By comparison, very little is known about reactions occurring at the liquid / solid interface. However, this interface is very important in many technologically-important processes in chemical- and bio- sensors and in biology. In this work, an attempt is made to start addressing the surface chemistry of liquid / solid interfaces. Specifically, a technique which is capable of quantitatively analyzing adsorption amounts, surface plasmon resonance (SPR), is extended in its capabilities and ease of use. It is applied to determine for the first time the sticking probabilities of species on a solid surface from a liquid solution. Sticking probability is the most fundamental way to express an adsorption rate constant and has already been very powerful in clarifying the mechanisms and dynamics of processes at the gas / solid interface. The sticking probabilities, measured here for alkylthiols from ethanolic solution onto gold, provide unique insight into the transition state for the primary adsorption step which forms this important class of self-assembled monolayers (SAMs). By incorporating poly(ethylene glycol) and biotin head groups into a mixed alkylthiolate SAM, a surface is prepared which is used to prepare a dense and selective monolayer of the protein streptavidin (SA). Furthermore, by using genetically-engineered mutants of SA, the nature of the SA-biotin bonding which links the SA monolayer to the surface has been clarified. This was done by comparing the adsorption and desorption kinetics of several SA mutants which have a wide range of SA-biotin binding strengths, and doing this as a detailed function of the mole fraction of biotin in the primary thiolate monolayer. This SA monolayer provides a biolayer with a high

density of biotin sites for further attachment of bioreceptors. It is used here to attach intact phospholipid vesicles to the surface, so that the on-rates, off-rates and equilibrium binding constants of protein binding to membrane surfaces can be easily studied. This is demonstrated here by measuring the equilibrium constant for phospholipase binding to the surfaces of these immobilized vesicles.

SURFACE PLASMON RESONANCE

Surface Plasmon Resonance (SPR) devices consist of a transparent optical substrate coated with a thin (~50 nm) metal film. (The optimum thickness of the metal film is a function of the index refraction of the glass (prism or fiber), the index refraction of the sample and the wavelength of the incident light.) In order to excite a strong SPR, the dielectric function of the metal should have a large negative real part at the wavelength of the light used, which for free electron-like metals such as silver and gold occurs in the visible region [1]. The metal film must be thin enough for the evanescent field of the incident light to penetrate it and excite a surface plasmon wave (SPW) at the interface. The evanescent wave decays exponentially away from the interface (and along the interface if the media is adsorbing) [2]. This decay length determines the sensing depth and is on the order of 0.2 to 0.3 of the wavelength of light [1]. This allows SPR sensors to be applied to the detection of adsorption onto the sensor surface from liquid solutions [1,3-16].

The specularly reflected light intensity is measured as the light passes through the substrate and reflects from the interface between the substrate and the analyte medium. When the wave vector of incident light matches the wave vector of the SPW, the incident light is adsorbed by the SPW and not reflected [17]. Essentially, part of the incident energy will couple into a SPW traveling along the interface between the gold layer and the solution to be analyzed. This loss of this energy is observed as a sharp

attenuation of reflectivity, known as the surface plasmon resonance effect. This occurs at a specific wavelength and angle of incident light as shown in Fig. 1.1. This angle and wavelength vary extremely sensitively with the refractive index (RI) (or complex dielectric constant) of the medium in contact with the metal surface of the SPR sensor. The are two basic methods to monitor this, by fixed angle or fixed wavelength scans. With fixed wavelength, collimated TM (transverse magnetic) polarized light (perpendicular to the electric field) is incident at fixed wavelength on the base of the prism. The prism is rotated until the resonance condition is satisfied at some unique angle. With the fixed angle method, the angle of incidence is held constant and the resonance condition occurs at some unique wavelength as monitored by a wavelength-dispersive device such as a spectrometer [2]. Changes in the dielectric properties of the sample within some "probe depth" of the interface (<300 nm) will therefore give rise to changes in the conditions necessary for SPR thus forming the basis for SPR based chemical sensing.

Improved sensitivity of these measurements over other optical measurements is due to an increased optical path length gained by efficiently coupling the light into the film along the metal surface, supporting the propagation of a guided optical mode. For the SPR systems used here, the light guiding is achieved with prism coupling (as in the planar system) or end fire launching (as in the fiber-optic probe) [18]. To use the evanescent field efficiently, the SPR sensing surface can be functionalized with the appropriate organofunctional thiolate, which, in chemical sensor applications, will "concentrate" the desired analyte to the sensing surface, improving the signal-to-noise ratio and sensitivity of the sensor to that analyte with minimal adverse effects [19,20]. The improved sensitivity allows SPR to detect the adsorption of sub-nanometer thick films [18]. If the sensor surface is functionalized to contain selective receptors for specific chemicals or biomolecules, selective chemical and biochemical sensors can be constructed. SPR spectroscopy also has many potential applications in fundamental surface science since it is one of the few techniques that can directly detect adsorption

onto a surface immersed in liquid with high time resolution and submonolayer sensitivity. When properly coupled to a flow system of fast time resolution, the system provides a unique opportunity for following the kinetics of adsorption processes from liquid solutions in detail.

THIOLS ON AU

The use of organofunctional thiolates for functionalizing these SPR gold surfaces is a simple process which has been studied extensively. The reaction is generally given as follows:

$$Au^0 + R(CH_2)_nSH \rightarrow R(CH_2)_n\,SAu + {}^1/_2\,H_2.\ [21].$$

Disulfides have also been used. The reaction mechanism is described as follows:

$$2\,Au^0 + R(CH_2)_n\,S\text{-}S(CH_2)_n\,R \rightarrow 2(R(CH_2)_n\,SAu)\ [22],$$

with a proposed intermediate step where the disulfide is initially bound to one Au before the disulfide bond cleaves. In either case, the end result is a thiolate bound to the Au surface. The adsorption is considered a multi-step process. A sparse monolayer is initially adsorbed very quickly due to the strong surface-head group reaction. A close - packed monolayer is not formed until much later [23] due to a very slow crystallization process. This process is associated with motion and incremental adsorption [21] related to chain disorder (defects), different components of the chain-chain interaction (Van der Waals, dipole-dipole, etc.) and the mobility of the chains [24,25].

Diffraction studies on Au(111) show the monolayer forms a $(\sqrt{3}x\ \sqrt{3})R30^\circ$ hexagonal overlayer [24], where the symmetry of the sulfurs is hexagonal with 4.99Å spacing (which is 3 times the Van der Waals diameter of sulfur, 1.85 Å), giving 21.7 $Å^2$

per molecule or ~4.16 x10^{14} chains/cm^2. This distance is greater than the distance of closest approach between alkyl chains (4.24 Å) so the chains must tilt to maximize the Van der Waals interaction (1.4 - 1.8 kcal/mol). The thickness of the alkyl groups immobilized on the surface is estimated to be between 1.16-1.5 Å per CH_2 group (depending on the terminal group) [21]. The choice of the terminal group obviously depends on the application. The spacing of the S atoms dictates that the terminal group should be ~5 Å or less to efficiently pack. However, larger bulky terminal groups may be used if it is "diluted" in a mixture with shorter chains [21]. Such mixtures typically segregate in 2-6 nm domains [24].

STREPTAVIDIN ON BIOTINYLATED AU SURFACES

A biotin functionalized surface can be formed from a binary mixture of a biotin-terminated alkylthiol (BAT) and a poly(ethylene oxide)-terminated alkylthiol (PEO, which serves to "dilute" the biotin groups and provides a background which resists non-specific binding). It has been previously demonstrated that streptavidin (SA) may be immobilized onto such biotin-containing alkylthiolate monolayers with high coverage, specificity, and activity [26-30] (and is shown schematically in Fig. 1.2). SA binds to these surfaces via either one or two such biotin linkages. The stability of the SA within the adlayer would be higher and show improved orientation if two biotin linkages are formed. Desorption (i.e., cleavage of the surface biotin-SA bond) provides a way to measure this stability.

This streptavidin-biotin couple is significant due to its extremely high binding affinity, K_a, which is ~10^{13} M [31]. SA has four equivalent sites for biotins and binds to these biotinylated surfaces with two of its binding sites away from the surface. Secondary molecules chemically modified with biotin can be rapidly and conveniently

immobilized on two remaining binding sites of the immobilized SA molecules with minimal impact on the biological activity (e.g. specificity). Therefore, once the SA layer is successfully immobilized, it can then be used to study more complex biologically relevant binding reactions such as further protein adsorption to ligands attached at these biotin sites.

OVERVIEW

Presented here is a study aimed at understanding fundamental issues of adlayer formation on gold surfaces and the use of these resulting films using SPR. The experimental details are presented in Chapter 2. First, A formalism for quantifying SPR adsorption measurements had to be developed, providing a way of accurately estimating adsorbed film thicknesses, surface coverages or surface concentrations without the necessity of calibration curves of response versus known thicknesses or surface concentrations. This formalism is demonstrated in Chapter 3. Second, a method for determining the sticking probabilities and intrinsic rate constants from adsorption in liquids from the SPR measured adsorption curves is presented in Chapter 4. Here, a complete analysis of the kinetics of alkylthiol adsorption on Au surfaces from ethanol solution is undertaken. Next, in Chapters 5 and 6, the character of the SA adlayers on biotinylated Au surfaces are assessed by studying the kinetics of adsorption and competitive desorption of SA to / from the biotinylated Au surface as a function of the surface composition of the mixed BAT / PEO layer. By comparing the kinetics of a variety of genetically-engineered mutant SAs, a fairly complete understanding of the binding and desorption mechanism is developed. Finally, an SPR-based method for quantifying high affinity binding of proteins to the surfaces of phospholipid bilayered vesicles which have been tethered onto the SA layer (by incorporating 0.3 % biotinylated lipids into the vesicles during their formation) is demonstrated in Chapter 7.

NOTES TO INTRODUCTION

[1] Liedberg, B.; Nylander, C.; Lundstrom, I. *Biosensors & Bioelectronics* **1995**, *10*, i-ix.

[2] Jung, C. C. *Surface Plasmon Resonace Light Modulators Using Electro-optic Polymers*; University of Washington, 1994.

[3] Jordan, C. E.; Frey, B. L.; Kornguth, S.; Corn, R. M. *Langmuir* **1994**, 3642-3648.

[4] Lundstrom, I. *Biosens. Bioelectron.* **1994**, *9*, 725.

[5] Lukosz, W. *Biosensors & Bioelectronics* **1997**, *12*, 175-184.

[6] Lukosz, W. *Biosensors & Bioelectronics* **1991**, *6*, 215-225.

[7] Melendez, J.; Carr, R.; Bartholomew, D.; Taneja, H. *Development of a Surface Plasmon Resonance Sensor for Commerical Applications*: Washington, D. C., 1996.

[8] Melendez, J.; Carr, R.; Bartholomew, D. U.; Kukanskis, K.; Elkind, J.; Yee, S.; Furlong, C.; Woodbury, R. *Sens. Acutators, B* **1996**, *35*, 212.

[9] Kooyman, R. P. H.; Kiolkman, H.; Gent, J. V.; Greve, J. *Analytica Chimica Acta* **1988**, 35-45.

[10] Stenberg, E.; Persson, B.; Roos, H.; Urbaniczky, C. *Journal of Colloid and Interface Science* **1990**, *142*, 513-526.

[11] O'Shannessy, D. J. *Curr. Opin. Biotechnol.* **1994**, *6*, 65.

[12] Liedberg, B.; Nylander, C.; Lundstrom, I. *Sensors and Actuators* **1983**, *4*, 299-304.

[13] Mrksich, M.; Sigal, G. B.; Whitesides, G. M. *Langmuir* **1995**, *11*, 4383-4385.

[14] Mrksich, M.; Grunwell, J. R.; Whitesides, G. M. *J. Am. Chem. Soc* **1995**, *117*, 12009-12010.

[15] Kunz, U.; Katerkamp, A.; Renneberg, R.; Spener, F.; Cammann, K. *Sensors and Actuators B* **1996**, *32*, 149-155.

[16] Peterlinz, K. A.; Georgiadis, R. *Langmuir* **1996**, *12*, 4731-4740.

[17] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[18] Yeatman, E. M. *Biosensors & Bioelectronics* **1996**, *11*, 635-649.

[19] Swalen, J. D. *Annu. Rev. Mater. Sci* **1991**, *21*, 373-408.

[20] Lofas, S.; Malmqvist, M.; Ronnberg, I.; Stenberg, E.; Liedberg, B.; Lunstrom, I. *Sensors and Actuators B* **1991**, *5*, 79-84.

[21] Ulman, A. *Chem. Rev.* **1996**, *96*, 1533-1554.

[22] Dubois, L. H.; Nuzzo, R. G. *Annu. Rev. Phys. Chem* **1992**, *43*, 437-63.

[23] Biebuyck, H. A.; Bain, C. D.; Whitsides, G. M. *Langmuir* **1994**, *10*, 1825-1831.

[24] Nuzzo, R. G. *J. Am. Chem. Soc.* **1987**, *109*, 2358-2368.

[25] Allara, D. L. *Biosensors & Bioelectronics* **1995**, *10*, 771-783.

[26] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

[27] Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

[28] Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

[29] Perez-Luna, V. H., O'Brien, M. J., Opperman, K. A., Hampton, P. D., Stayton, P. S., Klumb, L, Lopez, G. P. *J. Am. Chem. Soc* **1999**, *121*, 6469-6478.

[30] Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

[31] Green, N. M. *Adv. Protein Chem.* **1975**, *29*, 85-133.

FIGURES FOR INTRODUCTION

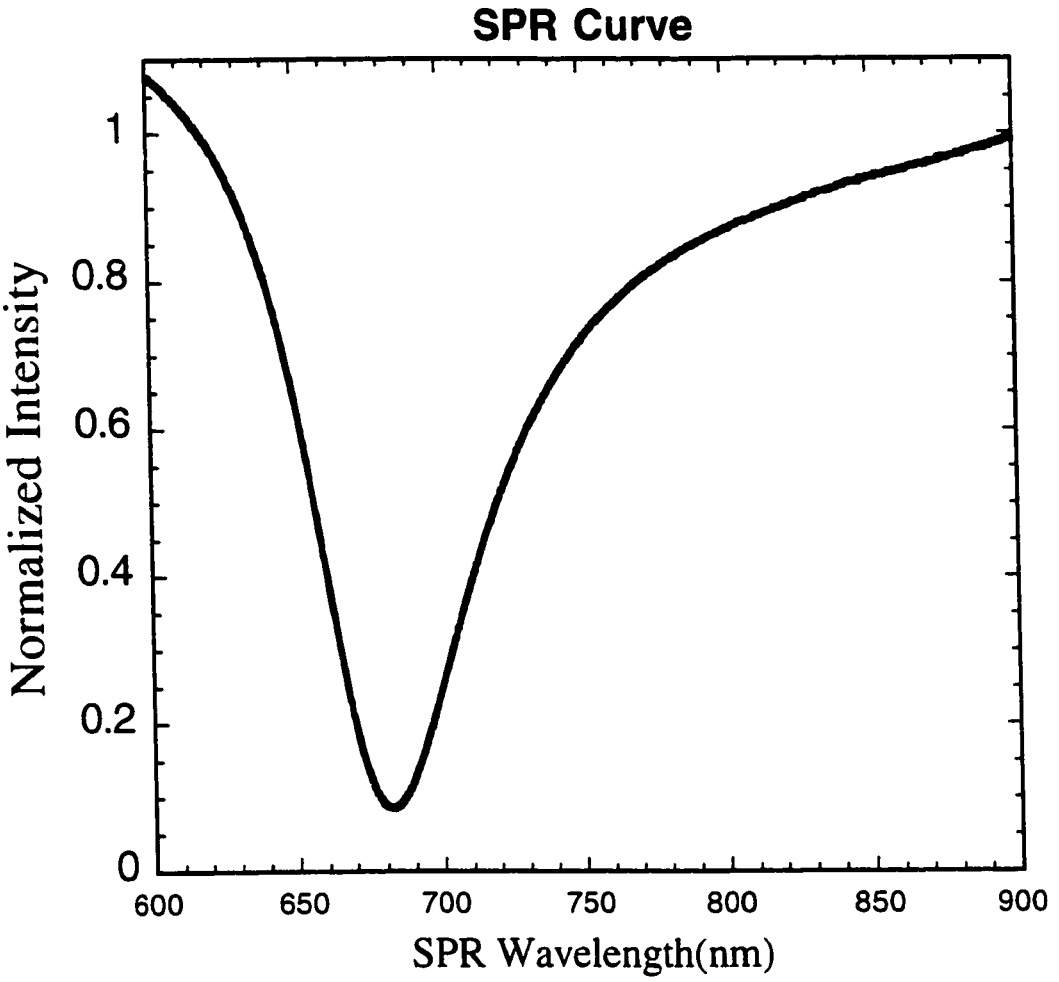


Figure 1.1 An example of an SPR curve

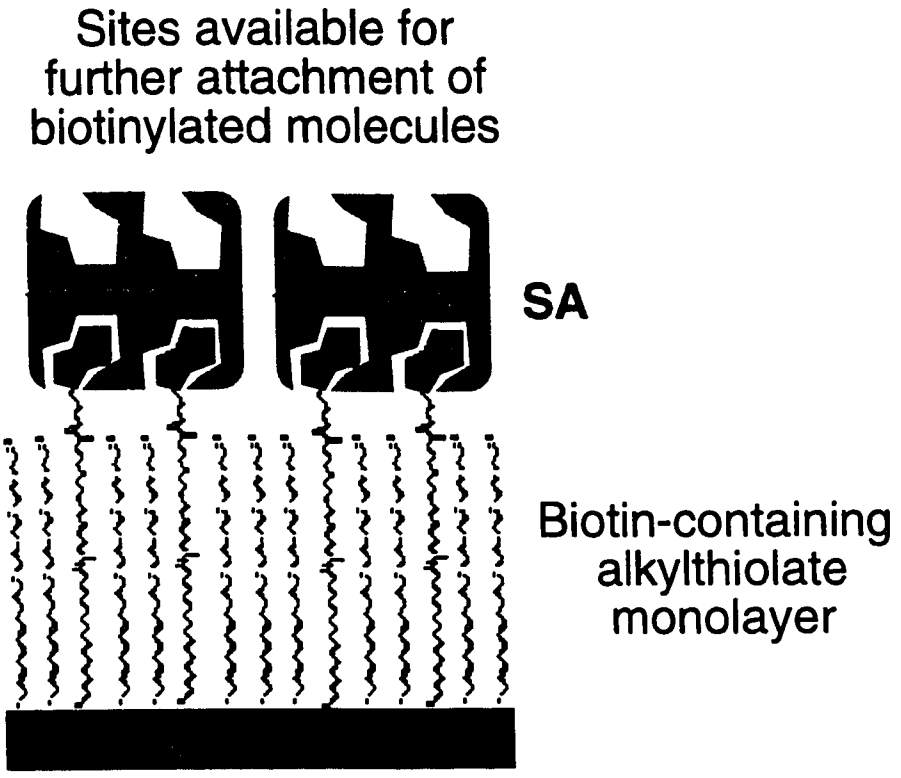


Figure 1.2 Idealized depiction of streptavidin adsorbed onto a biotin-containing alkylthiolate monolayer.

CHAPTER 2: EXPERIMENTAL

MATERIALS

Octadecanethiol, hexadecanethiol, dodecanethiol, decanethiol, octanethiol, heptanethiol, hexanethiol, ethanethiol were purchased from Aldrich and used without any further purification. Biotinylated alkylthiol (BAT), $HS(CH_2)_{15}BAT$, and poly(ethylene oxide) alkylthiol (PEO), $HS(CH_2)_{11}(OCH_2)_4OH$, were synthesized as described [1] and provided by Profs. Tomikazu Sasaki and Buddy D. Ratner, respectively (Univ. of Washington). $HS(CH_2)_{15}COOH$ was also synthesized by the Prof. Tomikazu Sasaki. . A refractive index of 1.48 was used for $C_{11}PEO$, which we determined by measuring its incremental change in the index of refraction when added to ethanol, as described in [2]. A value of 1.46 was used for all the other thiols, based on [2]. (Note all refractive indices are relative to air for Na light (589 nm))

The streptavidin and streptavidin mutants used in Chapters 5 and 6 were expressed and purified by Prof. Patrick S. Stayton as previously described [3,4]. The streptavidin solutions used in these experiments were ~0.05 mg / ml in phosphate buffered saline (PBS, 0.1 M, pH 7.4) unless otherwise specified. The known index of refraction for this class of proteins of 1.57 [2].

In Chapter 7, the 1,2-Dioleoyl-*sn*-glycero-3-phosphocholine ($DO_{et}PC$) was prepared as described [5] and provided by Prof. Michael H. Gelb (Univ. of Washington). 1,2-Dioleoyl-*sn*-glycero-3-phosphoethanolamine with a $BiotinCONH(CH_2)_6CO$ group amide linked to the ethanolamine amino group (Biotin-DOPE) was purchased from Avanti Polar Lipids Inc. PLA_2 from *Naja naja* venom was purchased from Sigma (cat. # P6139). Ellman's Reagent (5,5'-dithio-bis-(2-nitrobenzoic acid)) was from Pierce. 1,2-Dithiodecanoyl-1,2-dideoxyl-*sn*-glycero-3-phosphocholine was prepared as described [5]

and was a gift from H.S. Hendrickson (Univ. of Washington). Bovine serum albumin was purchased from Sigma.

SURFACE PLASMON RESONANCE

Two distinct SPR sensing systems are used here, a planar device and a fiber optic device (used only in Chapter 3). Both systems use a white-light source incident at angles fixed by the detector geometry. Reflected light was measured using a photodiode array spectrometer (Ocean Optics Inc. S1000); each spectrum was then manipulated by computer to find the wavelength of minimum reflection (software written by R. Jorgensen and provided by EBI Sensors).

The fiber optic SPR sensor is described elsewhere [6] and consists of a ~15 cm length of 400 μm diameter optical fiber, one end of which has been stripped of 10 mm of its jacket and optical cladding. A ~50 nm electron-beam-evaporated gold film is evaporated (on top of a 2 nm chromium adhesion layer) onto the surface of the cylindrical stripped region, forming the SPR sensor surface. A gold mirror (~150 nm thick) is deposited onto the polished flat end of the stripped end of the fiber. The other end of the fiber is mounted in a fiber-optic connector. White light travels down the fiber and strikes the SPR sensing surface at a range of angles fixed by the modal distribution of the fiber. The reflected light strikes the mirror on the end of the fiber and bounces back up the fiber into the spectrometer. The probe senses the RI of whatever medium surrounds its sensing surface. The probe was not used in a flow cell; instead, the probe tip was immersed in a beaker of analyte, and the solution in the beaker was changed (for example, by adding solute or solvent). Due to the temperature sensitivity of the index of refraction of solutions, it is important to keep its temperature constant.

With planar device (shown in Fig. 2.1), the ~50 nm gold (and chromium adhesion layer) is deposited onto a diposable glass microscope slide which is then index-matched to a prism. The collimated white light source is directed with a fiber optic through the prism and this glass slide to impinge onto the gold sensing surface at an angle of 78° from normal, and the specularly-reflected light exiting the prism is collected into a fiber optic jumper cable which directs it into a diode array spectrometer for analysis. (The incident light is linear transverse-magnetic-polarized, so that its magnetic field is parallel to the gold surface.) A flow cell is mounted onto the sensor / prism assembly so that solutions of sensing interest can be easily introduced to flow across the gold surface, and switching between different solutions can be accomplished rapidly. In order to provide better temperature stability and control, the optical system is fixed to an aluminum mount and enclosed in a thermally insulated box.

Proper refractive index matching between the prepared slide and the prism was achieved with a glycerol-silica solution. An 80 μl flow cell was fitted over the sample using a silicone gasket. This flow cell was attached, via poly(ethyl ether ketone) (PEEK) tubing and fittings (Upchurch), to a syringe pump (Cavro) and two valves (VICI) which allow rapid switching, with a time constant of ~1 second (see Fig. 2.2), between solvent and sample loops. The pump is controlled by software presented in Appendix B.

The basic experimental protocol is described in great detail in the SPR Manual (Appendix A). Experiments start with a steady baseline in solvent. Adsorption is then initiated by injection of sample from the sample loop to the flow cell. After the adsorption is complete, the starting solvent, buffer for protein experiments or absolute ethanol for thiol experiments, is then injected into the flow cell to rinse any physiosorbed species and to correct for a baseline shift due to refractive index difference with respect to the sample solution. The observed SPR shifts are converted into thicknesses or coverages with a formalism that is described in Chapter 3 and the SPR Manual (Appendix A).

A continuous flow procedure (only used in Chapter 7) is similar to that described above. A Milligat pump (GlobalFIA), a positive displacement piston array pump, provides continuous and nearly pulseless flow (at a rate of 160 μL / min here). As the solution exits the flow cell, the liquid can be directed out to waste or back to a reservoir for recirculation. The sample concentration is adjusted by addition of buffer or an aliquot of a more concentrated sample solution to the reservoir. Because of the nature of the recirculating flow it take a few minutes for such an addition to be fully mixed throughout the total system volume. In control experiments we followed the time dependence of the concentration as a reservoir containing 200 μl of solution with a higher refractive index was circulated into the system. The expected damped oscillations observed showed that complete mixing is achieved in less than 10 minutes (see Fig 1.3). To ensure complete mixing during experiments, the solution in the reservoir was stirred and recirculated for at least 10 minutes after each addition of sample. For these experiments, the solutions were recirculated to ensure that binding of the sample to all surfaces had reached steady state. Thereafter, analysis of a sample of solution from the reservoir should provide the solution concentration at equilibrium.

PREPARATION OF SUBSTRATES

The gold-coated fiber-optics probes were first cleaned by sequential soaking for several minutes in boiling solvents (toluene, octane, n-hexane, absolute ethanol and deionized water), rinsed with absolute ethanol and then dried with N_2. The probes were immersed in solvent immediately after drying.

The gold-coated glass microscope slides for the planar SPR were cleaned in basic peroxide solution (a 1:1:5 solution of NH_4OH, 30% by vol. H_2O_2, and deionized water [7]). The slides were placed in the solution and heated to 40-70°C for about five

minutes. The samples were then removed, rinsed thoroughly with deionized water (during which it was tested for hydrophillicity), rinsed with ethanol, and dried with N_2.

For pre-functionalized surfaces, the gold films were then placed in a jar containing 1.0-0.1 mM solution of the desired thiol in degassed (N_2) absolute ethanol. The jar was flushed with N_2 and then sealed (with the lid and parafilm). The samples were left in the solution for 24-72 hours to build the thiolate adlayer, and then removed and rinsed with ethanol to remove excess and weakly bound thiols before use. For biotin containing monolayers for SA immobilization in Chapter 5, 6, and 7, solutions of BAT and PEO (shown in Fig. 2.4) were prepared with a net thiol concentration of 0.1 mM in dried, deoxygenated 100% ethanol.

COMPETITIVE DESORPTION OF SA

For competitive desorption experiments in Chapters 5 and 6, after the streptavidin the streptavidin had time to adsorb to the biotinylated surface, buffer containing a large molar excess of biotin (1 mM biotin in PBS) was injected into the flow cell to initiate competitive desorption. PBS is injected again at the end of the experiments to correct for a baseline shift due to refractive index difference with respect to the biotin buffer.

NOTES TO CHAPTER 2

[1] Nelson, K. E.; Jung, L. S.; Gamble, L.; Boeckl, M.; Naeemi, E.; Campbell, C. T.; Castner, D. G.; Stayton., P.S., *Langmuir (submitted)*.

[2] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[3] Klumb, L. A.; Chu, V.; Stayton, P. S. *Biochemistry* **1998**, *37*, 7657-7663.

[4] Chilkoti, A.; Tan, P. H.; Stayton, P. S. *Proc. Natl. Acad. Sci. USA.* **1995**, *92*, 1754-8.

[5] Hixon, M. S.; Ball, A.; Gelb, M. H. *Biochemistry* **1998**, *37*, 8516.

[6] Jorgenson, R. C.; Yee, S. S. *Sens. Actuators B* **1993**, *12*, 213.

[7] Uvdal, K.; Bodo, P.; Liedber, B. *Journal of Colloid and Interface Science* **1991**, *149*, 162-173.

FIGURES FOR CHAPTER 2

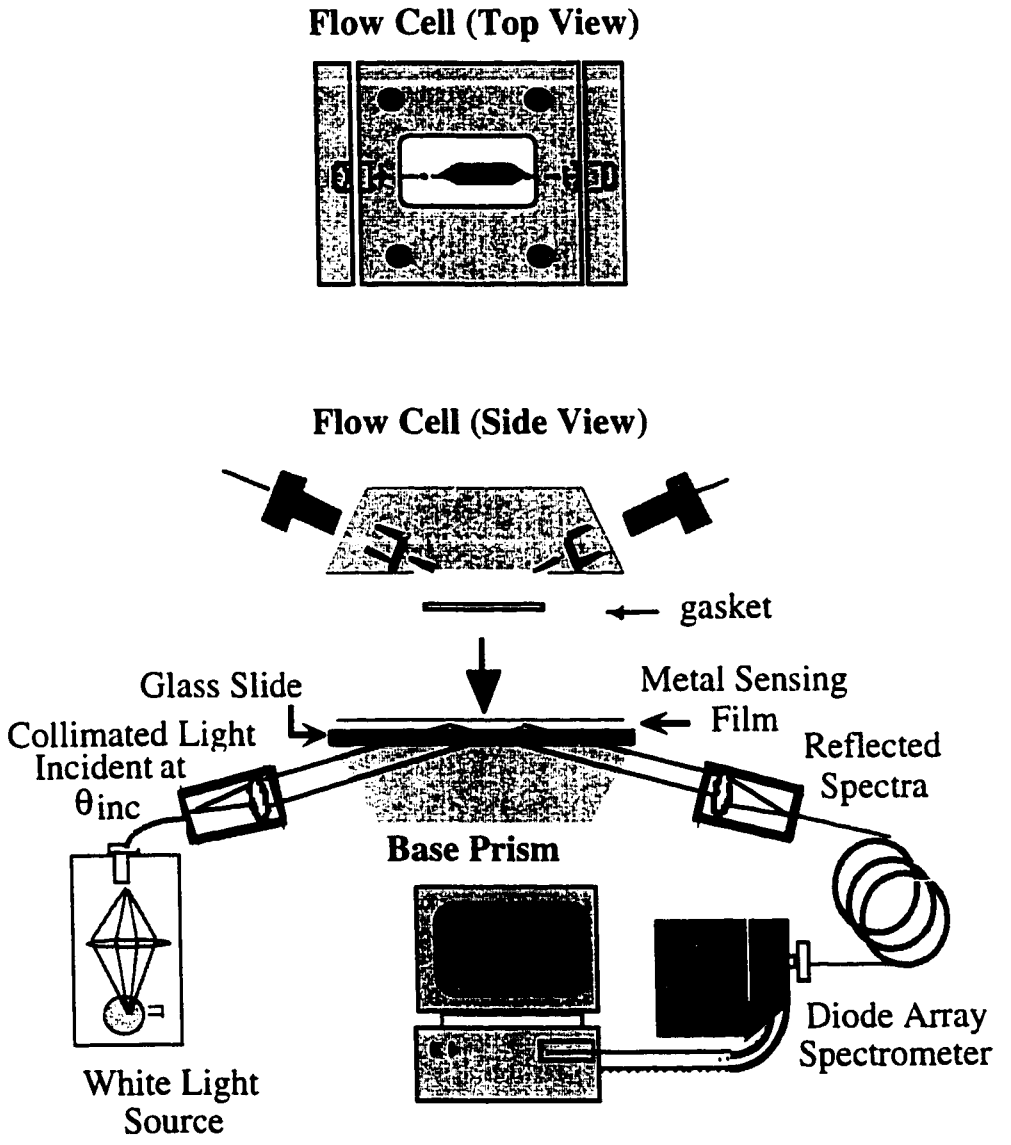


Figure 2.1 Schematic diagram of the planar SPR spectrometer and flow cell system (adapted from a figure originally provided by M. N. Mar). The top view of the flow cell shown at the top of the figure, shows how the gasket forms the flow channel which defines the cell volume. The side view shows how the flow cell is placed on the prism.

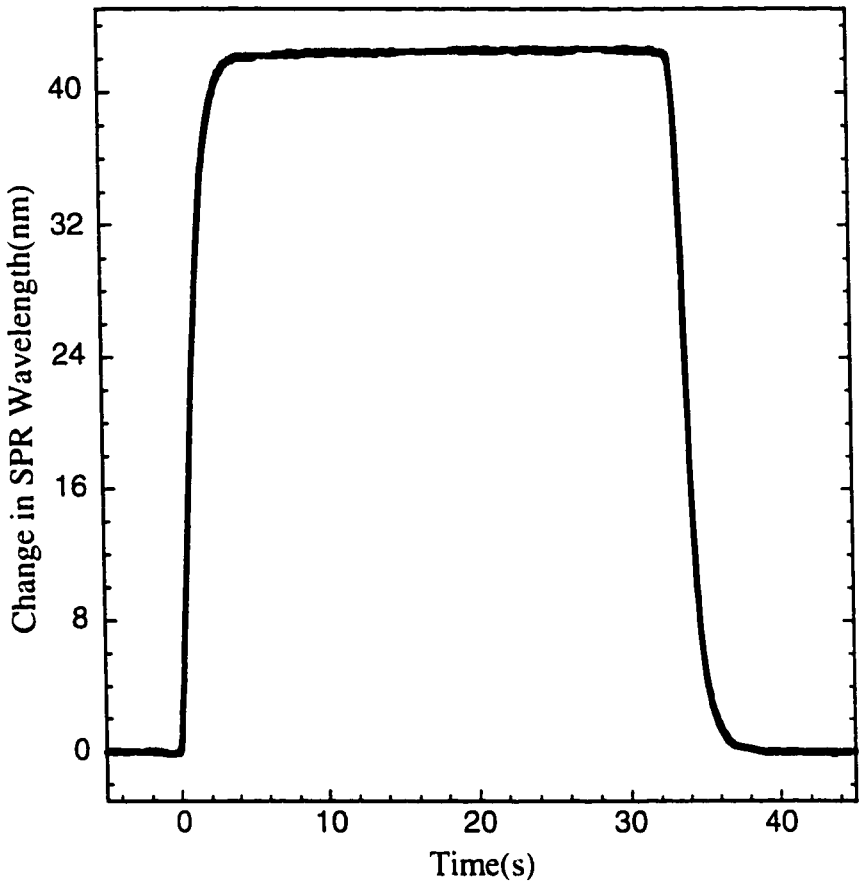


Figure 2.2 A demonstration of the time response of the SPR system. The flow cell which contained water was injected (at time 0) with a solution of 20% ethanol in water (by volume), resulting in a refractive index change of about $1x10^{-2}$. Water is then injected to rinse out the ethanol/water solution. A single exponential fit yields time constants of ~1 second for both steps, which gives the instrumental time response.

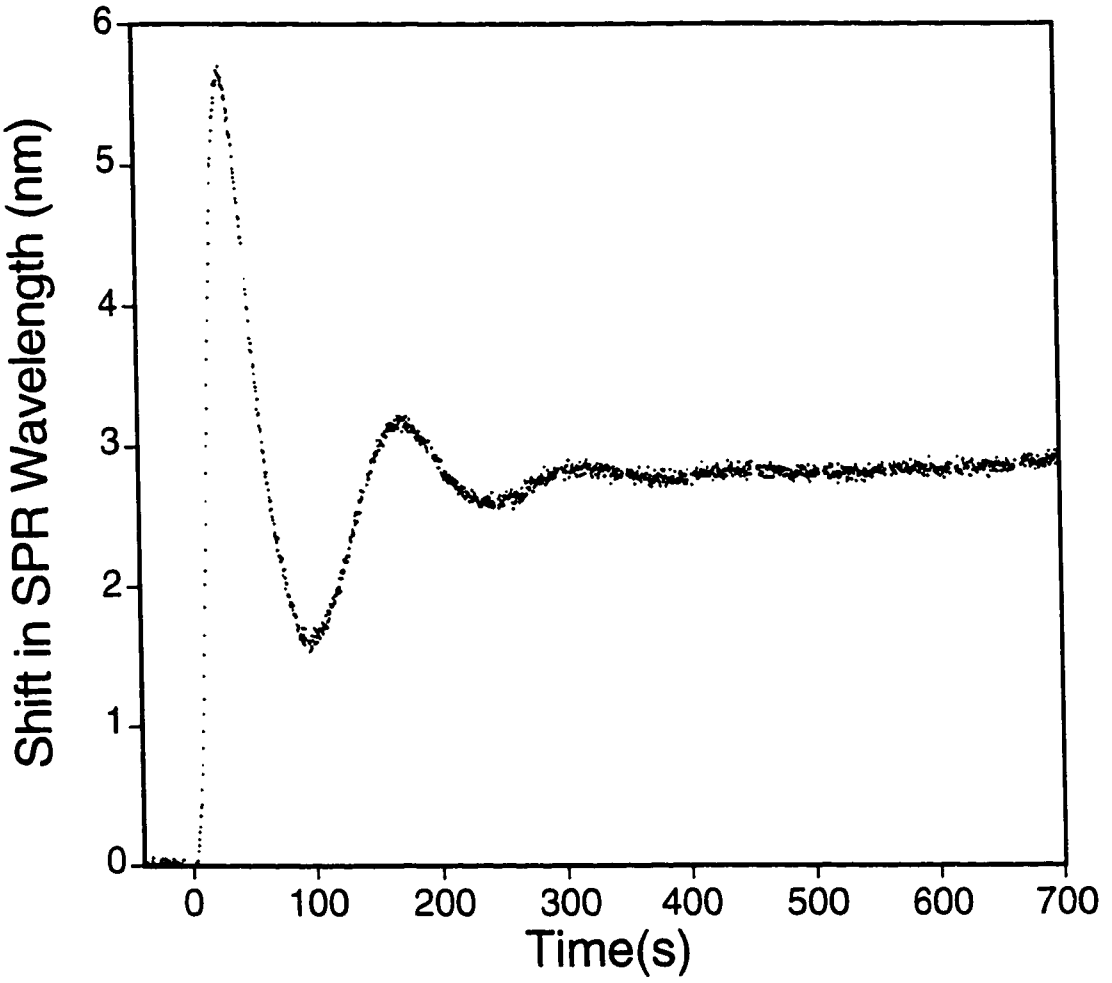


Figure 2.3 The time dependence of the concentration as a reservoir containing 200 µl of a higher refractive index solution is circulated into the system. The expected damped oscillations observed showed that complete mixing is achieved in less than 10 minutes

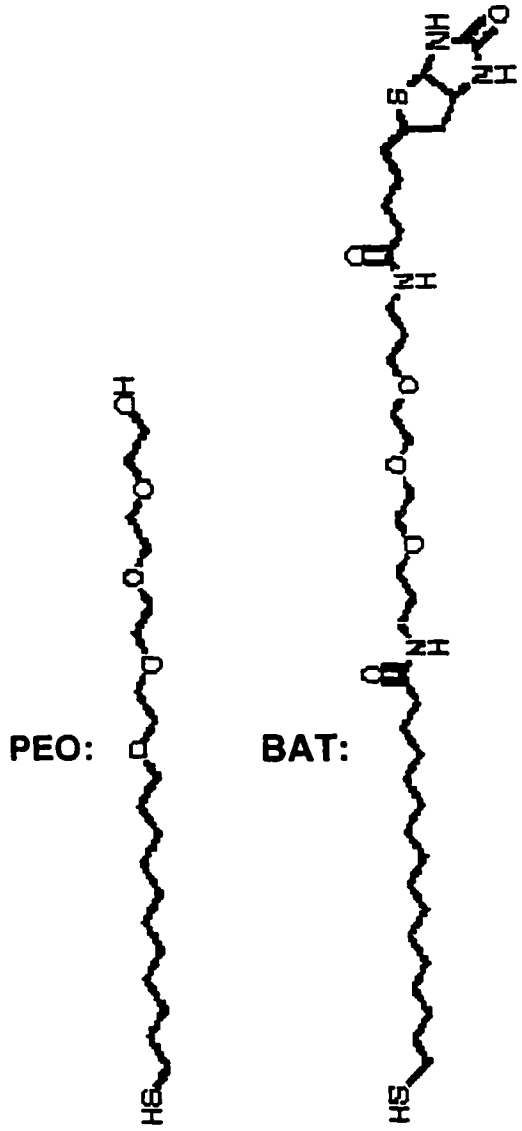


Figure 2.4 Bond-line formulas for PEO and BAT

CHAPTER 3: QUANTITATIVE INTERPRETATION OF THE RESPONSE OF SURFACE PLASMON RESONANCE SENSORS TO ADSORBED FILMS

INTRODUCTION

Sensors based upon surface plasmon resonance (SPR) spectroscopy are becoming increasingly popular due to their high sensitivity and simple construction [1-16]. SPR-based sensors can detect refractive index (RI) changes smaller than 10^{-5} with a time resolution of a few seconds (e.g., see [14] and data presented below). Because it senses with an evanescent wave, an SPR sensor responds to the RI of its analyte only to a depth of ~200 nm from the sensor surface. This allows SPR sensors to be applied to the detection of adsorption onto the sensor surface from liquid solutions [1-14, 16], even in a microscopic mode with ~5μm lateral resolution [17]. The binding of an adsorbate to the metal surface of the SPR sensor, or to an immobilized functionalization layer on that metal surface, may be monitored in real time if the adsorbate has a different refractive index than the bulk solution. If the sensor surface is functionalized to contain selective receptors for specific chemicals or biomolecules, selective chemical and biochemical sensors can be constructed. SPR spectroscopy also has many potential applications in fundamental surface science, since it is one of the few techniques that can directly detect adsorption onto a surface immersed in liquid with high time resolution and submonolayer sensitivity.

To apply SPR sensing to the monitoring of adsorption processes, some method must be available to extract from the SPR sensor response a quantitative measurement of the thickness or surface concentration of the adsorbed layer. However, in spite of the wide application of SPR, no simple yet general method for predicting the response of an SPR sensor to a given adsorbate has been published. It is the main

purpose of this chapter to present a new, simple but quantitative mathematical formalism for interpretation of SPR signals from adsorbed films of a wide variety of structures. We will also test this formalism by: (1) calibrating the response of two SPR spectrometers to changes in bulk index of refraction, (2) using these calibrations with this formalism to predict responses to several well-characterized adlayer structures, and then (3) comparing these predictions to measured SPR responses. We supplement these thin adlayer experiments with optical simulations of the sensor response to the growth of thicker films. We will also discuss methods for characterizing SPR detection limits in both bulk and adsorption-based analyses, and present these for the two SPR systems used here. Finally, we will characterize the temperature sensitivities of these two SPR spectrometers and discuss how they affect detection limits.

An SPR sensor consists of a transparent optical substrate coated with a thin (~50 nm) metal film (see Fig. 3.1, for example). Here the metal is gold, but silver is also often used. The fluid medium (here a liquid solution) to be analyzed is in contact with the metal surface, and adsorption can take place from this solution onto the metal surface. Often the metal is purposefully coated with something else, so that the adsorption sites are not necessarily metal sites. Light passes through the substrate and reflects from the interface between the substrate and the analyte medium at an incident angle or range of angles greater than the critical angle. One then measures the specularly reflected light intensity. Depending on the spectrometer, one can either use monochromatic light, in which case one monitors intensity versus incident angle, or a white light, in which case one measures intensity versus wavelength at a fixed angle. For certain wavelengths and angles of incident light, part of the incident energy will couple into a surface plasmon wave traveling along the interface between the gold layer and the solution to be analyzed. The loss of this energy is observed as a sharp attenuation of reflectivity, known as the surface plasmon resonance effect. The angles and wavelengths at which this occurs vary extremely sensitively with the refractive index (RI) (or complex

dielectric constant) of the medium in contact with the metal surface of the SPR sensor [2, 8, 18-20].

The sensor response is characterized by the wavelength λ or angle θ of minimum reflectivity. Shifts in these quantities ($\Delta\lambda$ or $\Delta\theta$) thus yield measurements of changes in the RI, $\Delta\eta$, of the medium to be interrogated. That medium need not have a uniform RI, but instead may have a more complex structure, such as an adsorbed film in a liquid solution (Fig. 3.1b). The sensor response thus reflects some sort of an *average* RI. The way this averaging occurs is a major subject of this chapter. We will show that the thickness of an adlayer or the surface concentration of an adsorbate can be determined by matching the predicted "average" RI for a given thickness or surface concentration with that actually measured by the SPR sensor.

Several papers have previously addressed some aspects of the quantitative interpretation of SPR signals in terms of adlayer structural parameters. [2, 5-8, 13, 15-17, 20-23]. Most of these addressed quantitative analysis of adsorbed films that are much thinner than the decay length of the evanescent field, in the so-called linear-response regime. Stenberg et al. [6] used full numerical simulations of Maxwell's equations for thin-film structures to show that a nearly linear response is obtained versus surface concentration of adsorbed proteins, and confirmed this by radiolabelling experiments. Kooyman et al. [5] similarly calculated thicknesses from the measured response to protein adsorption using Maxwell's (Fresnel) equations, and showed it to agree with the dimensions of the protein molecule. Liedberg et al. [8] measured a linear response to surface concentration of adsorbed proteins (in ng / mm^2) with a slope that was nearly independent of the protein. Peterlinz et al [13] calculated film thicknesses (versus their dielectric constant) from the measured SPR response to a monolayer of adsorbed alkyl thiols, and from this showed how to determine fractional adsorbate coverages from the response at lower coverages. In general, these papers show that quantitative estimates of adlayer film thickness or adsorbate coverage can be determined in this linear regime if a

calibration curve is first made wherein the response is plotted versus known thicknesses or surface concentrations. Such a linear calibration curve is the method generally used with commercial instruments [16] and is based on a model by Sjolander et al. [24]. It can only be used in studying the same adsorbate or adsorbates that have similar dielectric properties to that used in generating the calibration plot.

A few papers have addressed the response to thicker films, beyond the linear regime. Liedberg et al. [8] showed through Maxwell's (Fresnel) equations that the SPR response to local changes in the index of refraction decays exponentially with distance from the surface with a characteristic decay length equal to one-half that for the decay of the evanescent field Lukosz [2, 20] used a perturbation theory approach on Maxwell's equations to calculate the effective refractive index of uniform adlayers, showing that this varied linearly with film thickness for small thicknesses but showed increasing non-linearity at high thicknesses. He similarly showed how to estimate the effects of any anisotropy of the optical properties of the adlayer.

Here, we will show how to estimate adsorbed film thicknesses, surface coverages or surface concentrations from the SPR response without relying on calibration curves of response versus known thicknesses or surface concentrations. Such surface-based calibration curves are very difficult to make. Here, we will show how to use a much simpler calibration curve: SPR response versus bulk refractive index. We will also show how to apply this simpler calibration over the entire surface concentration range, from the linear-response regime of small thicknesses up to films much thicker than the decay length of the evanescent field.

A method for estimating the refractive index of the adlayer will also be discussed here which relies on knowledge of the composition of the adsorbate. More complex SPR measurements using more than one color [25, 26] or detailed lineshape analysis [27] have recently been suggested which potentially give information on both the film thickness and the index of refraction of the adsorbate. Those will not be discussed

here, since this chapter focuses on simpler sensors wherein only the SPR minimum (wavelength or angle) is monitored.

EXPERIMENTAL

The data presented in this chapter was taken using both the planar and fiber optic SPR systems as described in the Chapter 2.

The index of refraction of the various bulk solutions (sucrose / water, ethanol / water, toluene / ethanol) was measured with an Abbe-3L Refractometer (Milton Roy Co., Rochester, NY).

THE SPR RESPONSE TO BULK SOLUTIONS: MEASURED CALIBRATION PLOTS

Let us first define the SPR response as either the shift in wavelength ($\Delta\lambda$) or angle ($\Delta\theta$) of the SPR minimum in reflected light intensity associated with changes in the index of refraction of the medium in contact with the metal surface of the SPR device, $\Delta\eta$. We have frequently measured the SPR response to changes in the bulk index of refraction of solutions in contact with the gold surface for both of the SPR systems of this study. Some typical calibration plots are shown in Fig. 3.2, for both the planar and fiber-optic SPR probes. The wavelength of the SPR minimum is plotted versus refractive index of the solution in contact with the gold, using several different binary solutions (sucrose / water, ethanol / water, toluene / ethanol), each with a range of concentrations. For these solutions, the change in wavelength due to differences in adsorption between different solutions should be very small compared to the measured shifts, which are dominated by changes in the bulk index of refraction of these solutions. As can be seen,

the response in all cases is fairly linear over a narrow enough RI range, but curvature is obvious when viewing the broader index range available. (Note that this upward curvature is predicted by Maxwell's equations in simulations like those below.) Similar results have been reported previously [8]. Thus, over a narrow range, the response to changes in bulk index of refraction, in the absence of adsorption from the solution, can be approximated as linear:

$$R = m \cdot \Delta\eta = m \cdot (\eta_{final} - \eta_{initial}). \quad (1a)$$

The magnitude of the local slope, m, in any small range of indices can be thought of as a sensitivity factor for the measurement. In order to make the units on this slope clearer for the reader, we refer here to a "refractive index unit" or RIU. While this is unnecessary since the refractive index is really unitless, we believe it makes the chapter easier to follow.

Fig. 3.3 shows a plot of the wavelength of the SPR minimum for the planar system over a much smaller range of index of refraction near 1.335 RIU, collected over a narrower time range (several hours) to minimize drift. Linearity as in Eq. 1a with slope 3400 nm / RIU is obvious. Returning to Fig 2.2, this sensitivity factor, or local slope, varies from ~3100 to 8800 nm / RIU with increasing index of refraction for the planar system, and from ~1600 to 6000 nm / RIU for the fiber-optic probe. Note that these slopes depend on the gold film thickness and other system parameters, and so they should be measured regularly using a few points. The observation of a reasonable slope also provides an easy means for verifying an instrument's integrity.

As described in more detail below, the curvature in such calibration plots could be explicitly incorporated into a low-order polynomial equation, for example by adding a second term to Eq. 1a, $m_2 \cdot \Delta\eta^2$, which would be negligible for small $\Delta\eta$:

$$R = m_1 \cdot \Delta\eta + m_2 \cdot \Delta\eta^2. \quad (1b)$$

CALCULATING THE SPR RESPONSE TO ADSORBED FILMS

THE SPR RESPONSE TO A SINGLE ADLAYER OF UNIFORM THICKNESS

First consider the idealized bilayer structure in Fig 2.1b, wherein a thin adsorbed film of uniform thickness d and index of refraction η_a is bonded to the metal surface of an SPR probe. Above this adsorbate layer is a bulk liquid solution of index η_s. What should the SPR response be for such a structure?

In this case we define the SPR response, R, as either the shift in wavelength ($\Delta\lambda$) or angle ($\Delta\theta$) of the SPR minimum in reflected light intensity associated with this adsorption starting from the clean metal in contact with the same bulk solution, with no intermediate adlayer. If one could determine some effective index of refraction for the bilayer, η_{eff}, which would be the properly weighted average of η_a plus η_s, then one could simply estimate the response to adsorption (i.e., the change in position of the SPR minimum upon adsorption) using Eq. 1a or 1b. Thus, the estimated response would just be:

$$R = m\cdot(\eta_{eff} - \eta_s), \tag{2a}$$

or

$$R = m_1\cdot(\eta_{eff} - \eta_s) + m_2\cdot(\eta_{eff} - \eta_s)^2, \tag{2b}$$

depending upon whether one is using the linear or quadratic calibration plot.

In Eqs. 2, η_{eff} must be the properly weighted average of η_a plus η_s. Since light is being used to probe this index of refraction, it is natural to assume that the proper

weighting factor at each point in the bilayer structure should be proportional to the intensity of light that point. The evanescent electromagnetic field decays away into this medium with a characteristic decay length, l_d, of ~25-50% of the wavelength of the light [2, 8, 20, 36]. (The light is typically ~500-900 nm at the SPR minimum.) The intensity of light is the field strength squared, so it decays with height z above the metal surface as $[\exp(-z / l_d)]^2$. Thus, the proper weighting factor in calculating this average refractive index should just be $[\exp(-z / l_d)]^2 = \exp(-2z / l_d)$. This was indeed proven to be very accurate through Maxwell's equations by Liedberg et al. [8]. Thus, the effective index of refraction is calculated by averaging the index of refraction over the depth of the whole bilayer structure, always weighting the local index with this factor. This average is therefore calculated with the depth integral:

$$\eta_{eff} = (2/ l_d) \int_0^{\infty} \eta(z) \cdot \exp(-2z / l_d) \cdot dz, \qquad (3)$$

where $\eta(z)$ is the index of refraction at height z. This equation is not restricted to the bilayer structure, and should be generally useful even for much more complex multilayer structures. It is very different from the model used by Sjolander et al. [24], which assumes unity weighting factor within the penetration depth of the evanescent wave and zero beyond.

For the bilayer structure of Fig. 3.1b,

$$\eta(z) = \eta_a \quad \text{for} \quad 0 < z < d,$$

and

$$\eta(z) = \eta_s \quad \text{for} \quad d < z < \text{infinity}.$$

In this simple case the integral of Eq. 3 reduces to:

$$\eta_{eff} = \eta_a \cdot [1 - \exp(-2d / l_d)] + \eta_s \cdot \exp(-2d / l_d)$$

$$= \eta_s + (\eta_a - \eta_s) \cdot [1 - \exp(-2d / l_d)]. \quad (4)$$

A similar functional dependence on film thickness to Eq. 4 has been proven through a perturbation theory approach from Maxwell's equations by Lukosz [2]. Also, a similar approach to Eqs. 3 and 4 has been suggested for optical waveguide sensors [37], although the decay length suggested there is twice what we propose below.

If using a linear calibration plot of response versus bulk RI, then Eq. 2a applies and by substituting Eq. 4 for η_{eff} one gets from it the sensor response:

$$R = m \cdot (\eta_{eff} - \eta_s) = m \cdot (\eta_a - \eta_s) \cdot [1 - \exp(-2d / l_d)]. \quad (5a)$$

Behavior qualitatively similar to Eq. 5a has been predicted based on calculations using Maxwell's equations by Liedberg et al. [8]. A similar exponential dependence to Eq. 3 has been previously assumed for the effect of mass loading of receptors versus depth within a dextran coating on an SPR sensor, although in that case no connection was made to refractive index, and no method for absolute quantitative calibration or estimating the decay length was proposed [38]. Our present results substantiate the assumptions in that paper.

In Fig. 3.4 we plot as the solid curve the SPR response (in this case the angle of the reflection minimum at a fixed wavelength of 825 nm) predicted by Eq.5a as a function of adlayer thickness calculated for a particular value of m, η_a and η_s, and l_d. We also present there as dots the results of a much more complex calculation of the sensor response versus thickness from a thin-film optical model based on Maxwell's equations, similar to the methods described in [5, 6, 8, 39]. This calculation was for a planar multilayer consisting of a glass substrate, supporting a 1 nm thick chromium layer and then a 50 nm thick gold layer, and it used the same values for η_a and η_s and light wavelength as used in Eq. 5. Note that the value of m used in Eq. 5a as plotted here, 107

degrees per RIU, was chosen to reproduce the response to bulk solutions, calculated with Maxwell's equations and the same optical model, as shown in the insert of Fig 2.4. This is exactly how m would be determined experimentally: by calibrating the measured sensor response to changes in *bulk* refractive index as outlined above. The value of l_d (368 nm) was treated as a fitting parameter in Eq. 5a.

Note the excellent agreement in Fig. 3.4 between the full calculation of Maxwell's equations and the far simpler Eq. 5a. This verifies the validity of Eqs. 3 - 5a, and the assumptions used in deriving them. Also note that the best-fit value of l_d (368 nm) is in the range between 1/2 to 1/4 the wavelength of light used to probe the SPR minimum, as expected based on the characteristic decay length of the evanescent field [2, 8, 20, 36]. In the calculations with Maxwell's equations, we also calculated l_d for these conditions, and we found it to range from 370 to 320 nm with increasing refractive index (adlayer thickness). These values encompass the single value of l_d determined from fitting Eq. 5a to the more complex calculations.

The above development assumed that the SPR response was proportion to the change in bulk refractive index over the range between η_s and η_{eff}. If this is not the case, one can use the more complex nonlinear calibration curve, Eq. 1b, to replace Eq. 1a for bulk solutions, in which case the response to an adlayer is given by Eq. 2b. Upon substitution of Eq.(4) for the effective refractive index of the bilayer into Eq. 2b, we have:

$$R = m_1 \cdot (\eta_a - \eta_s) \cdot [1 - \exp(-2d / l_d)] + m_2 \cdot \{(\eta_a - \eta_s) \cdot [1 - \exp(-2d / l_d)]\}^2. \qquad (5b)$$

The constants m_1 and m_2 would be determined from the calibration curve, fitting the response to the index of bulk solutions with the quadratic Eq. 1b.

Equation 5b should be used in place of Eq. 5a whenever there is substantial nonlinearity in the SPR response to changes in the bulk refractive index in the range between η_s and η_{eff}, which generally occurs when their difference is large. Fig. 3.5 shows that it gives an excellent fit to the solution of Maxwell's equations under such

situations, again using the optical model described above and again a light wavelength of 825 nm. The geometry was the same as used in Fig 2.4, with the only difference being the much larger difference between η_a and η_s here. The constants m_1 and m_2 were again determined from a theoretical calibration curve, fitting Eq. 2b to the SPR response to bulk solutions calculated with Maxwell's equations, using a series of hypothetical bulk solutions covering the range between η_a and η_s here. This fit is shown as an insert to Fig 2.5, and gave values of m_1 = 89.7 degrees per RIU and m_2 = 980 degrees per RIU^2. Note that this is exactly what would be done experimentally to get m_1 and m_2. The best-fit value of l_d for Eq. 5b in Fig 2.5 was 307 nm, again within the expected range of 1/2-1/4 of the light's wavelength. The good fit further verifies Eqs. 3 - 5b. Note that the value of l_d found here is ~20% lower than the value determined by the fit to Eq. 5a in Fig 2.4, 368 nm. This is consistent with the value of l_d found from our optical calculations, which decreased more dramatically, from 370 to 220 nm with increasing refractive index in this range.

When the SPR response to changes in bulk index of refraction cannot be well fitted by Eq. 1a or 1b, respectively, then one can certainly expect substantial errors in the above methods.

Estimation of Adsorbate Film Thickness from Measured SPR Response

Equation 5a or 5b can be used to estimate an adlayer thickness from a measured SPR response as well as to predict the response from a certain adlayer structure. Note that the constant m in Eq. 5a or m_1 and m_2 would be determined first from calibration curves using bulk solutions of the type presented in Figs. 3.2 - 3.3. The index of refraction of the bulk solution, η_s, can either be obtained from prior measurements (see, for example [40]) or measured, as we have done on a standard Abbe refractometer. We present below methods for estimating η_a if it cannot be determined by these methods. The value of l_d can be roughly estimated as 0.37±0.13 the light wavelength (see above), but better methods are discussed below. Thus, Eq. 5a or 5b

above can be used to predict the SPR response to an adlayer of a certain thickness, or conversely to estimate the adlayer thickness from the measured SPR response. Equation 5a can be rearranged to solve for the adlayer thickness:

$$d = -(l_d / 2)\ \ln(1 - R / R_{max}), \tag{6a}$$

where R_{max} is the maximum response that would be measured for an infinitely thick adlayer, or:

$$R_{max} = m\cdot(\eta_a - \eta_s). \tag{6b}$$

Note that R_{max} could be directly measured in some cases, but in most cases (such as for protein or thiol adsorption, see below) it is just calculated from Eq. 6b as the calibration slope m times the difference between the refractive indices of the adsorbate and solvent.

In the special case where d is very small compared to l_d, Eq. 6a reduces to:

$$d = (l_d / 2)\cdot(R / R_{max}) = (l_d / 2)\cdot\{R/[(m\cdot(\eta_a - \eta_s)]\}, \tag{7}$$

so that the response is directly proportional to the thickness of the adlayer. This is what we will call the "linear response regime". The linear response to film thickness or adsorbate surface coverage has been predicted and observed in many papers where the adsorbate was not too thick [2, 5, 6, 8, 13, 16, 20].

In using Eqs. 5 - 7 to calculate film thickness, the absolute accuracy will only be as good as this estimate of l_d, which is only ±35% in the absence of some calibration or better means of estimating it (which we will discuss below). The precision, however, is within a few percent, as seen by the quality of the fit to Eqs. 5a and 5b. Inaccuracy can also arise if the value of m or η_a used is inaccurate. At small thicknesses, the relative error in m or $(\eta_a - \eta_s)$ or directly appears as the same relative error in the thickness. The relative error due to these grows rapidly as the true thickness increases to

values near l_d / 2. For example, the error in thickness is 35% for a 20% error at a thickness of 0.35 l_d.

Estimation of Adsorbate Coverage from Measured SPR Response

Once the average thickness, d, of a uniformly-spread adlayer is estimated by the above procedure, it is a trivial matter to convert this to surface concentration, θ, in molecules per cm^2. The conversion factor is just the bulk number density of the adsorbate, N, in units of molecules per cm^3:

$$\theta \text{ (in molecules per cm}^2\text{)} = d \text{ (in cm)} * N \text{ (in molecules per cm}^3\text{).} \quad (8)$$

The bulk number density of the adsorbate, N, can be estimated from the bulk density of the adsorbate, ρ, in units of grams per cm^3 just by dividing by the molecular weight and multiplying by Avogadro's number. The proper value of ρ to use is the value for pure, condensed bulk adsorbate, which is the same material whose index of refraction η_a was used in Eq. 6, 7 or 5b to get d. If the adsorbate is a molecular fragment, its density can be estimated from that of similar molecules. (See next section for suggestions on how to choose similar molecules.)

Estimating the Refractive Index of an Adsorbate, η_a

How does one determine η_a, the adsorbate's RI? If the adsorbate is a molecule that can be assumed to be relatively unperturbed upon adsorption, the appropriate value for η_a is just the index of refraction of the molecule in pure, condensed form, which can also usually be found (see [40]) or measured. Most proteins, for example, have an index of refraction near 1.6 [41], and this is not expected to be perturbed much upon adsorption since the molecule is so big relative to the fraction of it that would actually form bonds to the surface. If it is a molecule or fragment of a molecule whose refractive index cannot be measured, it can be estimated based on values for molecules with similar structure. For example, adsorbed alkyl thiolates on gold can

be expected to have a very similar RI to the corresponding thiol, since the H atom that is lost upon adsorption occupies such a tiny fraction of the molecular volume. Similarly, adsorbed alkyls, carboxylates or ammonium cations could be estimated from the corresponding alkane, carboxylic acid or amine, respectively.

A more complex method of estimating refractive indices are based on the Clausius-Mossotti Equation [42], which for a sample of pure compound j is:

$$(\eta_j^2 - 1) / (\eta_j^2 + 2) = N_{j,0} A_j / (3(\varepsilon_0), \quad (9)$$

Here η_j is its RI, ε_0 is the permittivity of vacuum, $N_{j,0}$ is the number density (number of molecule j per unit volume in pure j), and A_j is the frequency-dependent polarizability of the molecule. First, let's consider a fluid solution. For a mixture of compound j with k, each with number densities N_j and N_k, one can estimate the index of refraction of the solution, η_{soln}, using the same formula, but replacing η_j with η_{soln}, and $N_{j,0}A_j$ with the sum $N_j A_j + N_k A_k$. If the solution is ideal, then $N_j = f_j N_{j,0}$, where f_j is the volume fraction of j (i.e., the fraction of its volume occupied by j), and likewise for k. Using Eq. 9 to express A_j and A_k in terms of η_j and η_k, respectively, gives the Lorenz-Lorentz equation [43, 44]:

$$(\eta_{soln}^2 - 1) / (\eta_{soln}^2 + 2) = f_j [(\eta_j^2 - 1) / (\eta_j^2 + 2)] + f_k [(\eta_k^2 - 1) / (\eta_k^2 + 2)] \quad (10a)$$

This turns out to give a nearly linear variation in the RI of the solution with the fraction of its volume occupied by k, so that the following is true to within a few percent when η_j and η_k are both within the usual range of SPR applications (from 1.33 to 1.6):

$$\eta_{soln} = f_j \eta_j + f_k \eta_k = f_j (\eta_j - \eta_k) + \eta_k. \quad (10b)$$

Equation 10a can be applied to determine the RI of molecules by measuring the contribution they make to the RI of a solution. For example, the addition of most

proteins to aqueous buffer solution (0.3 M NaCl) causes the RI to increase by 1.8×10^{-4} RIU for every g / L of added protein [41]. (Glycoproteins and lipoproteins have somewhat lower increases [41].) The specific volume of proteins in aqueous buffer is ~0.77 mL / g [45, 46]. Using this in Eq. 10b gives that $\eta_{protein} - \eta_{buffer} = 0.234$ RIU. Since in those solutions $\eta_{buffer} = 1.336$ RIU, we get that $\eta_{protein} = 1.57$ RIU for the water-free protein,. A nearly identical value was obtained from Eq. 10a in [43], where it was also shown that $\eta_{protein}$ varies weakly with the wavelength of light. These values are also very close to the index of refraction measured for crystalline proteins, 1.60 RIU [24, 41]. They are greater than the refractive index that is estimated for "adsorbed protein films" using ellipsometric approaches which assume a single optical thickness, since that the film volume includes a great deal of water (see below and [17, 47]). Here, we are instead just referring to that part of such films which are made of protein material itself, not water. We believe this approach, which neglects the intermixed solvent in the adlayer, is more direct and general for quantitative analysis of adsorbate coverages for proteins and adsorbates in general.

Equation 10 can also be applied as an approximation in estimating the index of refraction of a sample of a single molecule (or adsorbed molecular fragment) whose value is unknown. The molecule can be separated into parts whose indices of refraction are known, and the index of refraction of the groups can be summed after scaling each with a weighting factor equal to the fraction of the molecule's volume that is occupied by that group. The RIs of groups can be estimated from applying the formula to molecules of known RI. For example, the RI of $CH_3(CH_2)_4CH_3$ is 1.3751 and that of $CH_3(CH_2)_{16}CH_3$ is 1.4390 [40]. Assuming that a CH_2 group and a CH_3 group occupy the same volume, the above equation gives that η_{CH2} and η_{CH3} equal to 1.471 and 1.183, respectively. Using these group values in the same equation predicts the correct RIs for $CH_3(CH_2)_4CH_3$ and $CH_3(CH_2)_8CH_3$, 1.375 and 1.41, respectively [40]. This also explains why the RIs of long chain alkyls, whether they be thiols, alcohols or carboxylic acids, all

approach that for the corresponding alkane as the chain length increases [40]. Similar analysis using the shorter chain molecules can give the group contribution from the functional group. The volume of a functional group can be estimated from its geometry (bond lengths, angles) and van der Waals radii of its atoms, or it can be treated as a parameter and determined by fitting the equation to known RIs.

One can also measure η_a in cases where very thick adlayers can somehow be grown. By simply measuring the maximum response for an infinitely thick adlayer, one gets $R_{max} = m \cdot (\eta_a - \eta_s)$. Since the slope of the calibration plot, m, and the RI of the solvent are known, one can solve for η_a.

Note also that analysis of the SPR response at two wavelengths or the SPR lineshape can give both the adlayer thickness and its effective index of refraction (see Introduction).

Estimating the Decay Length, l_d

The decay length, l_d, is a key parameter in these calculations. The probe depth of the technique is one-half this decay length. A rough but reasonable estimate is that l_d equals 37±13% of the light wavelength at the SPR minimum, λ, (see above), which for a typical minimum of 680 nm in aqueous solutions gives a probe depth $l_d / 2 = \sim 120$ nm.

A more accurate estimate of l_d comes from the Maxwell's equations, by which [26]:

$$l_d = (\lambda / 2\pi) / \mathrm{Re}\{[\eta_{eff}^2 \varepsilon_{metal} / (\eta_{eff}^2 + \varepsilon_{metal})] - \eta_{eff}^2\}^{1/2}$$

$$= (\lambda / 2\pi) / \mathrm{Re}\{-\eta_{eff}^4 / (\eta_{eff}^2 + \varepsilon_{metal})\}^{1/2}, \qquad (11)$$

where ε_{metal} is the complex dielectric constant of the metal at that wavelength (which is reported in [48, 49]) and η_{eff} is the effective index of refraction of the probed medium. The latter is measured experimentally (by comparing the observed SPR response to a calibration curve like Figs. 2.3 - 2.4). Note that l_d varies only weakly with η_{eff}: For example, with a gold sensor film at a wavelength of 665 nm, l_d only decreases by 30% when η_{eff} decreases from the value of pure water (1.33) to that of a pure protein (1.57). It only decreases by 7% when a protein adsorbs from water to give an effective film thickness of 22 nm (i.e., several monolayers of protein).

It should be remembered that the use of a single value of l_d throughout the probe depth is really an approximation in bilayer and multilayer structures, since its value locally depends on the RI of the local medium. Such a local dependence could easily be incorporated into Eq. 3. However, it is not needed for most applications since (1) l_d varies only weakly with RI, and (2) adlayer thicknesses are typically small compared to l_d. Thus, a single, average value of l_d in Eqs. 3 - 8 or Eqs. 12 - 13 below is usually suitable to provide adsorbed amounts that agree within 15% of that obtained from such a rigorous treatment. This more rigorous approach is only needed when *both* the RI in two different layers are enough different to give large differences in l_d, *and* these two layers both have thicknesses that are larger than ~0.2 l_d.

In SPR sensor applications, it is common to have a trilayer structure of the type depicted in Fig 2.6, where adsorbate **b** is first used to attached a selective receptor to the metal surface, and adsorbate **a** is the analyte which subsequently binds thereupon from solution **s**. One might even have a multilayer structure with several different substances. Estimating the SPR response to such multilayer structures is a simple extension of Eq. 5a or 5b. Again, one simply uses Eq. 3 to calculate a properly weighted average refractive index for the multilayer structure and incorporate it into Eq. 2a or 2b. For the trilayer of Fig 2.6,

$$\eta_{eff} = \eta_b \cdot [1 - \exp(-2d_b / l_d)] + \eta_a \cdot [\exp(-2d_b / l_d) - \exp[-2(d_a + d_b) / l_d]$$

$$+ \eta_s \cdot \exp[-2(d_a + d_b) / l_d]. \qquad (12)$$

When substituted into Eq. 2a, this just gives:

$$\Delta R = m \cdot (\eta_a - \eta_s) \cdot [1 - \exp(-2d_a / l_d)] \cdot \exp(-2d_b / l_d)] \qquad (13)$$

for the SPR response to the addition of the analyte film, **a**. (Here, ΔR is defined as the SPR response only to the addition of **a**, *after* **b** was already present, and m is the slope of the calibration plot, taken in this range of η_{eff}.) Note the similarity of this expression to Eq. 5a: They differ only by the additional scaling factor of $\exp(-2d_b / l_d)$ in Eq. 13. The SPR response to adding adlayer **a** is just the same as in the simpler case of species **a** alone, except that its magnitude is reduced by this factor due to the intermediate layer of **b**. When **b** is very thin compared to l_d, this factor is nearly 1.00, and the responsivity to **a** is unaffected by the intermediate receptor adlayer between the metal and **a**. On the other hand, when **b** is thicker than l_d, the sensitivity to **a** is severely decreased, dropping by a factor of ~7 when the thickness of layer **b** is equal to l_d. When **b** is thick, the slope m may also be different than in the absence of **b**, but m is almost identical for thin **b**. Again, the use of a single value of l_d in Eqs. 12 - 13 is an approximation that is usually justifiable (see above).

CORRECTIONS FOR NONUNIFORMITY IN COVERAGE: ADSORBATE ISLANDS OR CLUSTERS

Often, an adsorbate forms thick clusters on the surface rather than spreading uniformly across the sample. When the islands are comparable in thickness to l_d, or thicker, the SPR response to the adsorbate is not as strong as when uniformly spread. Consider the structure of Fig 2.7, where adsorbate a covers a fraction f of the metal surface in islands that are of thickness d. In this case the SPR response upon adsorption, R, is just f times the response when f = 1 (i.e., for uniform coverage by a layer of the same thickness as the islands or clusters, for which equations are presented above). One can think of this system as having an effective index of refraction for the adlayer / solution structure, wherein weighted depth averaging is done as in Eq. 3, except that at each depth one must use an index of refraction which has been simply averaged over the surface area which is probed. (Such averaging parallel to the surface is actually only correct for island sizes that are small compared to the decay length of surface plasmons parallel to the surface, which is ~3 micrometers [50].) Large islands would give a complex SPR lineshape with a double minimum that could only be interpreted from detailed analysis of the lineshape.) The net result is that any of the material **a** which is further from the metal surface than it would be if uniformly spread does not contribute as strongly to the SPR response, due to the exponential decay of the evanescent field. When the thickest parts of islands are still thin compared to l_d, the response to a certain volume of adsorbate in such islands is the same as it would be if that volume of adsorbate were spread over the entire surface in a uniform thickness as in Fig 2.1b. This would be the case for the adsorption of many proteins, each molecule of which can be thought of as an island of adsorbate, but still thin compared to l_d,. For large adsorbates such as cells, a more complex treatment is needed, such as that in [37], which describes methods for

treating various cell shapes within a related formalism (in which the decay length appears to be twice what we propose).

Again, when a preexisting adlayer **b** is already present on the metal surface, the SPR response to such a nonuniform adlayer, ΔR, is just reduced by the factor $\exp(-2d_b / l_d)$ similarly to Eq. 13. Thus, the SPR response to complex, nonuniform and multilayer adlayer structures can very generally be estimated within this formalism.

COMPARISON TO EXPERIMENT: MEASURED SPR RESPONSE TO ADSORBED FILMS

THIOLATE ADSORPTION ON GOLD

Fig. 3.8 shows the planar SPR wavelength response versus time upon exposure of the clean gold -coated glass slide, first to a solution of pure ethanol and then, at time t = 0, after switching to an ethanol solution containing: (a) 1.0 mM $HS(CH_2)_{15}CH_3$, and (b) 0.1 mM $HS(CH_2)_{15}COOH$. These thiol solutions are known to produce a self assembled monolayer of the corresponding thiolate on the gold surface [29-34]. These are irreversibly bonded, as evidenced by the fact that the SPR wavelength after a 600 second exposures to these thiol solutions decreased by <0.6 nm when switched back to pure ethanol flow.

The thickness of the adlayers can also be estimated from the results of Fig 2.8 by using Eq. 6 (or Eq. 7) since these thicknesses are very small compared to l_d). The SPR responses at saturation (after 250 s) upon adsorption of the $-CH_3$ and -COOH terminated thiolates were 12.0 and 12.2 nm, respectively. Seven such uptake curves gave a standard deviation of 5% in the saturation response. From a calibration plot using toluene in ethanol solutions which covers the appropriate minimum wavelength range, we

determined a sensitivity factor for this probe, m, of 7300 nm / RIU. The refractive index of the solvent, ethanol, is 1.361 [40]. The RI of hexadecanethiol is estimated to be 1.463 (based on interpolation between reported values for other long-chain thiols [40]). The RI of the -COOH terminated hexadecanethiol is estimated to be the same (1.463), since hexadecane and hexadecanoic acid have the same RI [40]. The adsorbed thiolates are estimated to have the same index of refraction as the corresponding bulk thiol (see above). The value of l_d is estimated to be ~0.41 λ = 336 nm from Eq. 11. Using these values in either Eq. 6 or 7 together with the measured SPR responses gives thiolate film thickness of 2.71 and 2.75 nm for the methyl-terminated and -COOH terminated C_{16} thiols, respectively.

These measured thicknesses estimated using our formalism can be compared to the known adlayer thicknesses based on prior literature. The packing density of these thiolate monolayers are known to be 4.4 ± 0.2 x 10^{14} molecules / cm^2, based on prior radiolabelling [34] and structural studies [51]. This translates into a film which is 2.2 nm and 2.4 nm thick in the case of the methyl-terminated and -COOH terminated C_{16} thiols, respectively, based on the bulk density of the corresponding thiol. (These are 0.85 and 0.88 g / cm^3 [40], or 1.98x10^{21} and 1.84x10^{21} molecules/cm^3, respectively). The thicknesses estimated with our formalism from the measured SPR response agree within 15-23% of these expected thicknesses. They agree similarly well with other measurements of the film thicknesses [13, 27, 31].

PROTEIN ADSORPTION ON GOLD

Serum albumin proteins are known to adsorb to a variety of surfaces from aqueous solution to make a nearly dense-packed monolayer with a packing density of ~2.5 x 10^{-7} g / cm^2 [52]. Using the specific volume of such proteins (0.77 cm^3 / g [24, 45, 46]), this corresponds to an effective film thickness of pure protein of ~1.9 nm. The SPR

response to binding of the protein bovine serum albumin (BSA) to a clean gold surface on the planar probe is shown in Fig 2.9. The SPR response observed after adsorption for 800 s, where a saturation coverage is approached, is 8.0 nm in wavelength, with a reproducibility of ±0.3 nm.

The measured protein film thickness can be calculated from this response using Eqs. 5a, 6 or 7 and the following constants. Albumin proteins have an index of refraction of 1.57 (see above), compared to 1.334 for the aqueous buffer solution used to dose this protein. The SPR sensitivity factor over this range was determined by calibration to be 3100 nm / RIU (Fig 2.2). We get from Eq. 11 that $l_d = \sim 0.34\ \lambda = 233$ nm. The observed SPR response of 8.0 nm near saturation (i.e., after 800 s) thus gives an observed adlayer effective thickness of 1.3 nm using either Eq. 5a, 6 or 7. This converts to a protein concentration on the surface of 1.71×10^{-7} g / cm^2, using the specific volume of proteins (0.77 cm^3 / g). This is 32% below the estimated close-packed coverage of $\sim 2.5 \times 10^{-7}$ g / cm^2 (or thickness of ~1.8 nm), which itself is probably only accurate to ~30% [52]. A smaller value may be found because of the very short adsorption time used here (800 s), at which the adsorbed amount is still increasing very slowly (see Fig 2.9). Note that in studies of BSA adsorption on gold surfaces that were prefunctionalized with a wide variety of organic thiols, and on a variety of polymer surfaces, the most adsorptive surfaces showed very similar saturation coverages of BSA ($1.2\text{-}1.8 \times 10^{-7}$ g/cm^2) [23, 47].

Note that this measured thickness of 1.3 nm is the <u>average effective thickness of pure protein</u>, and it will be considerably less that the <u>optical thickness</u> measured in ellipsometry and some applications of SPR. That optical thickness generally includes a great deal of water that fills the voids between proteins [47, 53-55], which also results in a smaller "effective index of refraction" than that for pure proteins [47]. Quantitative determination of the amount of adsorbed protein (in ng / cm^2 or molecules / cm^2) is best done using the effective thickness of pure protein as we do, since its index of

refraction and specific volume are known constants, and since the amount of water included in the protein film will vary greatly between different proteins and different surfaces. If the optical thickness approaches l_d, however, one must include a factor f discussed above to take into account the volume fraction of water in the adlayer. At smaller thicknesses, the factor f can be ignored without loss in accuracy since the weight of adsorbed protein calculated from the SPR response will be independent of f, and therefore estimation of f is not necessary.

INFLUENCE OF A PREFUNCTIONALIZED FILM ON THE MEASURED RESPONSE

Equation 13, which is used for predicting thicknesses of bilayer structures, is supported by the experimental results shown in Fig 2.10. Here, the response of the fiber optic SPR system was measured as a function of changes in the index of refraction of bulk solutions in contact with the gold, both with and without an intermediate prefunctionalizing layer of decane thiolate. The thiolate adlayer has a negligible effect on the response slope, which is within 0.7% of 2466 nm / RIU in both cases, and therefore well within the reproducibility in determining the slope (~2%).

In applying Eq. 13, the decane thiolate can be considered as layer **b** (see Fig 2.6) with a thickness d_b of ~1.5 nm based on its packing density (4.4×10^{14} thiolates / cm^2, see above) and the bulk density of the corresponding thiol (0.844 g / cm^3, [40]). The bulk solution can be considered as an infinitely thick layer of **a**. In this case, Eq. 13 reduces to:

$$\Delta R = m \cdot (\Delta\eta) \cdot \exp(-2d_b / l_d), \qquad (14)$$

where $\Delta\eta$ refers to the change in index of refraction of **a**, and ΔR refers to the corresponding SPR response. While this thiolate would cause a change in the wavelength of the SPR minimum for any given bulk index of refraction above it, as described above,

this change is constant so that the slopes of the two curves in Fig 2.10 are the same. This is because the factor $\exp(-2d_b / l_d)$ in Eqs. 13 and 14 is nearly unity when d_b is so small compared to l_d (~300 nm).

In principle, the offset between the two curves can be predicted using Eq. 5a or 7, since the thiolate adlayer in this case is expected to be ~1.5 nm thick. However, these two curves were collected three days apart, and the fiber optic probe was moved through air to another beaker between these measurements. This causes baseline changes which are large relative to the shift due to thiolate adsorption, so this comparison is meaningless for these particular data.

PROPYLAMINE ADSORPTION ON -COOH PREFUNCTIONALIZED GOLD

The response of the planar SPR spectrometer to the binding of n-propylamine in ethanol solution to a -COOH prefunctionalized gold surface is shown in Fig 2.11. The gold surface was prefunctionalized with a -COOH terminated thiolate self-assembled monolayer as described above. A driving force for the observed reversible binding reaction of the protonated amine to this surface should be similar to the binding reaction occurring at the -COOH terminated resin surface of ion exchange columns, and that occurring in poly-L-lysine adsorption from aqueous solution [56, 57] and alkyl amine adsorption from the gas phase [58, 59] on a similar -COOH terminated thiolate self-assembled monolayers on gold. In solutions, the species are mostly ionic and solvated, and the cations are held to the anionic surface by weak electrostatic interactions:

$$\text{Au-S-(CH}_2)_n\text{-COO}^{-\cdots+}\text{H}_2\text{OEt} + \text{EtO}^{-\cdots+}\text{H}_3\text{N(CH}_2)_2\text{CH}_3\text{ (soln)} \rightleftharpoons$$

$$\text{Au-S-(CH}_2)_n\text{-COO}^{-\cdots+}\text{H}_3\text{N(CH}_2)_2\text{CH}_3 + 2\text{ EtOH(soln).} \qquad (15)$$

Basic conditions help drive this reaction [56], which may help explain why it is observed under the present conditions, but not with a lysine monomer in water at pH = 8.5 [56].

The saturation change in the SPR wavelength minimum in Fig 2.11 is ~0.68 nm (with a reproducibility within 0.04 nm based on other runs), and the calibration of this probe in this range gave a sensitivity factor of 7400 nm / RIU. From Eq. 5a, 6, 7 or 13, this saturation response gives a propylamine monolayer thickness of 0.63 nm using that $\eta_a = 1.387$ and $\eta_s = 1.361$ [40], and that $l_d = 358$ nm = 0.43 λ from Eq. 11. Using the bulk liquid density of this molecule (0.7173 g / cm^3), and assuming simple cubic packing in the bulk gives a size for propylamine of 0.52 nm (or the edge length of the "cubes"), very close to this measured monolayer thickness. The measured thickness corresponds to a coverage of propylamine of 4.6×10^{14} molecules / cm^2, again using the bulk density. The surface concentration of the $-COO^-$ headgroups is ~4.4×10^{14} / cm^2 (see above). One expects a 1:1 stoichiometry for the binding reaction, so the expected surface coverage at saturation is ~4.4×10^{14} molecules / cm^2. Again, the measured saturation coverage is very close to that expected.

DETECTION LIMITS

MEASURED DETECTION LIMITS FOR BULK INDICES OF REFRACTION

The detection limits for the planar system based on the scatter in the SPR wavelengths in Fig 2.2 (~0.01 nm) is about 3×10^{-6} RIU for the large time span of those measurements. Experiments with the flow cell wherein a quicker change in solution was accomplished using a chromatographic sampling valve showed that changes of <~0.007 nm in wavelength or 2×10^{-6} in RIUs could be detected easily when occurring within a few minutes time. With the SPR fiber-optics probe, a detection limit of <~0.03 nm in

wavelength or $1x10^{-5}$ in RIUs was found by stepwise additions of solute to the beaker of solution. The SPR detection limits for changes in bulk index of refraction depend on the noise and baseline drift in the wavelength of the SPR minimum, as well as the magnitude of m, the slope of wavelength versus RI. The noise can be reduced by signal averaging, but baseline drift can be a problem. Slow measurement generally degrades the detection limit, since baseline drift becomes increasingly problematic with increasing time. Comparable detection limits ($2x10^{-6}$ - $1x10^{-5}$ RIU) have been reported previously [14-16].

The detection limit of the planar device was not significantly affected by using a variety of different preparation methods for the gold films as long as their thickness was maintained close to 50 nm. The major factor influencing detection limits was baseline stability, which for the planar sensor was limited by temperature drift, we believe. The fiber optic probe was less stable, perhaps due to interactions of the probe fluid with the fiber / cladding interface. Mechanical stability of any index-matching fluid interfaces is also quite important. The thickness of the gold film influences the calibration slope, m, especially at high RI. Neither slope nor detection limit degrade significantly with time unless the surface becomes heavily contaminated.

DETECTION LIMITS FOR SURFACE CONCENTRATIONS AND ADLAYER THICKNESSES

For a single adlayer of uniform thickness, one can easily predict a detection limit for that thickness, d_{min}, based on the detection limit for changes in bulk index of refraction in the absence of that film. We showed above that our planar system has a detection limit of ~$2x10^{-6}$ RIU, which according to the above discussion can be set equal to $(\eta_a - \eta_s)\cdot(2d_{min} / l_d)$. Thus, the detection limit is $1x10^{-6}$ l_d divided by the RIU difference between the adsorbate and solution. For a typical value of l_d of 300 nm and $(\eta_a - \eta_s) = 0.1$, this gives a detection limit of 0.003 nm or 0.03 Å in average film thickness. Since a typical atom is ~ 2 Å in diameter, this corresponds to only ~1.5% of a

single atomic layer. As described above, the thickness detection limit can easily be converted to a coverage detection limit (molecules / cm^2) by multiplying by the bulk density of the adlayer in molecules / cm^3. If the sensor is precoated with another adlayer, these detection limits would be increased by the factor $\exp(2d_b / \lambda)$ as follows from the derivation of Eq. 13. This factor is nearly unity when d_b is very small compared to λ. If the molecule is exceedingly nonuniformly spread across the surface with severe clustering, this detection limit could be increased as well, as also follows from the discussion above.

A detection limit of ~6% of a monolayer of propylamine, or <0.04 nm average film thickness, can be seen from the raw data points in Fig 2.11. In this case the difference in index of refraction between the adsorbate and the solution is only 0.026 RIU. For a more typical difference of 0.1 RIU, this corresponds to a detection limit of ~0.01 nm in film thickness. For protein adsorption in aqueous buffer, where the difference is ~0.24 RIU, it corresponds to a detection limit of 0.004 nm thickness, or 0.5 ng / cm^2, or ~2×10^{-3} monolayers. Data smoothing would improve this considerably, but at the cost of poorer time resolution. (The response time to a step change in bulk index of refraction was < 2 seconds under the conditions of Fig 2.11.)

Kunz et al. [14] discussed the detection limits of planar and fiber-optic SPR devices in terms of bulk index of refraction changes and bulk analyte concentrations, but not in terms of adsorbate coverages or thicknesses. From Maxwell's equations, Yeatman [15] estimated a detection limit of 3×10^{-6} RIU, or 3 nm thickness for an adsorbate whose index of refraction was 0.1 RIU different than the surrounding medium, for certain characteristics of the SPR spectrometer. A detection limit of ~0.1 nm thickness was estimated by Häussling et al. [17]. We show thickness detection limits significantly better than either of these estimates, although in the former case it may be just an interpretation difference.

DETECTION LIMITS FOR BULK CONCENTRATIONS OF ANALYTES IN SOLUTIONS

In adsorption-based chemical sensors or in immunoassays, one is generally more interested in the detection limits in terms of the bulk solution concentration of the analyte, antigen or antibody. Connecting to these units from a detection limit in units of adlayer effective thickness or surface coverage is non-trivial, since the conversion factor depends on the equilibrium constant, K_{eq}, for binding of the analyte to the sites on the SPR surface, which may not be known. If it is known, then one can use its definition to estimate the detection limit in units of bulk concentration. Consider the simplest binding reaction:

$$A + S \rightleftharpoons A\text{-}S, \tag{16}$$

where A is the solution phase analyte, and S is its surface site in a prefunctionalized adlayer on the sensor surface. The equilibrium constant is defined as:

$$K_{eq} = [A\text{-}S] / \{[A][S]\} = \theta / \{(1-\theta).[A]\}, \tag{17}$$

where [A] is the concentration of the analyte, and θ is the fraction of its sites that are occupied at equilibrium. If the adlayer system is well designed, θ will be small at the detection limits, so that $(1-\theta)$ is ~1.00. In this case, $[A] = \theta / K_{eq}$. Thus, the detection limit for the bulk concentration of A will just be the detection limit for the fraction of its bound sites divided by K_{eq}. The detection limit for the fraction of its bound sites can be estimated as described above: First estimate the coverage detection limit (molecules / cm^2), and then divide that number by the number of binding sites per cm^2 in the prefunctionalized layer. The number of binding sites per cm^2 can be estimated by measuring the saturation coverage at high concentration of A using the SPR response and the procedures outlined above.

While one often knows the equilibrium constant for the analogous reaction where the sites S are dissolved in solution, the way in which S is immobilized onto the

sensor surface may alter the value of K_{eq}. It is best to make separate measurements of the equilibrium constant for the surface-immobilized S, which can be done by using the SPR itself to measure θ versus [A].

TEMPERATURE SENSITIVITY AND ITS EFFECT ON DETECTION LIMITS

To achieve Figs. 2.2 and 2.3 and the above-stated detection limits, we controlled the temperatures of the solutions to some extent, since we found that the baseline drifted sensitively with changing temperature. The SPR response to changes in temperature for water solutions of fixed composition (pure water) are plotted in Fig 2.12, for both the planar and fiber-optic spectrometers. As can be seen, the wavelength minimum decreases by about 0.29 and 0.16 nm per degree Kelvin for the planar and fiber-optic probes, respectively. The refractive index of water is known to decrease by ~$8x10^{-5}$ RIU per degree K near room temperature [40]. Ignoring the changes associated with temperature effects on the solid parts of the SPR probe, and using the sensitivity factors of 3100 and 1600 nm / RIU, respectively, from above, one would predict a slope of ($-8x10^{-5}$ RIU / K) x (3100 nm / RIU) = -0.25 nm / K for the planar probe, and -0.13 nm / K for the fiber-optic probe, very close to the observed slopes. This suggests that the dominant effect of temperature is in its effect on the refractive index of the liquid solution. This is expected, since the index of refraction of the liquid (water here) will typically change much more sensitively with temperature than that for solids. From the above discussion, it is obvious that the temperature of the liquid must be controlled to better than $2x10^{-2}$ K over the time scale of the two measurements to achieve detection limits of $2x10^{-6}$ RIU or ~0.003 nm in thickness.

CONCLUSIONS

We have outlined here a procedure for estimating structural parameters of adsorbed layers (thickness, surface concentration, or fractional coverage) based on measured SPR responses. All that is needed is a knowledge, through calibration, of the SPR response versus bulk index of refraction, and absolute thicknesses can be estimated using this method with reasonable absolute accuracy and very high relative accuracy and precision. That is, no calibration versus adlayer thickness is required with this method, although such a calibration would improve the absolute accuracy. In this method, the index of refraction of the adlayer must be assumed, but we outline here ways for estimating it. The method is tested on several thin adlayers of known thickness, and an absolute accuracy within ~25% obtained. The accuracy is limited by inaccurate knowledge of the penetration depth of the SPR evanescent field and the index of refraction of the adlayer. It is a useful quantitative method for adlayers up to >200 nm in thickness. It is shown that, if the SPR is capable of detecting changes in bulk index of refraction of $2x10^{-6}$ RIU, as is the planar system described here, then the detection limit for adsorbed layers is a thickness of only ~$3x10^{-4}$ nm, divided by the difference in index of refraction between the adsorbate and the dosing solution. This typically gives a limit below 10^{-2} nm, or 0.1 Å average thickness. The dynamic range is therefore rather large, from ~10^{-2} to >200 nm in average film thickness.

Methods are also presented for correcting the quantitative estimates of adsorbate coverage or adlayer thickness for the presence of a prefunctionalizing adlayers (for example, due to a receptor adlayer on the sensor), and for nonuniformities in the thickness of the adlayer across the SPR probe area.

NOTES TO CHAPTER 3

[1] I. Lundstrom, Biosensors and Bioelectronics 9 (1994) 725.

[2] W. Lukosz, Biosensors and Bioelectronics 12 (1997) 175.

[3] J. Melendez, R. Carr, D. Bartholomew and H. Taneja, Sixth International Conference on Chemical Sensors (Washington, D.C.), Development of a surface plasmon resonance sensor for commercial applications (1996).

[4] J. Melendez, R. Carr, D. U. Bartholomew, K. Kukanskis, J. Elkind, S. Yee, C. Furlong and R. Woodbury, Sensors and Actuators B 35 (1996) 212.

[5] R. P. H. Kooyman, H. Kolkman, J. V. Gent and J. Greve, Anal. Chim. Acta 213 (1988) 35.

[6] E. Stenberg, B. Persson, H. Roos and C. Urbaniczky, J. Coll. Interface Sci. 143 (1991) 513.

[7] D. J. O'Shannessy, Current Opinion in Biotechnology 5 (1994) 65.

[8] B. Liedberg, I. Lundstrom and E. Stenberg, Sensors and Actuators B 11 (1993) 63.

[9] B. Liedberg, C. Nylander and I. Lundström, Biosensors and Bioelectronics 10 (1995).

[10] M. Mrksich, J. R. Grunwell and G. M. Whitesides, J. Am. Chem. Soc. 117 (1995) 12009.

[11] M. Mrksich, G. B. Sigal and G. M. Whitesides, Langmuir 11 (1995) 4383.

[12] M. Mrksich and G. M. Whitesides, Annu. Rev. Biophysics Biomol. Struct. 25 (1996) 55.

[13] K. A. Peterlinz and R. Georgiadis, Langmuir 12 (1996) 4731.

[14] U. Kunz, A. Katerkamp, R. Renneberg, F. Spener and K. Cammann, Sensors and Actuators B 32 (1996) 149.

[15] E. M. Yeatman, Biosensors and Bioelectronics 11 (1996) 635.

[16] Pharmacia, *BIAcore*^{TM} *System Manual*, Pharmacia Biosensor AB, Scientific Information, Upsala, Sweden 1991.

[17] L. Haeussling, H. Ringsdorf, F.-J. Schmitt and W. Knoll, Langmuir 7 (1991) 1837.

[18] J. R. Sambles, G. W. Bradbery and F. Yang, Contemporary Physics 32 (1991) 173.

[19] A. S. Barker, Physical Review B 8 (1973) 5418.

[20] W. Lukosz, Biosensors and Bioelectronics 6 (1991) 215.

[21] D. G. Hanken and R. M. Corn, Anal. Chem. 67 (1995) 3767.

[22] V. I. Silin, G. A. Balchytis and V. A. Yakovlev, Optics Communications 97 (1993) 19.

[23] V. Silin, H. Weetall and D. J. Vanderah, Journal of Colloid and Interface Science 185 (1997) 94.

[24] S. Sjölander and C. Urbaniczky, Anal. Chem. 63 (1991) 2338.

[25] K. A. Peterlinz and R. Georgiadis, Optics Commun. 130 (1996) 260.

[26] K. S. Johnston, S. R. Karlsen, C. C. Jung and S. S. Yee, Materials Chemistry and Physics 42 (1995) 242.

[27] T. T. Ehler, N. Malmberg and L. J. Noe, J. Phys. Chem. B 101 (1997) 1268.

[28] R. C. Jorgenson and S. S. Yee, Sensors and Actuators (Chemical) B12 (1993) 213.

[29] C. D. Bain, J. Evall and G. M. Whitesides, J. Am. Chem. Soc. 111 (1989) 7155.

[30] C. D. Bain and G. M. Whitesides, J. Am. Chem. Soc. 111 (1989) 7164.

[31] C. D. Bain, E. B. Troughton, Y.-T. Tao, J. Evall, G. M. Whitesides and R. G. Nuzzo, J. Am. Chem. Soc. 111 (1989) 321.

[32] D. L. Allara, Biosensors & Bioelectronics 10 (1995) 771.

[33] R. F. DeBono, G. D. Loucks, D. Della Manna and U. J. Krull, Can. J. Chem. 74 (1996) 677.

[34] L. H. Dubois and R. G. Nuzzo, Annu. Rev. Phys. Chem. 43 (1992) 437.

[35] N. O. Perrson, K. Uvdal, B. Liegberg and M. Hellsten, Progress in Colloid & Polymer Science 88 (1992) 100.

[36] K. Kurosawa, R. M. Pierce, S. Ushioda and J. C. Hemminger, Phys. Rev. B 33 (1986) 789.

[37] J. J. Ramsden, S.-Y. Li, J. E. Prenosil and E. Heinzle, Biotechnology and Bioengineering 43 (1994) 939.

[38] P. Schuck, Biophysical Journal 70 (1996) 1230.

[39] O. S. Heavens, Optical Properties of Thin Films, Dover, N.Y. 1955.

[40] D. R. Lide, ed., *Handbook of Chemistry and Physics, 71st ed.*, CRC Press, Inc., Boston 1990.

[41] S. H. Armstrong, Jr., M. J. E. Budka, K. C. Morrison and M. Hasson, J. Am. Chem. Soc. 69 (1947) 1747.

[42] I. N. Levine, Physical Chemistry, McGraw-Hill, New York 1988.

[43] T. L. McMeekin, M. L. Groves and N. J. Hipp, Adv. Chemistry series 44 (1964) 54.

[44] P. Doty and E. P. Geiduschek, in H. N. a. K. Bailey (Ed.): The Proteins, Academic Press, N. Y. 1953, p. 1A.

[45] J. E. Darnell, H. Lodish and D. Baltimore, Molecular Cell Biology, Scientific American Books, New York 1990.

[46] T. E. Leslie and T. H. Lilley, Biopolymers 24 (1985) 695.

[47] C.-G. Gölander and E. Kiss, Journal of Colloid and Interface Science 121 (1988) 240.

[48] R. A. Innes and J. R. Sambles, J. Phys. F: Met. Phys. 17 (1987) 277.

[49] E. D. Palik, ed., *Handbook of Optical Contants of Solids*, Academic Press, Orlando, Florida 1985.

[50] C. E. H. Berger, R. P. H. Kooyman and J. Greve, Rev. Sci. Instrum. 65 (1994) 2829.

[51] J. Li, K. S. Liang, N. I. Camillone, T. Y. B. Leung and G. Scoles, J. Chem. Phys. 102 (1995) 5012.

[52] B. D. Fair and A. M. Jamieson, Journal of Colloid and Interface Science 77 (1980) 525.

[53] J. W. Corsel, G. M. Willems, J. M. M. Kop, P. A. Cuypers and W. T. Hermens, Journal of Colloid and Interface Science 111 (1986) 544.

[54] J.-E. Sundgren, P. Bodö, B. Ivarsson and I. Lundström, Journal of Colloid and Interface Science 113 (1986) 530.

[55] K. L. Prime and G. M. Whitesides, J. Am. Chem. Soc. 115 (1993) 10714.

[56] C. E. Jordan and R. M. Corn, Anal. Chem. 69 (1997) 1449.

[57] C. E. Jordon, B. L. Frey, S. Kornguth and R. M. Corn, Langmuir 10 (1994) 3642.

[58] H. C. Yang, D. L. Dermody, C. Xu, A. J. Ricco and R. M. Crooks, Langmuir 12 (1996) 726.

[59] K. Matsuura, Y. Ebara and Y. Okahata, Thin Solid Films 273 (1996) 61.

FIGURES FOR CHAPTER 3

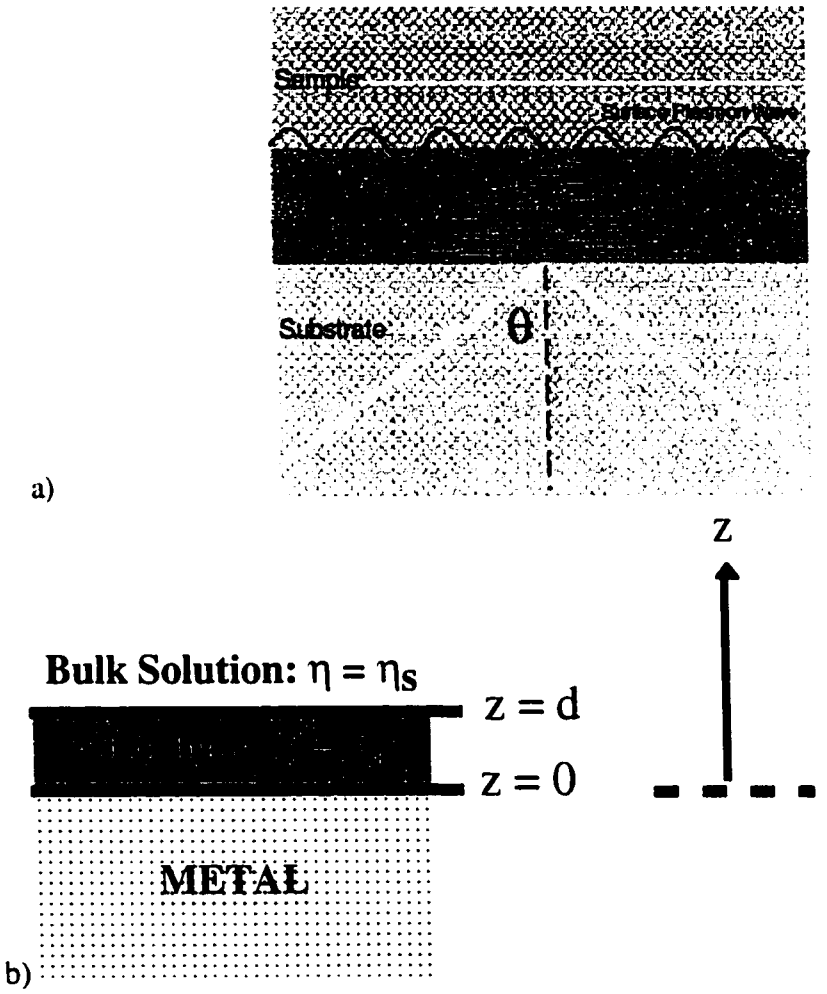


Figure 3.1 (a) Schematic diagram of the working interface of an SPR spectrometer and the SPR effect. The working interface involves a thin metal film on a transparent substrate, in contact with a medium to be probed, usually a liquid solution. (b). Schematic diagram of a bilayer structure involving an adsorbate a of thickness d and refractive index η_a directly on the metal probe surface, above which is solution s, of refractive index η_s.

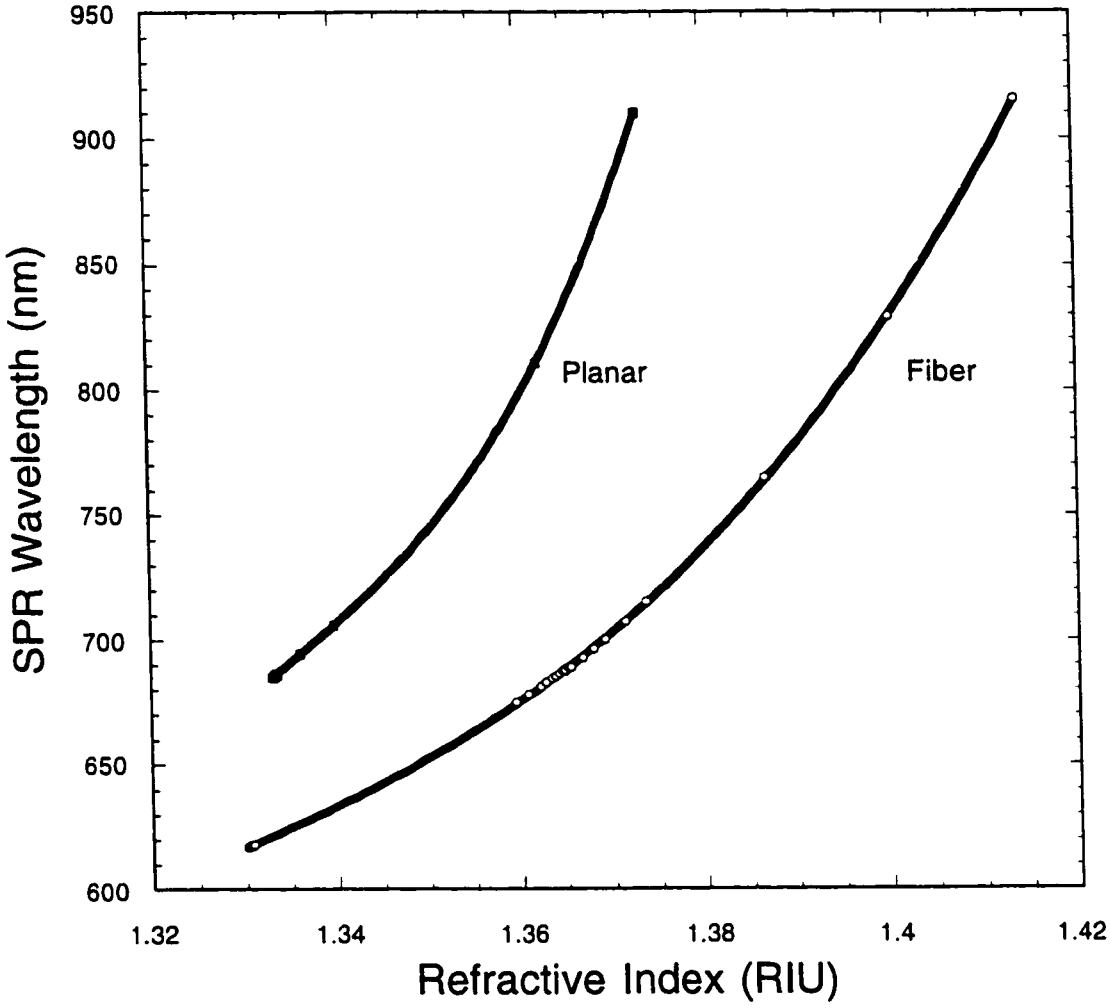


Figure 3.2 Measured SPR response (wavelength of minimum in the reflected light intensity) versus the bulk index of refraction (η) of solutions in contact with the gold SPR probe surface, for both the planar and fiber-optic SPR spectrometers, shown over a broad range of η.

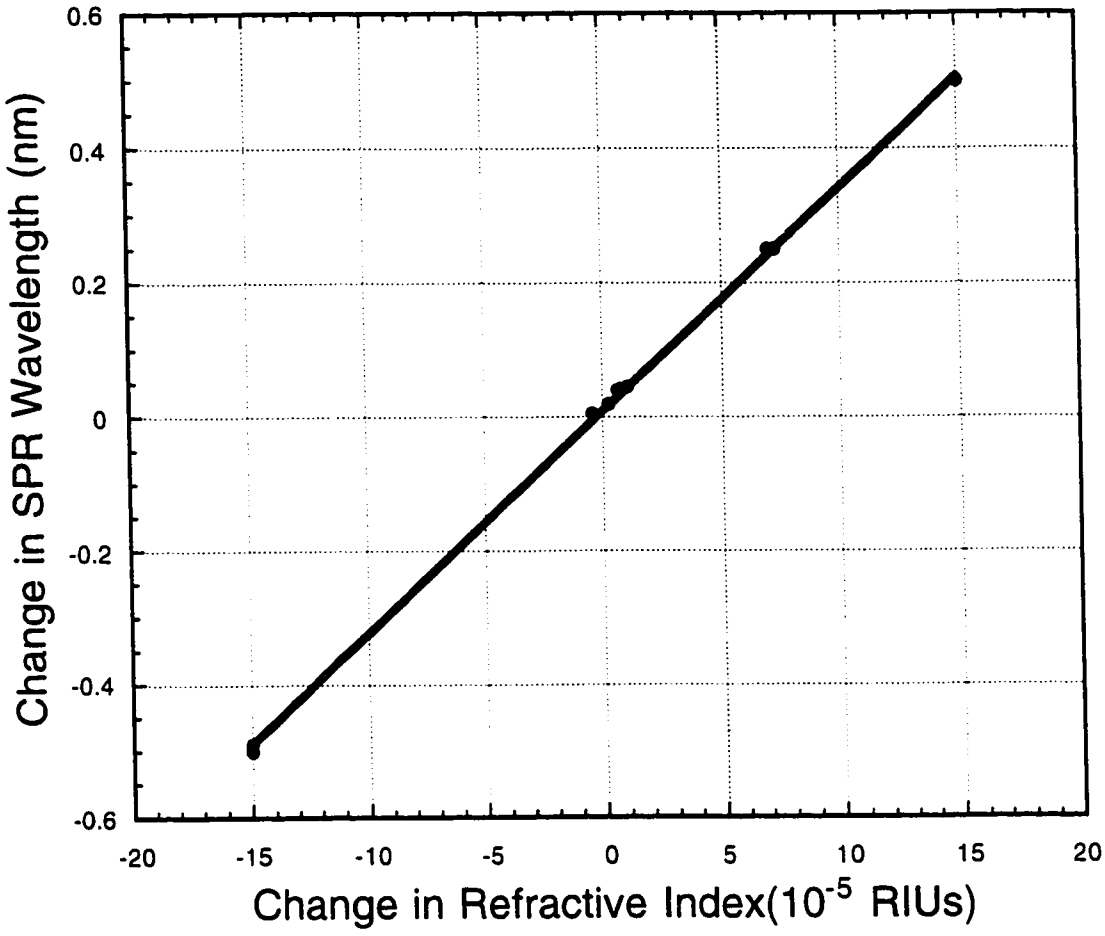


Figure 3.3 Same as Fig 3.2, except over a narrow range of η near that for water (1.330), and only for the planar system. Slope = 3450 nm / RIU.

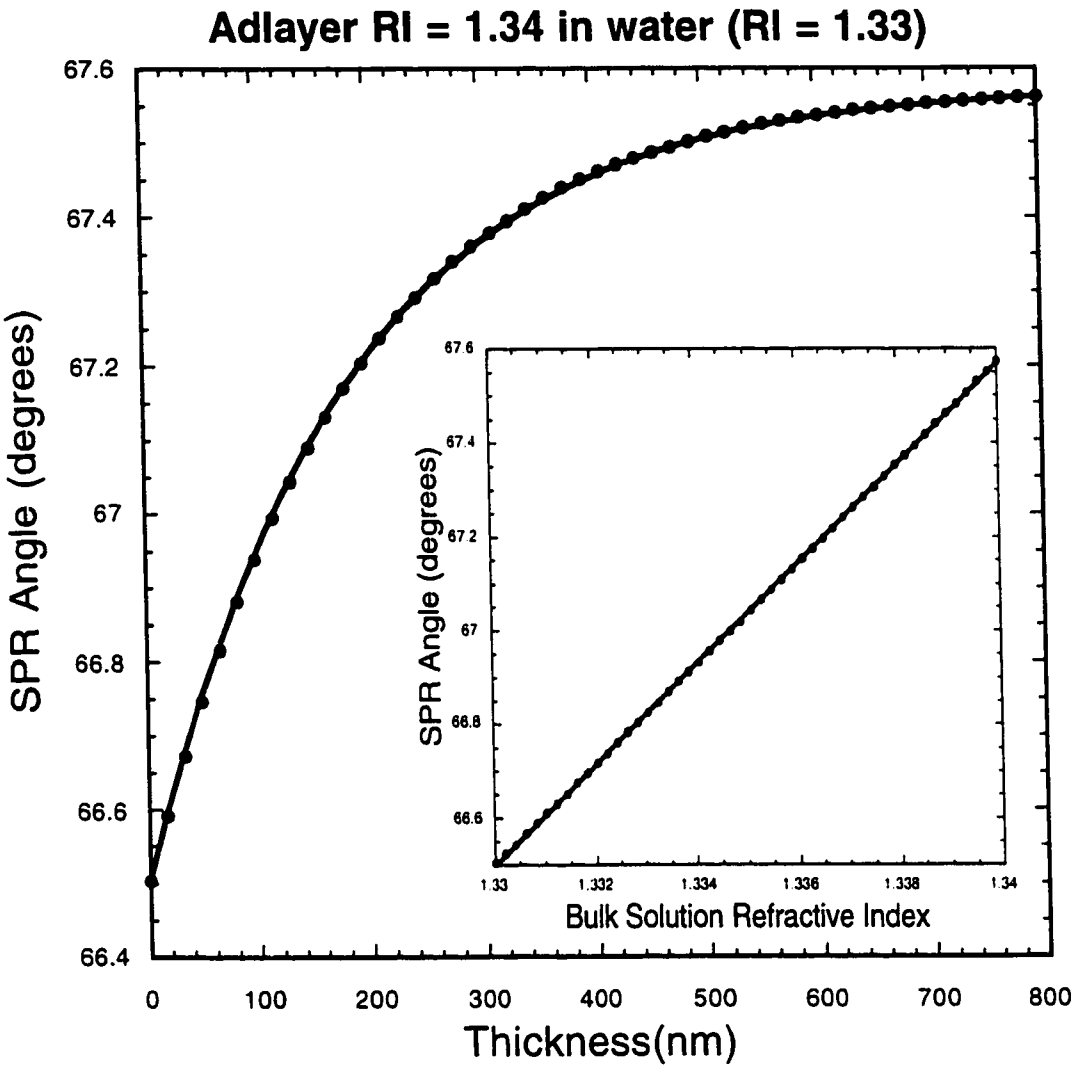


Figure 3.4 Calculated SPR response versus adlayer thickness, d, for the bilayer structure of Fig 3.1b, for the special case where $\eta_a = 1.330$, $\eta_s = 1.340$ and $\lambda = 825$ nm. Dots: Full calculation with Maxwell's equations and thin-film optical model. Solid curve: Calculation using Eq. 5a, letting l_d be 368 nm to achieve the best fit to the dotted curve. INSERT: Calculated SPR response versus bulk index of refraction of a pure, homogeneous solution in contact with the metal surface, for the special case where $\lambda = 825$ nm. Dots: Full calculation with Maxwell's equations and the same thin-film optical model. Solid curve: The best fit of Eq. 1a to the dotted curve in the range from $\eta = 1.330$ to 1.340, giving m = 107 degrees per RIU.

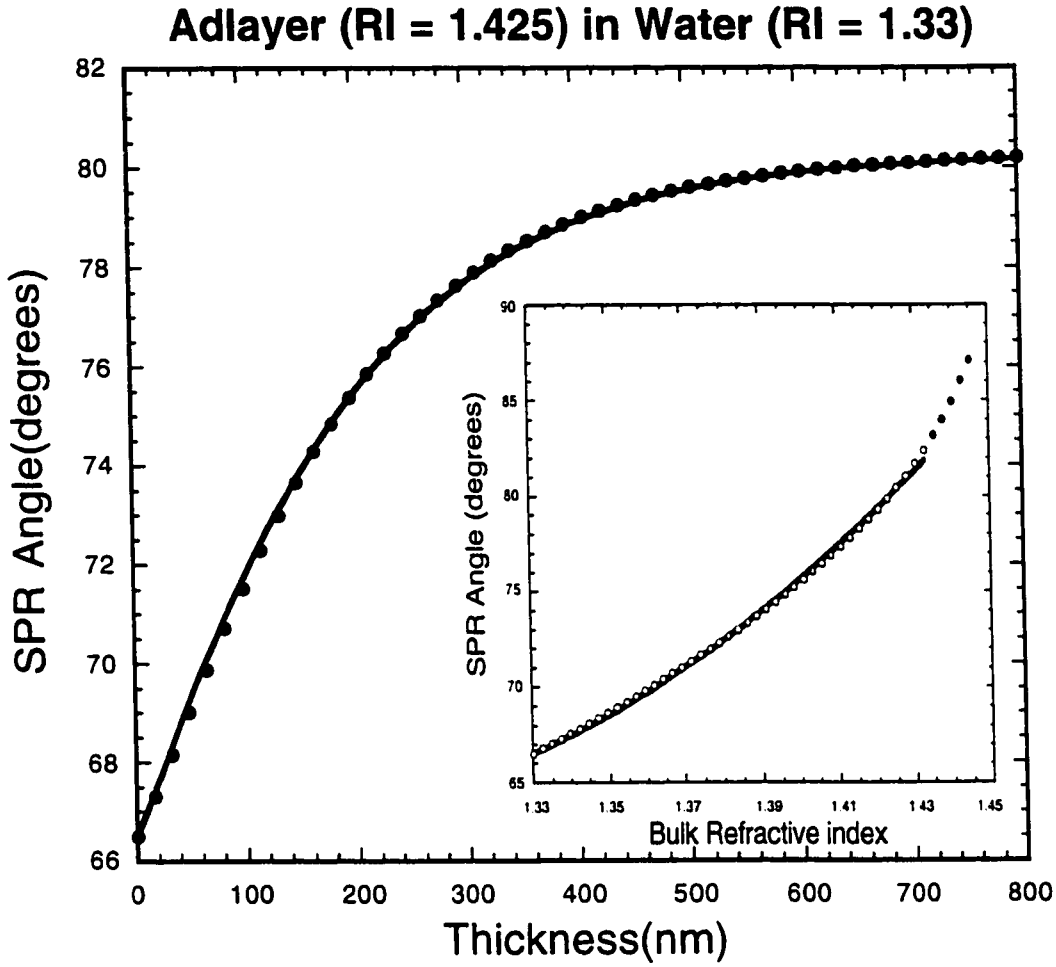


Figure 3. 5 Calculated SPR response versus adlayer thickness, d, for the bilayer structure of Fig 3.1b, for the special case where η_a = 1.330, η_s = 1.425 and λ = 825 nm. Dots: Full calculation with Maxwell's equations and thin-film optical model. Solid curve: Calculation using Eq. 5b, letting l_d be 307 nm to achieve the best fit to the dotted curve. INSERT: Calculated SPR response versus bulk index of refraction of a pure, homogeneous solution in contact with the metal surface, for the special case where λ = 825 nm. Dots: Full calculation with Maxwell's equations and the same thin-film optical model. Solid curve: The best fit of Eq. 2b to the dotted curve in the range from η = 1.330 to 1.433, giving m_1 =89.7 degrees per RIU and m_2 = 980 degrees per RIU^2.

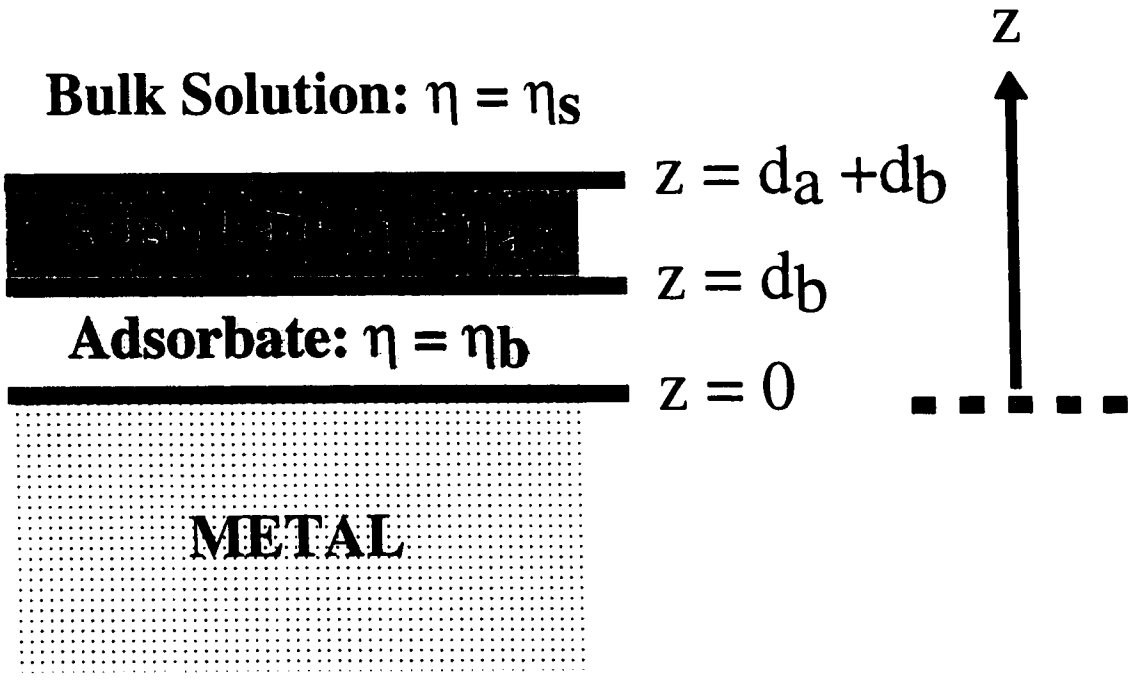


Figure 3.6 Schematic diagram of the trilayer structure involving a preadsorbate layer **b** of thickness d_b directly functionalizing the metal probe surface, above which is a second adsorbate layer **a** of thickness d_a and finally the bulk solution **s**.

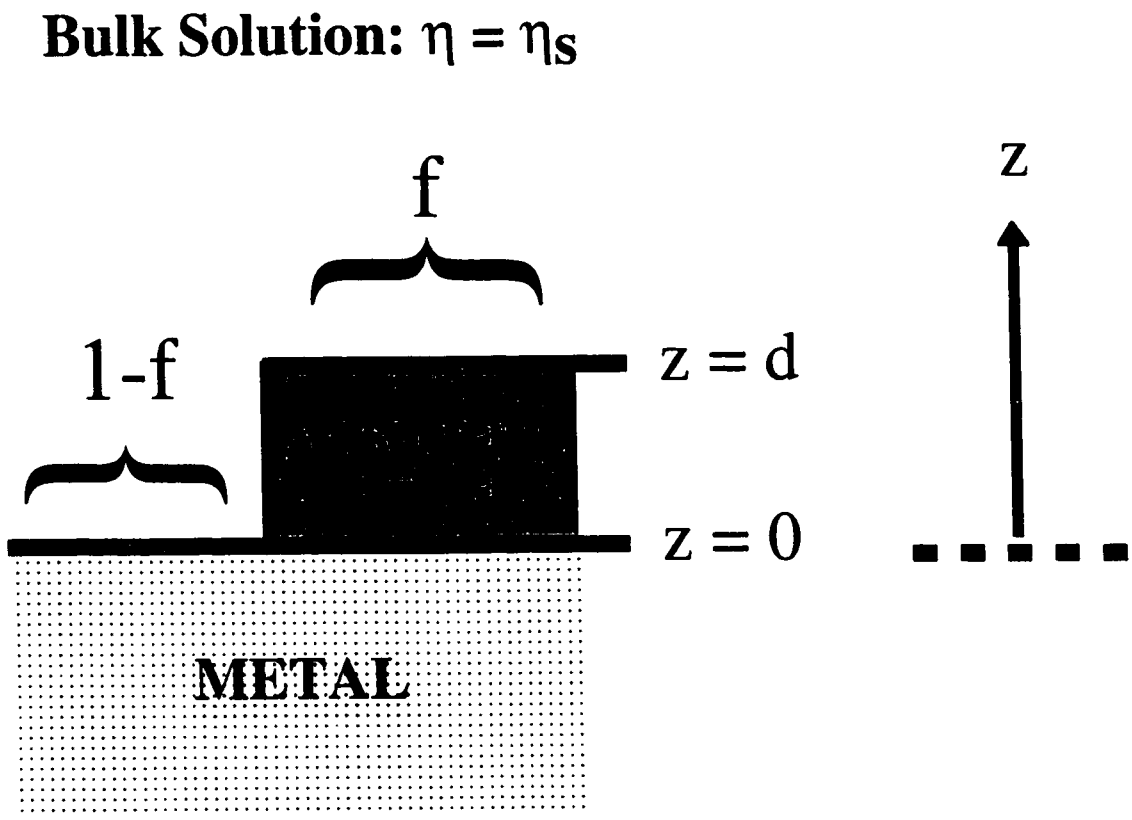


Figure 3.7 Schematic diagram of a bilayer structure with islanding, involving an adsorbate **a** of thickness d and refractive index η_a directly on the metal probe surface, but only covering a fraction f of its surface, above which is solution s, of refractive index η_s.

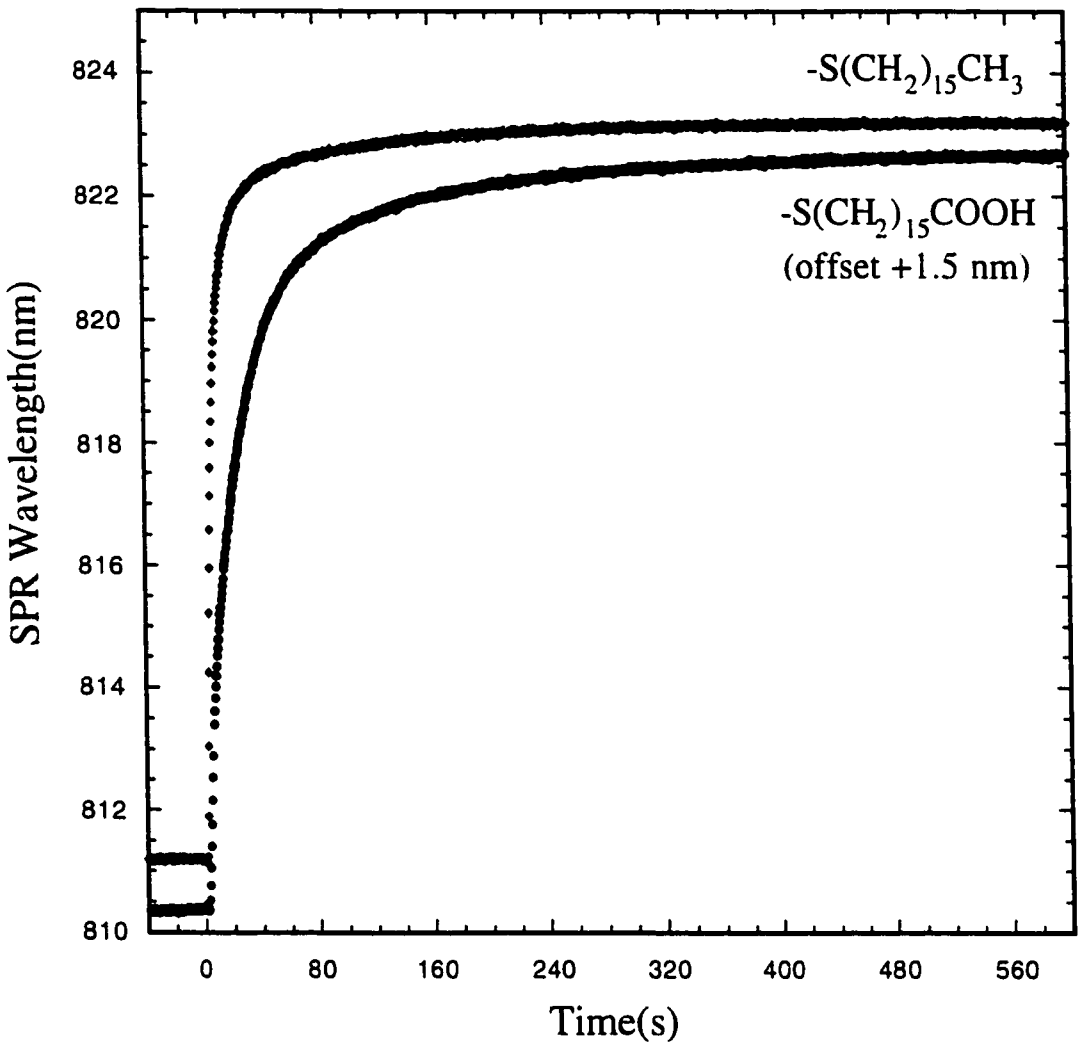


Figure 3.8 Response of the planar SPR to the adsorption of two thiolates, both from dilute ethanol solutions, at room temperature. Before time zero, pure ethanol was passing over the clean gold surface. At time zero, the solution was switched to one containing the thiol of interest in the following concentration: (a) 1.0 mM $HS(CH_2)_{15}CH_3$, and (b) 0.1 mM $HS(CH_2)_{15}COOH$.

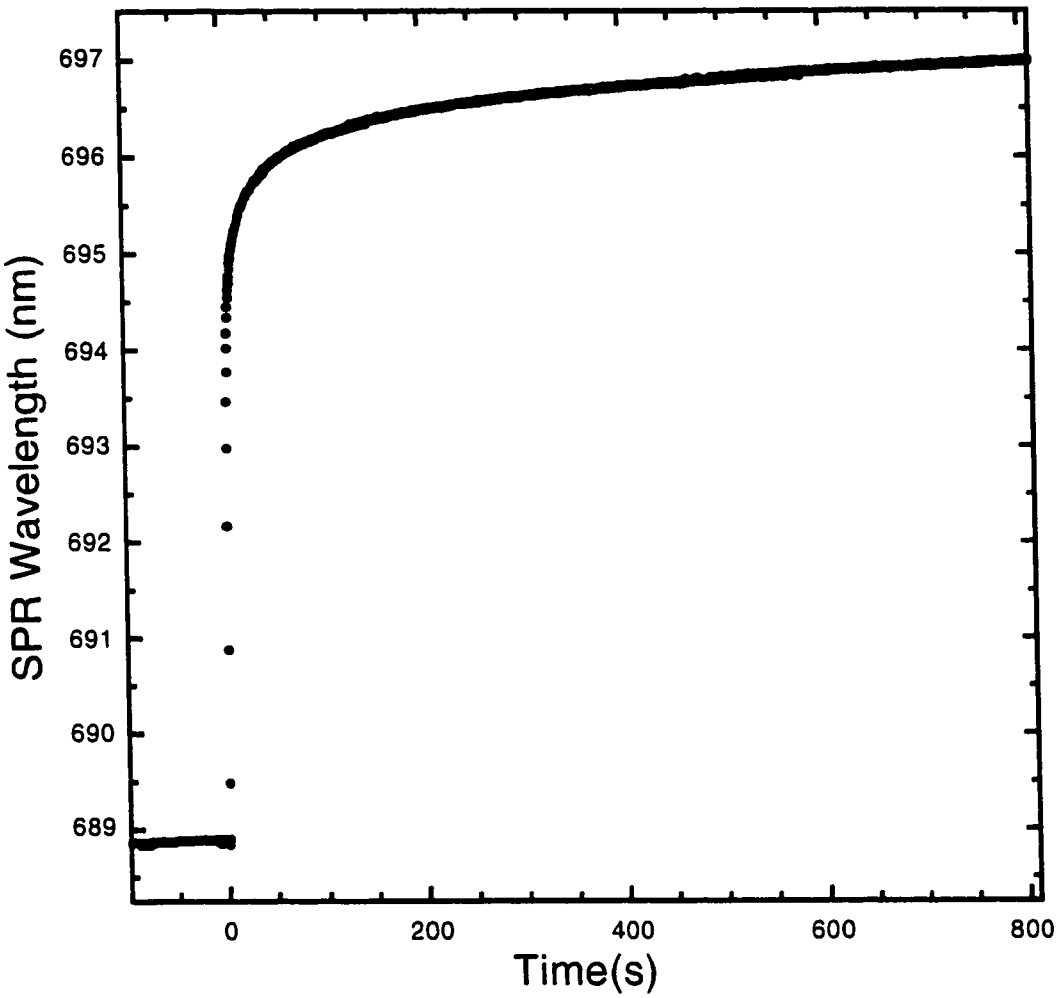


Figure 3.9 Response of the planar SPR to the adsorption of the protein BSA (bovine serum albumin) from dilute aqueous buffer solution (1 mg / ml in PBS buffer at pH 7.0), onto the clean gold surface of the SPR probe at room temperature.

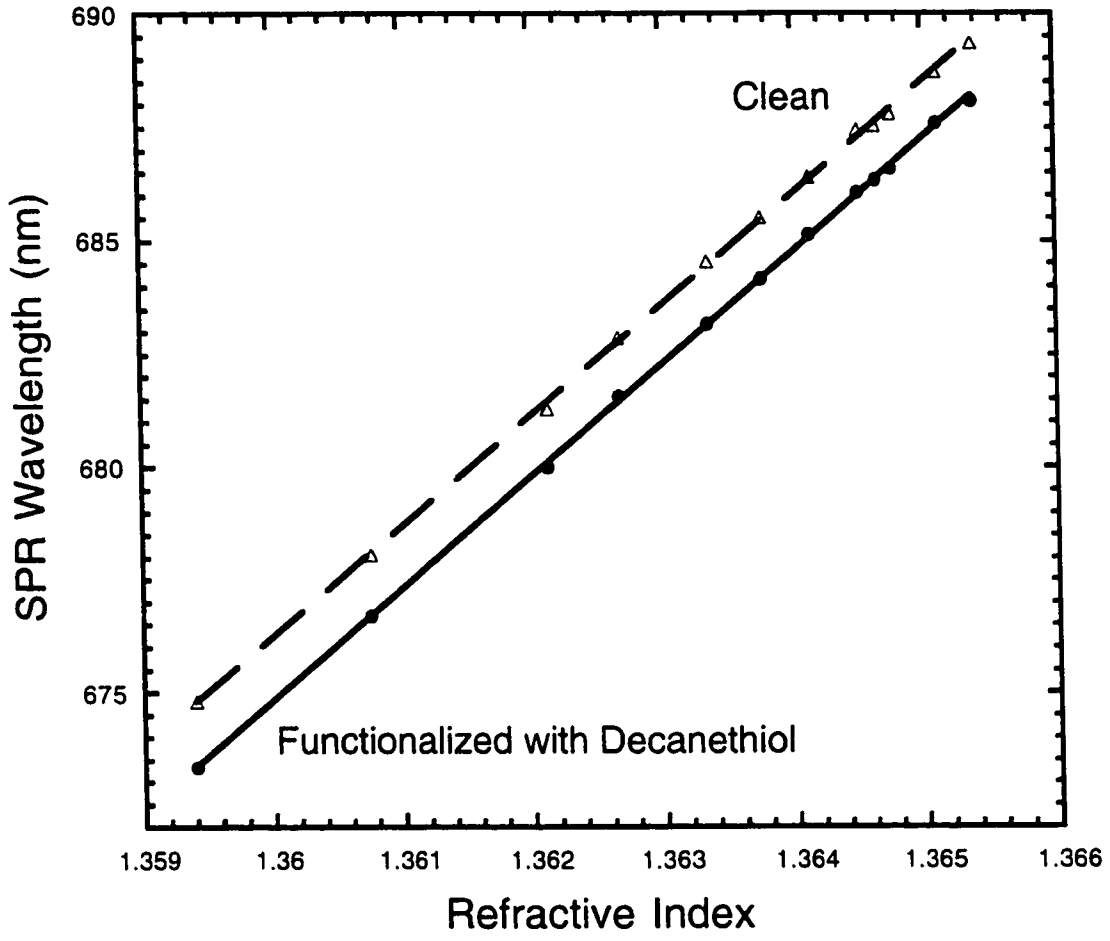


Figure 3.10 Measured SPR response (wavelength of minimum in the reflected light intensity) versus the bulk index of refraction (η) of solutions in contact with the gold SPR fiber-optic probe surface, with and without prefunctionalization with a decane thiolate adlayer (like those in Fig 2.8). Room temperature.

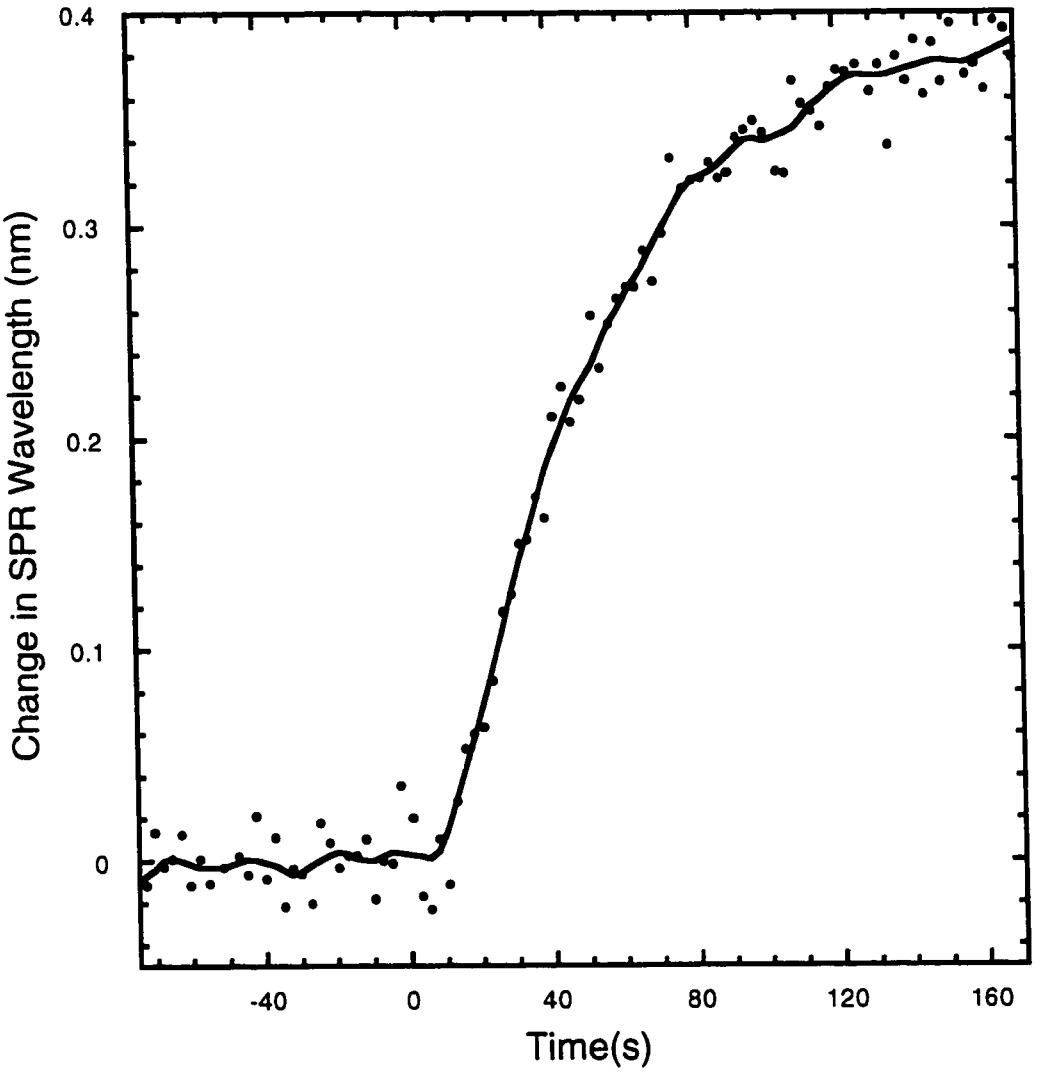


Figure 3.11 Response of the planar SPR, prefunctionalized with -$S(CH_2)_{16}COOH$ as in Fig 3.8, to the adsorption of propylamine from dilute ethanol solution (2.4 mM) at room temperature.

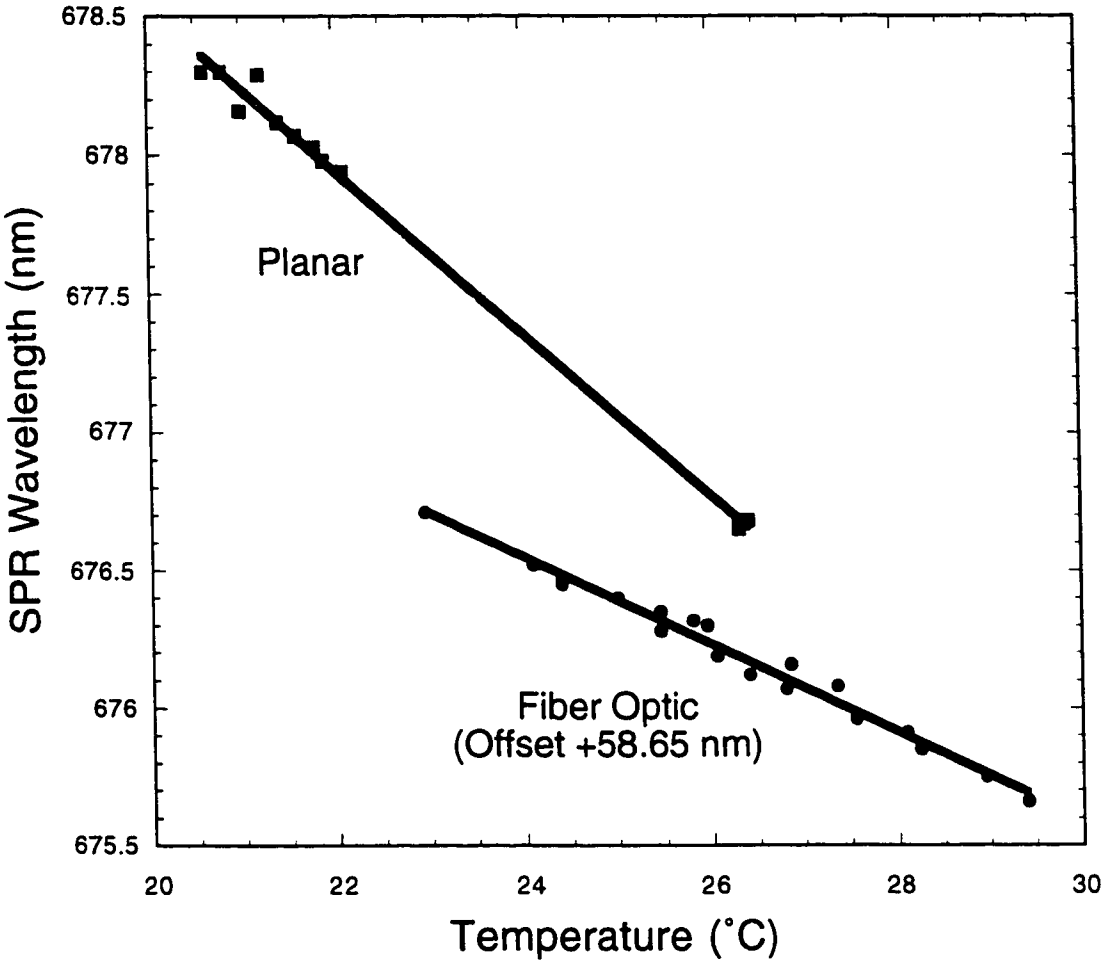


Figure 3.12 Temperature dependence of the SPR response in water, for both the planar (squares) and fiber-optic (circles) SPR sensors. The fiber-optic data have been uniformly shifted vertically to bring onto the same scale.

CHAPTER 4: DEFINING AND MEASURING STICKING PROBABILITES IN LIQUIDS

INTRODUCTION

The most fundamental way to express an adsorption rate constant is in terms of a sticking probability (S). It is defined as the probability that the molecule adsorb upon its collision with the surface. It is just the rate of adsorption divided by the collision frequency with the surface:

$$S = \frac{R_{ads}}{J_s}, \tag{1}$$

where R_{ads} is the rate of adsorption per unit area and J_s is the collision frequency, i.e. the number of molecules which collide with the surface per second per unit area, or the flux to the surface. Measurements of its value versus coverage, partial pressure, temperature and other parameters have allowed surface scientists to extract many details about the kinetics, activation energy, mechanism and dynamics of adsorption processes from the gas phase. Yet sticking probability is rarely mentioned in studies which address the kinetics of adsorption from liquid solutions, and when it has been mentioned [1-3], we feel it has not been accurately treated.

The difficulty in liquids is in knowing how to properly calculate the collision frequency of a liquid-phase species with the surface. This is simple in gas phase adsorption where J_s is defined as

$$J_s = n^* \sqrt{\frac{kT}{2\pi m}}, \tag{2}$$

where k is Boltzmann's constant, T is the temperature, m is the mass of adsorbates and n* is the number of molecules per unit volume or the concentration of adsorbates in the gas phase. Since the pressure is usually uniform throughout the vacuum chamber, so is n*. In liquids, this is not the case as concentration gradients usually form when measuring adsorption kinetics, so depletion and diffusion need to be considered.

A good way to estimate if the formation of such a concentration gradient will be significant in a transient adsorption experiment when a surface is suddenly exposed to a liquid solution containing a molecule that will adsorb is to compare the characteristic diffusion time, τ_{diff}, with the observed half-life ($t_{1/2}$, the time taken to reach half the saturation coverage) of adsorption [4]. To do this, first a characteristic diffusion distance, d_{diff}, is defined as:

$$d_{diff} = \frac{\frac{\theta_1}{2}}{C_{bulk}}, \tag{3}$$

where $\theta_{1/2}$ is the number of molecules / cm^2 at half saturation coverage, and C_{bulk} is the concentration of the bulk solution. Essentially, this is the distance away from the surface that the solution would be depleted if a half of a monolayer instaneously adsorbed to that surface (as illustrated in Fig. 4.1). The average time it takes for a molecule on the edge of this depletion layer to reach the surface, τ_{diff}, can be calculated using

$$\tau_{diff} = \frac{d_{diff}^2}{D}, \tag{4}$$

where D is the diffusion constant of the adsorbate through the liquid. If τ_{diff} is much greater than $t_{1/2}$ of adsorption then a strong concentration gradient will form. Conversely, if τ is much, much less than the half-life ($t_{1/2}$) of adsorption, depletion will most likely be negligible. If τ and the $t_{1/2}$ are similar, some gradient should form.

In the literature a vast number of studies have analyzed adsorption in liquids. The majority of these kinetic analyses have been done by fitting the adsorbate coverage versus time to one of three equations. One, a simple first order Langmuir model, is given as

$$\theta = \theta_{sat}(1 - \exp[-kt]),$$

where θ is the adsorbate coverage, θ_{sat} is the its saturation coverage, t is time and k is the rate constant for adsorption [5-7]. This model does not take depletion of adsorbate from the solution into account and therefore should not be applied unless it is known that depletion is negligible. A second model, the diffusion limited Langmuir model, gives:

$$\theta = \theta_{sat}\left(1 - \exp\left(-\sqrt{\frac{t}{\tau}}\right)\right), \tag{5}$$

and incorporates D into the time constant, τ [4,8,9]. However, a careful inspection of the derivation of this equation [4] reveals that it is only applicable in cases if the sticking probability is always unity and where the saturation coverage, θ_{sat}, is tiny so that an adsorbate will not encounter a filled site when it approaches the surface. Therefore, it is only valid in cases where the initial sticking probability is unity and where much less than a monolayer adsorbs at equilibrium or during the early stages of adsorption.

The third, somewhat related model is based on purely diffusion controlled adsorption [2,3,10-15]. In this model, the solution adjacent to the surface is considered completely depleted in the solute which adsorbs, and the concentration gradient drives its diffusion from the bulk toward the surface. Therefore, the rate of adsorption is proportional to rate of diffusion giving

$$\theta = C_{bulk}\left(\frac{Dt}{2}\right)^{\frac{1}{2}}. \tag{6}$$

This model assumes the sticking probability, S, is always 1.0, and thus can only be valid at the initial stage of adsorption for cases where the surface saturates with adsorbate. In order to extract sticking probabilities from adsorption transients, some authors [3,13] have assumed that a similar equation which incorporates a non-unit sticking probability is valid.

$$\theta = SC_{bulk}(\frac{Dt}{2})^{\frac{1}{2}}. \tag{7}$$

We show below that this is a very poor assumption and results in S values which are far too large.

A more recent discussion of S was approached in conjunction with a modified Kisliuk Model which incorporates S into the rates [1]. However, the authors were unable to determine S values, since they could not determine the flux to the surface.

One might be tempted to try to calculate the flux by describing the diffusion of the solute throughout the solvent as a random walk with a fixed characteristic step length, λ, or mean free path, using the Einstein-Smoluchowsi equation [16]. However, this is unlikely to work well in liquids, since there is no characteristic diffusional step length [17] and a classical mechanical description is more appropriate for large solute molecules (M > 300 g / mol) diffusing through a small solvent like water or ethanol [17].

Within the classical mechanical limit, the flux of a solute to the surface of a solid from liquid solution is just $C_s \bullet v_T$, where C_s is the concentration of the solute in that liquid nearest to the surface [18] and v_T is the "mean thermal velocity of the molecule relative to the solid surface," as has been shown by Schurr [18]. This thermal velocity for a spherical molecule of mass m relative to an infinitely heavy spherical solid of infinite radius can be shown following Schurr [18] to be:

$$v_T = \sqrt{\frac{k_B T}{2\pi m}}, \tag{8}$$

where k_B is Boltzmann's constant. This gives the flux onto a surface, or the collision frequency, as:

$$J_s = C_s \sqrt{\frac{k_B T}{2\pi m}}. \tag{9}$$

Interestingly, this is the same as Eq. 2 for the collision frequency in gas phase adsorption. Since diffusion is fast for gases, one can usually use the "bulk" concentration for gases. The flux in liquids must be calculated by accounting for the fact that the concentration of that solution in nearest contact to the surface, C_s, can be depleted due to adsorption. The equation for S in liquids then becomes

$$S = \frac{R_{ads}}{J_s} = \frac{R_{ads}}{C_s} \sqrt{\frac{2\pi m}{k_B T}}, \tag{10}$$

where R_{ads} is experimentally measured (with SPR here) and v_T is calculated for the molecules at the desired temperature, leaving only C_s to be determined. However, if the diffusion constant for the solute, D, is known, one can determine C_s since there is only one set of concentrations versus distance and time from the surface which are consistent with: (1) the initial conditions, (2) the observed adsorption rate versus time, and (3) Fick's law.

This chapter will describe a method for estimating C_s and, from it, the intrinsic rate constant (k) and sticking probability as a function of time and coverage from measurements of adsorbate coverage versus time. A finite difference method is utilized, where Fick's law is used to calculate the adsorbate concentrations at incremental distances from the surface with incrementally increasing time, under the boundary conditions that the rate of change in concentration in the liquid volume element nearest to

the solid surface must equal the diffusive flux coming from the next volume element further from the surface minus the measured adsorption rate onto the surface. The diffusion constant of the adsorbate through the liquid, D, must be known to apply this method. The method is similar to methods previously reported [15,19-23] whereby the adsorption transient is fit to simulations of the adsorption kinetic equation taking into account diffusion and concentration gradients, via finite difference methods. However, our current treatment differs in that the intrinsic adsorption rate constant versus coverage is directly extracted from the adsorption transient data, rather than assuming some form of the intrinsic rate law and seeing if it fits the data. Furthermore, this intrinsic rate constant is used to determine a sticking probability. Examples of the application of these principles to quantitative measurements of adsorption kinetics from liquid solutions by surface plasmon resonance (SPR) spectroscopy will be discussed. Primarily, this method will be demonstrated with the determination of S and the intrinsic rates for the adsorption of alkylthiols onto gold. To our knowledge, this is the first time accurate sticking probabilities have been determined for adsorption from liquid solutions.

The adsorption of alkylthiols on gold has been studied actively, as it provides a well-ordered monolayer at the liquid / solid interface which forms the basis for many recipes for the organofunctionalization of gold [24-43]. Several studies have addressed the kinetics of their adsorption from liquids [1,5-8,14,44-52].

METHODS

The methods and materials have been described in the Chapter 2. The measured SPR shifts were converted into coverages with the formalism presented in Chapter 3.

The adsorption transient is initiated by injecting 0.5 ml of the adsorbate-containing solution through the SPR sample cell at a flow rate of 10 ml / min, which lasts for a few seconds during the start of adsorption, but then it stops. This flow introduces some minor complications into the calculations to determine C_s at short times, as will be discussed in more detail below. Control experiments switching in this way between solutions of different refractive indices were monitored with SPR (where the solutions were chose to avoid adsorption onto the SPR sensor's surface), and the response was reasonably fitted by an equation of the form $1 - \exp(-t / \tau_{sys})$, where τ_{sys} , or the system time constant, was found to be ~0.8 s [55].

CALCULATION METHODS:

The finite-difference numerical method used to calculate the flux is similar to those previously described [19-21]. The solution is divided up into differential layers of equal height, Δh, parallel to the surface. In this paper, the flow cell was divided into 2000 layers, each 2500 Å thick. Since the concentration gradients here are not too steep, this step size allows accurate calculations to estimate the concentration nearest the surface. Each layer is assigned an integer index, i, where the layer in contact with the surface is 0 and each successive layer is numbered consecutively, as shown in Fig. 4.2. Each layer has its own concentration with corresponds to the concentration in its center. At time 0 (i.e., when the adsorption transient is starting), each layer is assigned the known initial bulk concentration (molecules / cm^3), which is expected for such a fast-response stop-flow system as used here. As the adsorbate then sticks to the surface, solution layer 0 will be depleted, resulting in a concentration gradient between it and the next layer. A finite difference representation of Fick's law is used to calculate the resulting change in concentration of these two layers and each successive layer, due to diffusion, with each incremental time step, dt. We impose in each time step the boundary condition that the irreversible flux out of the zeroth layer (to the surface) is just the

measured rate of adsorption at that time. (This software is presented in Appendix B.) These rates are determined by differentiation of the measured coverage versus time. To eliminate noise that is amplified by this numerical differentiation, the adsorption curves were first fit with a triple exponential sum of the form:

$$\theta(t) = \theta_{max}\left(1 - \sum_i \left[X_i \exp\left(-\frac{t}{\tau_i}\right)\right]\right) \qquad (11)$$

where x_i and τ_i are fitting parameters, and i ran from 1 to 3. This effectively smooths the curve and fits the data well within its scatter before differentiation.

The diffusion constant, D, for the thiols in ethanol were estimated to vary between 9.5×10^{-6} to 5.7×10^{-6} cm^2 / s from C_2 to C_{18}, respectively, based on the value of D for a similar molecule, allyl alcohol, in ethanol [56] and then adjusting for the thiols volume by assuming they are spherical molecules with their bulk density and using the Stokes-Einstein equation relating D to the effective hydrodynamic radius. The values for $C_{15}COOH$ and $C_{11}PEO$ were estimated to be 5.8×10^{-6} cm^2 / s and 5.7×10^{-6} cm^2 / s, respectively in this same way.

The injection of sample solution does not give a perfect step function in concentration of thiol versus time, but rather this flow system has a instrument response of ~0.8 s [55]. In addition, the injections are not perfect stop-flow experiments, since the flow was continued for a few seconds after the adsorbate was introduced into the flow cell, to ensure complete exchange of solutions. During these few seconds of flow there is a stagnant region which depletes close to the surface, despite the flow. This stagnant region is known as the diffusion layer and its depth, δ, can be estimated from [57]

$$\delta = D^{\frac{1}{3}} \nu^{\frac{1}{6}} \sqrt{\frac{x}{u_0}} \qquad (12)$$

where u_0 is the mean stream velocity of 10 ml / min, v is the kinematic viscosity which is estimated to the about same as for water, 0.009 cm^2 / s [16], and x is the distance from the entrance of flow cell to sensing region, ~0.9 cm. This results in a δ of ~0.004 cm for alkylthiols. Small corrections in the boundary conditions of the solution to Fick's law were used to simulate this effect, but it was minor.

RESULTS

ALKYLTHIOL ADSORPTION

Fig. 4.3 shows SPR measurements of the surface coverage versus time for the adsorption of a series of alkylthiols onto the gold surface at room temperature from ethanol solutions. (Only the most important 40 s of adsorption is shown here, although the adsorption was followed to long times (~400 - 800 s) until saturation is slowly reached.). The thiol concentration was 0.5 mM in all cases except for the longest chains: C_{16} (0.25 mM), and C_{18} (0.2 mM). The thiol solutions were injected into the flow cell (initially filled with pure ethanol) at time 0, initiating adsorption and assembly. The SPR results for $C_{15}COOH$ (0.12 mM) and $C_{11}PEO$ (0.1 mM) in ethanol are shown in Fig. 4.4a, along with the same C_{16} alkylthiol adsorption curve for comparison. The inset in Fig. 4.4a. shows the initial adsorption times for these curves. In Fig. 4.4b, the SPR shifts of Fig. 4.4a have been converted to thiol coverage (molecules / cm^2) following the method described above.

Saturation is reached at about the same coverage, $4.8\pm0.2 \times 10^{14}$ molecules / cm^2, for all the alkylthiols shown in Fig. 4.3, consistent with prior results [52]. The $C_{11}PEO$ saturation coverage of $\sim 3.78 \times 10^{14}$ molecules / cm^2 is about 80% of this value, which agrees with other studies with similar thiols [58]. The resulting 4.35×10^{14}

molecules / cm^2 saturation coverage for $C_{15}COOH$ is nearly as close packed as the alkylthiols, agreeing well with previous XPS results [59].

Fig. 4.5 shows the results of Fig. 4.3 replotted as the rate of adsorption versus coverage. The rates in each case were determined from the local derivative of the coverage versus time curve from Fig. 4.3, smoothed as described above. As can be seen in Fig. 4.5, the rates generally increase with increasing number of carbon atoms in the alkylthiol. While the rate for C_{16} and C_{18} look about the same as for C_{10}, it should be remembered that these curves were measured at 2-2.5-fold lower concentration than the other curves, which slows their rates almost proportionately. Below we compare rate constants which take this (and diffusion-induced concentration depletion) into account.

As also obvious in Fig.3.5, the rates decay nearly linearly with coverage, extrapolating to zero at a coverage of ~4.0±0.4 $x10^{14}$ molecules / cm^2. This is very close to the expected packing densities, which are controlled by the lateral attractions between alkyl chains [60]. Note, however, that another ~20% adsorbs, beyond the zero-point extrapolation, albeit at a much, much slower rate. This suggests that another process controls the adsorption kinetics in this last 20% to saturation, perhaps due to the population of another site or adsorption geometry. The linear decrease seen in the first ~83% of saturation is expected for first-order Langmuir adsorption [16]. However, a more appropriate test of this is to examine the rates after correction for diffusional-induced concentration depletion near the surface, as we do next.

Fig.3.6 shows the data of Figs. 3.3 and 3.5 replotted as sticking probability (S) versus coverage, using Eq. 10 to calculate the sticking probability. The concentration in the liquid nearest the surface versus time, which was needed for this calculation, was determined by a finite-differences solution to Fick's law as described above, using the observed rate as a boundary condition. As can be seen, the sticking probability for each species decreases nearly linearly with coverage as expected for a first-order Langmuir adsorption mechanism [16]:

$$S = S_0(1 - \theta / \theta_{max}), \quad (13)$$

where S_0 is the (extrapolated) initial sticking probability and θ / θ_{max} is the coverage relative to saturation (of this linear behavior). For the two species with the fastest rates (C_{16} and C_{18}), the lowest coverage points fall slightly below the linear trend extrapolated from higher coverages. This is probably due to the fact that these points are at the shortest times (< 2 s) and correspond to the fastest rates of signal change, so that the observed rates are slightly depressed by the instrument response time (~0.8 s).

In converting data from Fig. 4.5 into Fig. 4.6, the corrections due to diffusion limitations and concentration gradients were found to be relatively minor for the short-chain alkyl thiols, but increased as the chain-length grew, since longer thiols have faster adsorption rates and slower diffusion rates. The corrections are quite substantial for the longest chains, which were also measured at lower bulk concentrations, increasing the extent of depletion. The way in which this depletion evolves with time and distance from the surface, according to our solution to Fick's law, is shown in Fig. 4.7 for the C_{18} thiol. For C_{18}, the concentration nearest the surface was depleted to ~40% of the bulk value over most of the adsorption coverage range.

Note that each curve in Fig. 4.6 is scaled by the initial sticking probability, S_0. It was found that So increased by a factor of ~65 with chain length. This can be seen more clearly in Fig. 4.8, where the initial sticking probability (S_0, determined by fitting data from Fig. 4.6 to Eq. 13) is plotted in log format versus chain length. The solid line shows the linear best fit to these data, which corresponds to the equation:

$$S_0 = P_0 \bullet \exp\left(b \bullet N_{CH_2}\right), \quad (14)$$

where N_{CH2} is the number of methylene groups in the alkyl chain (not counting the terminal CH_3), and P_0 and b are constants. Their best-fit values are $P_0 = 1.15 \times 10^{-8}$ and b

= 0.27 per methylene group. The value P_0 presumably corresponds to the (extrapolated) S_0 for no CH_2 groups (i.e., for methanethiol).

The chain-length dependence of the sticking probability in Fig. 4.8 is remarkable in that it follows an exponential behavior, Eq. 14. This is easily understood on the basis of energetic effects of added methylene units on the stability (i.e., Gibb's free energy) of the transition state for adsorption (relative to the solution phase species). Within this picture, the slope b = 0.27 corresponds to the transition-state stabilization energy per methylene group, in units of RT at the experimental temperature (T = 298K). This corresponds to a decrease in the activation energy for adsorption by 0.67 kJ / mol per methylene group. One can thus express S_0 as:

$$S_0 = \upsilon \exp\left(-\frac{E_0 - \left(0.67\frac{kJ}{mol}\right)N_{CH_2}}{RT}\right), \qquad (15)$$

where E_0 is the activation energy for an alkylthiol with no methylene groups in the alkyl chain (i.e., $HSCH_3$) and ν is the prefactor. Note that $P_0 = \nu \exp[-(E_0)] = 1.1\text{x}10^{-8}$.

Just the opposite trend was seen in the desorption rates of these same thiolate species by Bain et al. [46], who concluded that their activation energy for desorption into hexadecane solvent increases by ~0.8 kJ / mol per methylene group. It was assumed that this arises from stabilization in the adsorbed state due to additional attractive interactions to the surface with each added methylene group, so we assume here that this same value applies in ethanol (our solvent). Together with our current data, a fairly complete picture of the free-energy surface for the adsorption / desorption process thus evolves, as shown in Fig. 4.9. This figure takes as the reference state the species in liquid phase solution. The absolute energy differences between the states are not exactly known, but the relative energy between the different chain lengths shown there are now clear. The absolute free energy of the transition state has been estimated by assuming

that the prefactor for adsorption, ν in Eq. 15, is unity. This gives, for example, and activation free energy for adsorption of the C_{18} thiol of 33 kJ / mol. The absolute energy of the adsorbed state was estimated from the desorption activation energy of adsorbed docosanethiolate in hexadecane, which was reported to be 117 kJ / mol [52]. (The product there might have been the dissolved disulfide rather than the thiol.) By summing the effects of chain length on the desorption and adsorption activation energies (0.8 and 0.67 kJ / mol per methylene, respectively), the stabilization of the adsorbed state results: 1.5 kJ / mol per methylene group.

Fig. 4.9 displays the classic behavior of linear-free-energy relationships [61]. The transition state for adsorption is stabilized by each additional methylene unit by 0.67 kJ / mol, or 43% of the stabilization of the final adsorbed product. As a consequence, the activation energy state for desorption is higher by 57% of this value (per CH_2). This means that the transition state resembles the adsorbed state to some extent (43%) but more closely resembles the solution phase species (57%). This suggests a picture for the transition state whereby the molecule is partially bonding to the Au surface, but wherein it still has not lost all its surrounding solvent, nor has the gold completely lost the adsorbed solvent molecule in that area. The activation barrier is thus associated to some extent with the displacement of solvent molecule(s) from the gold. The involvement of steric repulsions with solvent in creating the adsorption activation barrier is supported by the observation that these same thiols have an initial sticking probability that is many orders of magnitude larger when adsorbing from the gas phase onto clean gold in ultrahigh vacuum [62]. Fig. 4.9 helps explain the difficulty Bain et al. [44] had in understanding the fact that adsorption is favored for those alkylthiols that have the strongest adsorption energies. This is not, however, due to the fact that "thermodynamics control the adsorption" as they proposed. Equilibrium is far too slow at room temperature. It simply looks this way, since about half the product stabilization is already present in the transition state for adsorption.

The stabilization of the transition state and adsorbed state with chain length are similar in magnitude to the stabilization energy of the pure, bulk solids relative to their liquid melts, which is ~2.5 kJ / mol per methylene group in long-chain paraffins [46]. This suggests some cooperativity in the adsorption transition state (stabilization due to the proximity of a nearby adsorbate). Alternatively, the incremental stabilization per methylene group could arise due to induced dipole interactions with the polarizable conduction electrons of the gold. The first-order Langmuir adsorption kinetics observed here, and the first-order desorption kinetics reported by Bain et al. [46] are both usually interpreted to indicate lack of cooperativity. First-order Langmuir adsorption kinetics are also consistent with adsorption at all coverages taking place in the bare areas between large high density adsorbate islands, where attractive lateral interactions are maximized. Poirier [63] reported STM images that prove that large islands are indeed formed in vacuum surrounded by less dense areas. However, it is not obvious how first-order Langmuir desorption kinetics would results from such islanding.

Intrinsic rate constants for adsorption ($k_{intrinsic}$) are more commonly reported than sticking probabilities for rate measurements with liquid phase reactants. These are usually defined as the initial (i.e., extrapolated to zero-coverage) rate per unit area per unit concentration (in the liquid nearest the surface), or, for first-order Langmuir adsorption like we see here, as the rate per unit of concentration (in the liquid nearest the surface) per unit of unoccupied surface area [22,23]. These have been accurately determined in some cases where the concentrations were high enough and the rate slow enough to avoid depletion near the surface, or by numerical simulations of rate equations that also consider Fick's law [15,22,23]. The intrinsic rate constant can easily be calculated from the initial sticking probabilities in Fig. 4.8 by simply dividing by the thermal velocity (see Eq. 8):

$$k_{intrinsic} = \frac{R_{ads,0}}{C_s} = \frac{S_0}{\nu_T} = \frac{S_0}{\sqrt{\frac{k_B T}{2\pi m}}}. \qquad (15)$$

These rate constants usually have units of molecules cm^{-2} s^{-1} per (mole per liter), or monolayers s^{-1} per (mole per liter) [61]. They are more easily manipulated in units of molecules cm^{-2} s^{-1} per (molecule per cm^3). Their values are presented in Fig. 4.10. They do not vary quite as much with chain length as do the values of S_0 in Fig. 4.8, due to the effect of mass on flux at a given concentration. (The mass here reduces the difference between C_8 and C_{18} from a 67-fold to a 31-fold increase.)

The sticking probabilities versus coverage are presented for C_{11}PEO and C_{15}COOH in Fig. 4.11. For comparison, the values for the C_6 and C_{16} alkylthiols are also shown there. The slight curvature observed at the lowest coverages for these molecules are most likely due to non-ideality of the injection and time response at very short times, as described above. A small portion of the data points (at the high-coverage end) also deviate from this linear model, as it did with the alkylthiols. The initial sticking probabilities of the C_{11}PEO and C_{15}COOH terminated alkylthiols are also reported in Fig. 4.8, where only their CH_2 groups were used to locate them on the x-axis. As can be seen, the PEO and COOH groups also strongly affect their adsorption rates. The eight extra CH_2 units in the PEO group increased the S_0. The COOH group decrease S_0.

As can be seen in Fig.3.11, the value of S for C_{11}PEO falls to < 5% of its initial value at about half the coverage required for all the other thiols. However, this fast initial adsorption stage (in which Langmuir kinetics are only very approximately behaved) is followed by a much longer extended adsorption stage than the other thiols, proceeding at a greatly reduced sticking probability. In the end, a relatively high saturation coverage is reached (~3.78×10^{14} molecules / cm^2) for C_{11}PEO, which is only ~79% below the value seen for the alkylthiols (~$4.8\pm0.2\times10^{14}$ molecules / cm^2). This implies that a slow process is involved in the self-organization of the C_{11}PEO monolayer which allows its dense packing. Without this self-organization, the area blocked by a C_{11}PEO thiol is about twice that of a simple alkylthiol.

DISCUSSION

As mentioned above, the purely diffusion - controlled adsorption model (Eq. 6, which assumes S = 1.0) has been modified by some authors [3,13] into Eq. 7, which they assumed was approximately correct (but could not prove). When our adsorption data were fit to Eq. 7, S_0 values were obtained that were ~10^6 - fold greater than the values reported above. For example, S_0 values of 0.053 and 0.45 were estimated for C_2 and C_{16}, respectively, as compared to the values obtained above of $2x10^{-8}$ and $6x10^{-7}$. This proves that Eq. 7 is grossly inaccurate, as are all sticking probabilities reported previously using it [2,13]. This inaccuracy arises because Eq. 6 results from a steady-state approximation and the assumption that S is unity at all times, which causes the concentration nearest the surface to drop many orders of magnitude below the bulk value. A much, much lower sticking probability is needed to cause that concentration to increase to a value that is, for example, only a factor of two below the bulk's. Since the (nearly linear) concentration gradient extends across roughly the same distance (i.e., the diffusion length defined above), then the diffusive flux (and therefore net rate) will only be reduced by a factor of about two. In other words, the net adsorption rate is fairly insensitive to decreases in the sticking probability for large sticking probabilities, since the concentration nearest the surface responds nearly proportionately in the opposite direction, which can be ascertained from the simulations by Corsel et. al [22]. As another consequence of this, the new method presented here for determining sticking probabilities will probably only be able to set a lower limit for molecules with large sticking probabilities (> ~10^{-4} for concentrations in the range used here).

The increase in the S_0 with chain length (Fig. 4.8) agrees with some previous results which found that longer chains assembled faster [46,51] and adsorbed preferentially [45,46,64] over shorter chains. Previous studies have also shown pronounced increases in adsorption rate between the short (n < 9) and long (n > 9) chains, which they qualitatively attributed to the interchain van der Waals attraction energy [65]. It is also interesting that the ratios of thiolates obtained by adsorption of asymetric

diaklylthiols are similar to the ratios of the sticking probabilities obtained here for the corresponding thiols. For example, a C_8 - C_{16} dialkylthiol gave a mole fraction of C_8 thiolate on the surface of only 0.12-0.25 by XPS, and a C_{12} - C_{16} dialkylthiol gave a mole fraction of C_{12} thiolate on the surface of only 0.5-0.32 [66]. This is comparable to the ratios of initial sticking probabilities of 0.14 and 0.20 for the C_8:C_{16} and C_{12}:C_{16} alkylthiols respectively.

At least two studies appear to contradict the trend observed here of increasing adsorption rate with alkyl thiol chain length [1,8]. In these cases, it is very likely that the experiments were performed in such a way that concentration depletion was a major problem, and thus the adsorption was strongly dependent on diffusion rates. For example, increasing chain length decreased the adsorption rate constant according to Dannenberger et. al [1]. For their experiment, however, the half-life for adsorption of dodecanthiol at 0.2 μM concentration was found to be ~150 s, which is much less than the diffusion time, τ_{diff} (calculated from Eq. 4), of 5150 s. This proves that a very strong diffusion depletion occurred, which would be worst for the larger molecules (since they have lower D values). A totally diffusion-limited rate (Eq. 6) would show an decrease with rate with chain length. Depletion was not a problem for the data of Peterlinz et al [8] who used concentrations as high as 1 mM. However, their half-lives of ~50 s (taken from their published adsorption curves) at 1 mM in ethanol is significantly slower than in our present results and reports in [6,45,65]. We are not sure what caused this difference for their data.

Lavrich et al. [67] studied the kinetics of the conversion of physisorbed alkylthiols to chemisorbed alkylthiolates on Au(111) under ultrahigh vacuum and found a barrier of ~29 kJ / mol, independent of chain length. The physisorbed states was stabilized by each additional CH_2 group by ~6 kJ / mol (relative to the gas phase species), which implies that the energy of the transition state to chemisorption is also stabilized by ~6 kJ / mol per CH_2 relative to the gas phase. Since the species is stabilized by each CH_2

group also when dissolved in ethanol, it is clear that this transition state will be stabilized less than 6 kJ / mol per CH_2 group relative to the species in ethanol liquid. This is consistent with the stabilization of 0.67 kJ / mol per CH_2 seen here, although the nature of the transition state may be quite different for the process here. Interestingly, however, the value we estimate for the energy of the transition state relative to the dissolved species (for zero methylene groups) of 33 kJ/mol is close to the 29 kJ/mol they report for the energy of the transition state for chemisorption above that of the physisorbed thiol [67].

In general, the thiols with bulkier functional groups have been previously shown to adsorb more slowly than their corresponding alkylthiol [45]. For example, in an equimolar mixture of $HS(CH_2)_{10}COOH$ and $HS(CH_2)_{10}CH_3$ alkylthiol solution, the $C_{15}COOH$-terminated thiol composed only ~30% of the surface thiolates [45]. Similarly, the S_0 presented here for the $C_{15}COOH$ is only about 40% of that for the C_{16} thiol. There are no similar previous $C_{11}PEO$ thiol kinetic studies to compare the results for the $C_{11}PEO$ here. It adsorbs faster than the simple C_{12} alkylthiol, which suggest that the PEO groups (OCH_2CH_2) help stabilize the transition state, as with CH_2 groups. This effect is not as strong as with pure CH_2 groups, however, since S_0 for $HS(CH_2)_{11}(OCH_2CH_2)_4OH$ is lower than that predicted for a C_{19} alkylthiol (by extrapolation of the line in Fig. 4.8).

A number of things might improve the values of S_0 or $k_{intrinsic}$. Particularly noticeable in this paper would be to reduce the non-ideality of the injection. Since the depletion of the solution can be estimated with time, the simplest way to eliminate this uncertainty would be performing these adsorption experiment at a slightly lower concentration so that the time constant of the flow system is not significant when compared to the time constant of the adsorption. Direct measurements of the diffusion constants (rather than estimating them based on reported values for related molecules) would improve the results as well. For example, dynamic light scattering can be used to measure the diffusion constants for many large biological molecules within a few percent error. Other factors, including the quality and cleanliness of the substrates, can also be

significant. For example, this is important when adsorbing alkylthiols, where an induction time was observed by others onto an unclean surface. This was believed to be caused by the thiols having to displace contaminates in order to adsorb to the Au [8].

CONCLUSIONS

A method is presented for determining sticking probabilities of molecules adsorbing from liquid solution by measurements of coverage versus time. Concentration depletion due to finite diffusion rate is taken into consideration by a numerical solution to Fick's law, with the measured rate as a boundary condition. Sticking probabilities, as well as intrinsic rate constants, have been determined from measured adsorption rates of alkylthiols on gold from ethanol solution. The adsorbate flux to the surface during adsorption was determined from application of Fick's Law. The initial sticking probabilities (S_0) of alkylthiols increase with chain length from 10^{-8} to 10^{-6} for the alkylthiols studied here. This implies that the transition state for adsorption is stabilized by 0.67 kJ / mol per CH_2 group, or about half the stabilization in the adsorbed state. First-order Langmuir adsorption kinetics are followed up to a coverage of ~4.0×10^{14} molecules / cm^2, after which another ~20% adsorbs at a much slower rate. Bulky functional groups lower the S_0 when compared to alkylthiols of equivalent length.

NOTES TO CHAPTER 4

[1] Dannenberger, O.; Buck, M.; Grunze, M. *J. Phys. Chem. B* **1999**, *103*, 2202-2213.

[2] Grabar, K. C.; Smith, P. C.; Musick, M. D.; Davis, J. A.; Walter, D. G.; Jackson, M. A.; Guthrie, A. P.; Natan, M. J. *J. Am. Chem. Soc.* **1996**, *118*, 1148-1153.

[3] Park, K.; Park, H.; Albrecht, R. M. *Quantitative Characterization of Colloidal Gold Statining*; Hyatt, M. A., Ed.; Academic Press: San Diego, 1989; Vol. Chapter 28.

[4] Bond, W. N.; Puls, H. O. *Phil. Mag.* **1937**, *7*, 864-888.

[5] DeBono, R. F.; Loucks, G. D.; Mann, D. D.; Krull, U. J. *Can. J. Chem.* **1996**, *74*, 677-688.

[6] Karpovich, D. S.; Blanchard, G. J. *Langmuir* **1994**, *20*, 3315-3322.

[7] Pan, W.; Durning, C. J.; Turro, N. J. *Langmuir* **1996**, *12*, 4469-4473.

[8] Peterlinz, K. A.; Georgiadis, R. *Langmuir* **1996**, *12*, 4731-4740.

[9] Rahn, J. R.; Hallock, R. B. *Langmuir* **1995**, *11*, 650-654.

[10] Golander, C.-G.; Kiss, E. *J. of Coll. Int. Sci.* **1988**, *121*, 240-253.

[11] Andrade, J. D. *Protein Adsorption*; Plenum Press: New York, 1985; Vol. 2.

[12] Motschmann, H.; Stamm, M. *Macromolecules* **1991**, *24*, 3681-3688.

[13] Park, K.; Simmons, S. R.; Albrecht, R. M. *Scanning Microscopy* **1987**, *1*, 339-350.

[14] Schlenoff, J. B.; Li, M.; Ly, H. *J. Am Chem. Soc.* **1995**, *117*, 12528-12536.

[15] Ward, A. F. H.; Tordai, L. *J. Chem. Phys.* **1946**, *14*, 453-461.

[16] Atkins, P. W. *Physical Chemistry*; 4th ed.; W. H. Freeman and Company: New York, 1990.

[17] Nir, S.; Stein, W. D. *J. Chem. Phys.* **1971**, *55*, 1598-1603.

[18] Schurr, J. M. *Biophysical Journal* **1970**, *10*, 700-716.

[19] Prater, K. B.; Bard, A. J. *J. Electrochem. Soc.* **1970**, *117*, 209-213.

[20] Ruzic, I.; Feldberg, S. *Electroanalytical Chemistry and Interfacial Electrochemistry* **1974**, *50*.

[21] Feldberg, S. W. *Digital Simulation: A General Method for Solving Electrochemical Diffusion-Kinetic Problems*; Bard, A. J., Ed.; Marcel Dekker Inc.: New York, 1969; Vol. 3, pp 199-296.

[22] Corsel, J. W.; Willems, G. M.; Kop, J. M. M.; Cuypers, P. A.; Hermens, W. T. *Journal of Colloid and Interface Science* **1986**, *111*, 544-554.

[23] Zhdanov, V. P.; Kasemo, B. *Proteins: Structure, Function, and Genetics* **1998**, *30*, 177-182.

[24] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys.* **1993**, *9*, 7012-7019.

[25] Tengvall, P.; Lestelius, M.; Leidberg, B.; Lundstrom, I. *Langmuir* **1992**, *8*, 1236-1238.

[26] Lang, H.; Duschl, C.; Vogel, H. *Langmuir* **1994**, *10*, 197-210.

[27] Mrksich, M.; Sigal, G. B.; Whitesides, G. M. *Langmuir* **1995**, *11*, 4383-4385.

[28] Mrksich, M.; Grunwell, J. R.; Whitesides, G. M. *J. Am. Chem. Soc* **1995**, *117*, 12009-12010.

[29] Mrksich, M.; Whitesides, G. M. *Annu. Rev. Biophys. Biomol. Struct* **1996**, *25*, 55-78.

[30] Frey, B. L.; Claire, E. J.; Kornguth, S.; Corn, R. M. *Anal. Chem* **1995**, *67*, 4452-4457.

[31] Sigal, G. B.; Barndad, C.; Barberis, A.; Strominger, J.; Whitesides, G. M. *Anal. Chem.* **1996**, *68*, 490-497.

[32] Delamarche, E.; Sundarababu, G.; Biebuyck, H.; Michel, B.; Gerber, C.; Sigrist, H.; Wolf, H.; Ringsdorf, H.; Xanthopoulos, N.; Mathieu, M. J. *Langmuir* **1996**, *12*, 1997-2006.

[33] Hegner, M.; Dreier, M.; Wagner, P.; Semenza, G.; Güntherodt, H. J. *J. Vac. Sci. Technol. B* **1996**, *14*, 1418-1421.

[34] Rickert, J.; Gopel, W.; Beck, W.; Jung, G.; Heiduschka, P. *Biosensors & Bioelectronics* **1996**, *11*, 757-768.

[35] Silin, V.; Weetall, H.; Vanderah, D. J. *Journal of Colloid and Interface Science* **1997**, *185*, 94-103.

[36] Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

[37] Jordan, C. E.; Corn, R. M. *Anal. Chem* **1997**, *69*, 1449-1456.

[38] Plant, A. L. *Langmuir* **1993**, *9*, 2764-2767.

[39] Prime, K. L.; Whitesides, G. M. *J. Am. Chem. Soc.* **1993**, *115*, 10714-10721.

[40] Patel, N.; Davies, M. C.; Hartshorne, M.; Heaton, R. J.; Roberts, C. J.; Tendler, S. J. B.; Williams, P. M. *Langmuir* **1997**, *13*, 6485-6490.

[41] Terrettaz, S.; Stora, T.; Duschl, C.; Vogel, H. *Langmuir* **1993**, *9*, 1361-1369.

[42] Pertsin, A. J.; Grunze, M.; Garbuzova, I. A. *J. Phys. Chem. B.* **1998**, *102*, 4918-4926.

[43] Bunjes, N.; Schmidt, E. K.; Jonczyk, A.; Rippmann, F.; Beyer, D.; Ringsdorf, H.; Graber, P.; Knoll, W.; Naumann, R. *Langmuir* **1997**, *13*, 6188-6194.

[44] Bain, C. D.; Whitesides, G. M. *J. Am. Chem. Soc.* **1989**, *111*, 7164-7175.

[45] Bain, C. D.; Evall, J.; Whitesides, G. M. *J. Am. Chem. Soc.* **1989**, *111*, 7155-7164.

[46] Bain, C. D.; Troughton, E. B.; Tao, Y.-T.; Evall, J.; Whitesides, G. M.; Nuzzo, R. G. *J. Am. Chem. Soc.* **1989**, *111*, 321-335.

[47] Biebuyck, H. A.; Bain, C. D.; Whitsides, G. M. *Langmuir* **1994**, *10*, 1825-1831.

[48] Buck, M.; Eisert, F.; Fischer, J.; Grunze, M.; Trager, F. *Appl. Phys. A* **1991**, *552*, 556.

[49] Kim, H. J.; Kwak, S.; Kim, Y. S.; Seo, B. I.; Kim, E. R.; Lee, H. *Thin Solid Films* **1998**, *327-329*, 191-194.

[50] Schessler, H. M.; Karpovich, D. S.; Blanchard, G. J. *J. Am Chem. Soc.* **1996**, *118*, 9645-9651.

[51] Xu, S.; Cruchon-Dupeyrat, S.; Garno, J. C.; Liu, G.-Y.; Jennings, G. K.; Yong, T.-H.; Laibinis, P. E. *J. Chem. Phys.* **1998**, *108*, 1-11.

[52] Nuzzo, R. G.; Zegarski, B. R.; Dubois, L. H. *J. Am. Chem. Soc.* **1987**, *109*, 733-740.

[53] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[54] Uvdal, K.; Bodo, P.; Liedber, B. *Journal of Colloid and Interface Science* **1991**, *149*, 162-173.

[55] Jung, L. S.; Nelson, K. E.; Campbell, C. T.; Stayton, P. S.; Yee, S. S.; Perez-Luna, V.; Lopez, G. P. *Sensors and Actuators B* **1999**, *54*, 137-144.

[56] Lide, D. R. *Handbook of Chemistry and Physics*; 71st ed.; CRC Press: Boston, 1990.

[57] Levich, V. G. *Physicochemical Hydrodynamics*; Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1962.

[58] Harder, P.; Grunze, M.; Dahint, R.; Whitesides, G. M.; Laibinis, P. E. *J. Phys. Chem. B* **1998**, *102*, 426-436.

[59] Yan, L.; Marzolin, C.; Terfort, A.; Whitesides, G. M. *Langmuir* **1997**, *13*, 6704-6712.

[60] Dubois, L. H.; Nuzzo, R. G. *43* **1992**, 437-64.

[61] Gardner, W. C. *Rate and Mechanisms of Chemical Reactions*; W. A. Benjamin Inc.: Menlo Park, CA, 1972.

[62] Dixon-Warren, S. J.; Bondzie, V.; Burson, N.; Lucchesi, L.; Yu, Y.; Zhang, L. *J. Vac. Sci. Technol. A* **1999**, *17*, 2982-2986.

[63] Poirier, G. E. *Langmuir* **1999**, *15*, 1167-1175.

[64] Bain, C. D.; Whitesides, G. M. *Science* **1988**, *240*, 62-63.

[65] Ulman, A. *Chem. Rev.* **1996**, *96*, 1533-1554.

[66] Heister, K.; Allara, D. L.; Bahnck, K.; Frey, S.; Zharnikov, M.; Grunze, M. *Langmuir* **1999**, *15*, 5440-5445.

[67] Lavrich, D. J.; Wetterer, S. M.; Bernasek, S. L.; Scoles, G. *J. Phys. Chem. B* **1998**, *102*, 3456-3465.

FIGURES FOR CHAPTER 4

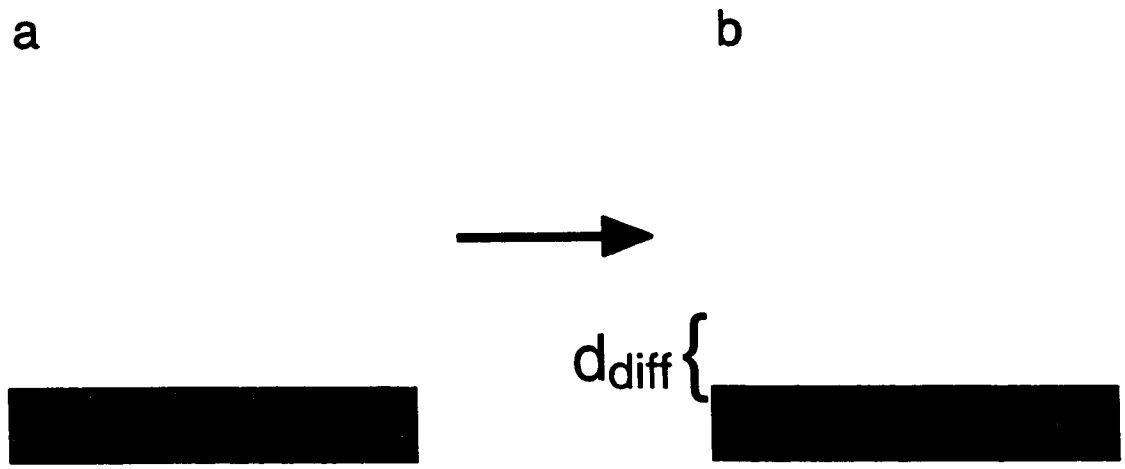


Figure 4.1 Illustration of characteristic diffusion distance, d_{diff}. (a) represents a solution of some concentration in contact with a solid. (b) represents that same solution if half a the saturation coverage were suddenly adsorbed to the surface to give an (unrealistic) step-like concentration profile with 100% depletion into the solution to some distance, d_{diff}.

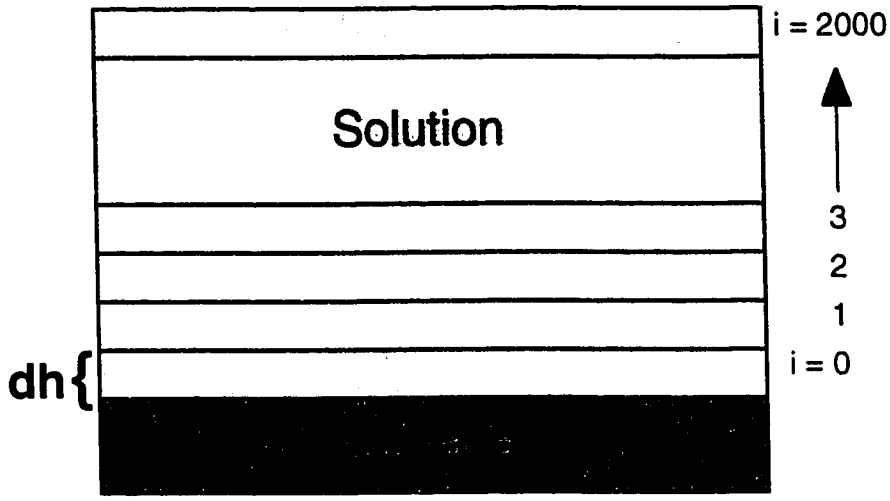


Figure 4.2 Illustration of how the solution is divided up in to differential layers above the surface. Each layer is given a unique number, i and has equal height, Δh. The concentration at the surface, i=0, is C_s.

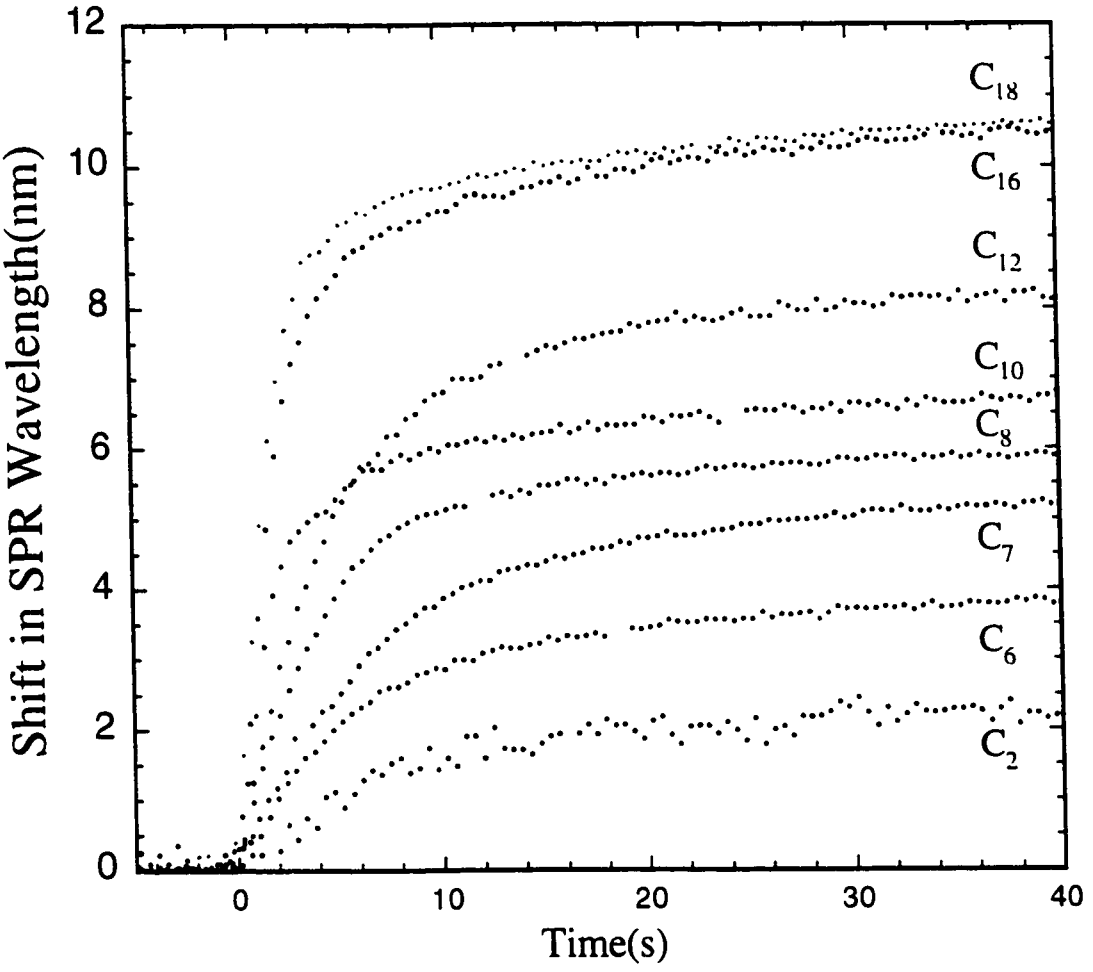


Figure 4.3 The SPR response time versus time upon adsorption of C_2 (0.5 mM), C_6 (0.5 mM), C_7 (0.5 mM), C_8 (0.5 mM), C_{10} (0.5 mM), C_{12} (0.5 mM), C_{16} (0.25 mM), and C_{18} (0.2 mM) alkylthiols from ethanol solution onto Au, initiated by the injection of the thiol solution into the SPR cell at time 0. The time range shown is much less than that followed and that needed to reach true saturation.

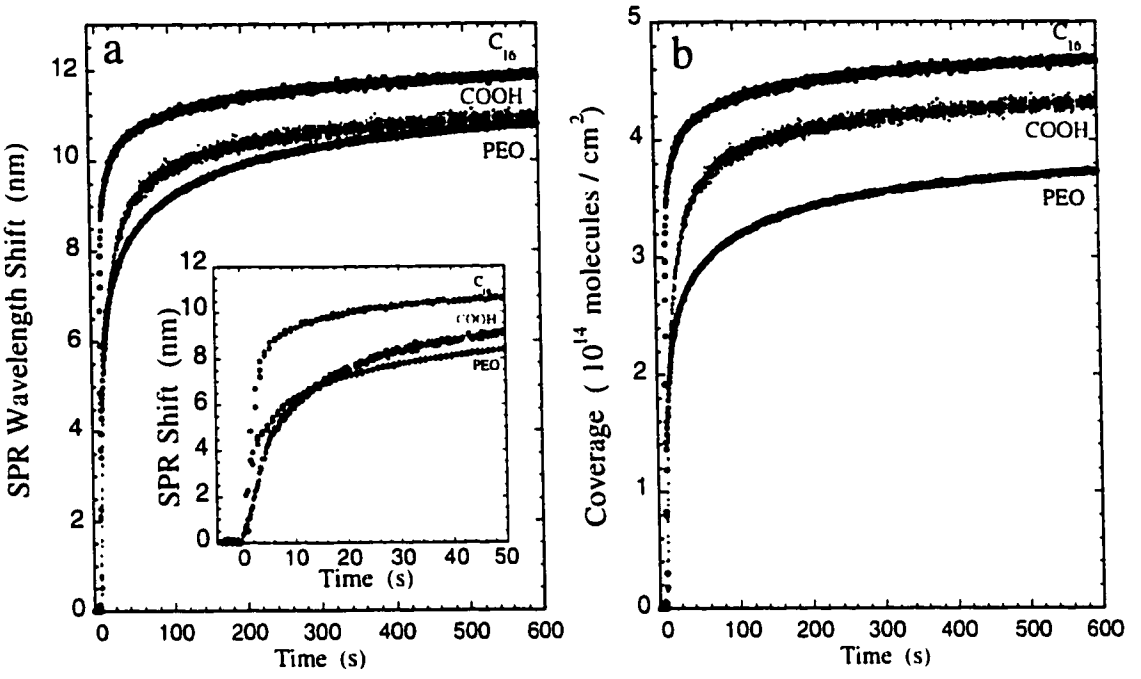


Figure 4.4 The adsorption of C_{16} (0.25 mM in ethanol), COOH-terminated thiol, $C_{15}COOH$ (0.12 mM), and PEO-terminated thiol, $C_{11}PEO$ (0.1 mM) from ethanol onto Au. (a) SPR wavelength shift versus time (inset shows a smaller time window). (b) Coverage versus time.

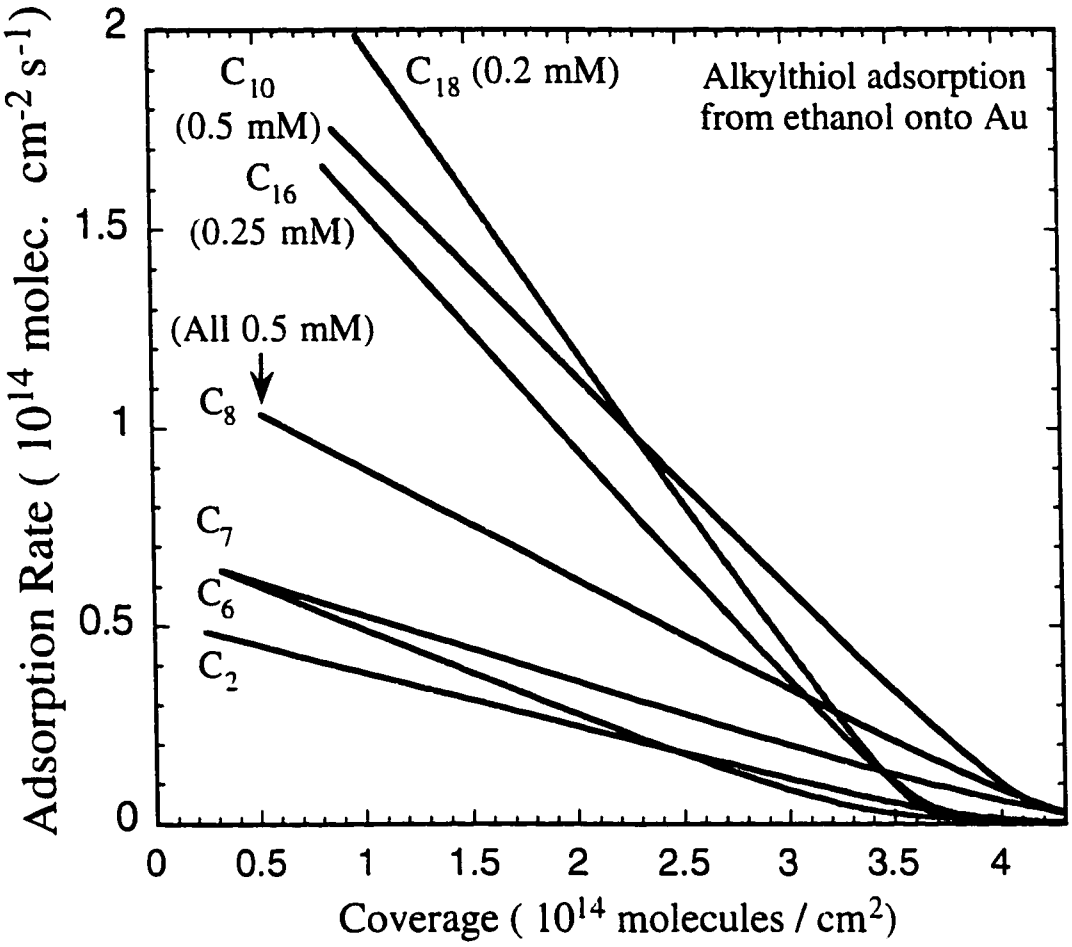


Figure 4.5 The results of Fig. 4.3 replotted as the rate of adsorption versus coverage. The rates in each case were determined from the local derivative of the coverage versus time, after fitting to a triple exponential function (see text).

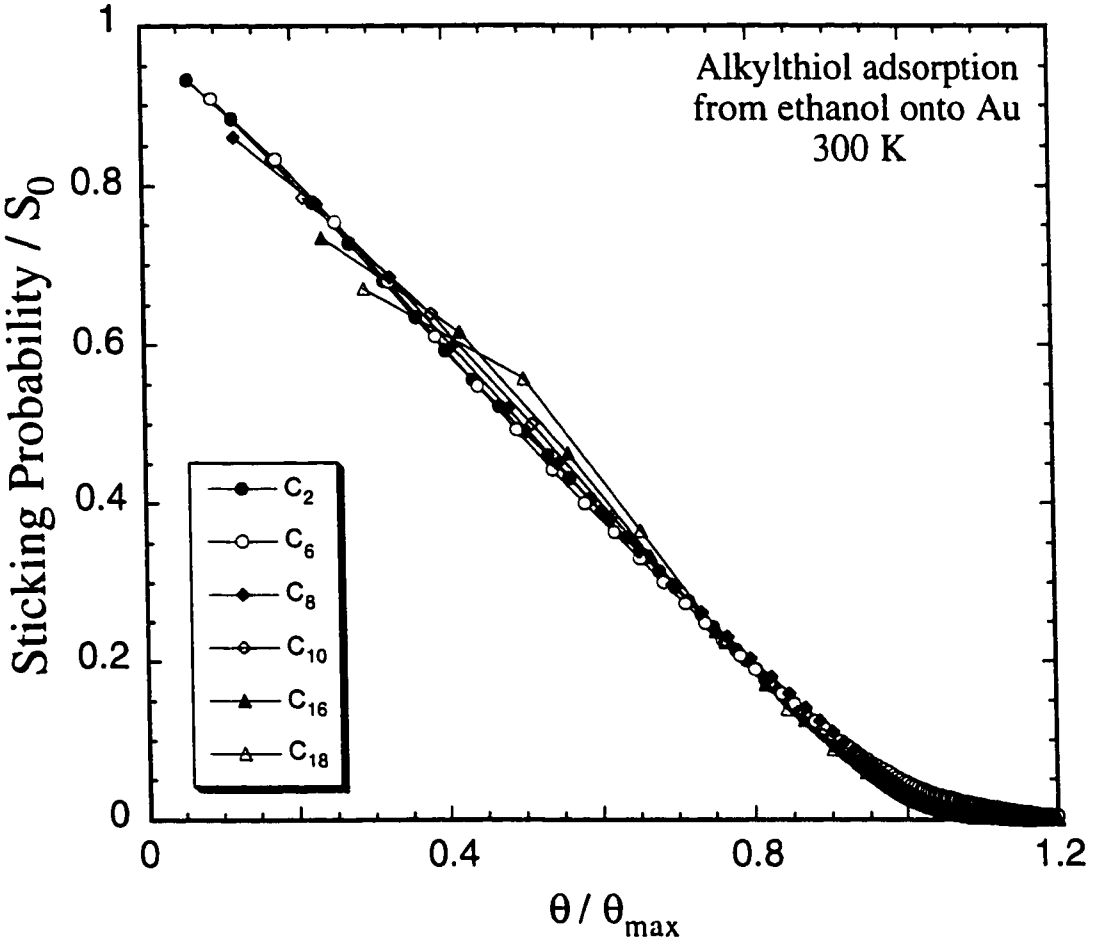


Figure 4.6 The sticking probability for selected thiols from Fig. 4.5 replotted as normalized sticking probability (S) versus normalized coverage. S_0 and θ_{max} used to normalize the data were determined for each thiol by fitting the data to a first-order Langmuir model: $S=S_0(1-\theta / \theta_{max})$.

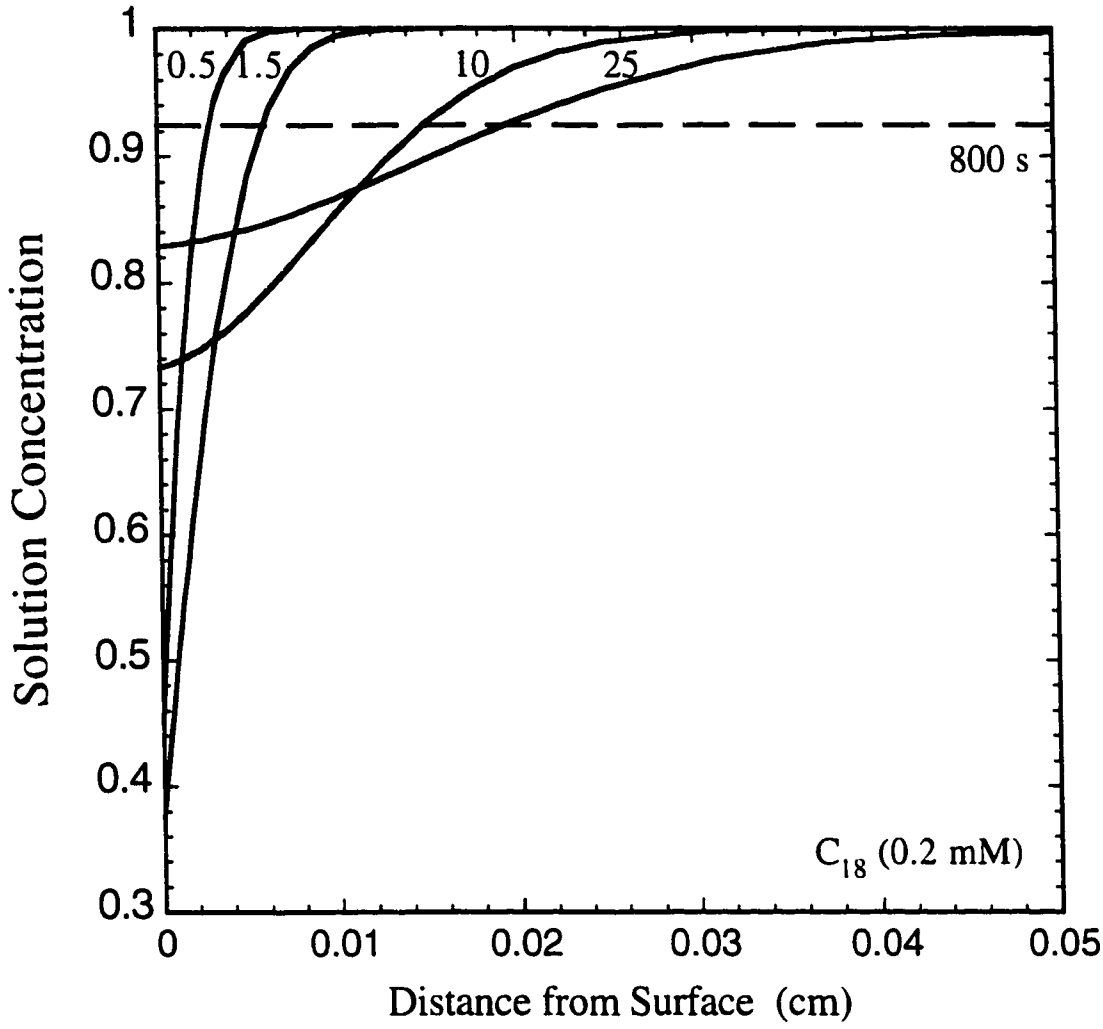


Figure 4.7 The variation of the thiol concentration versus distance from the surface at various times during the adsorption of the C_{18} thiol from 0.2 mM solution onto Au.

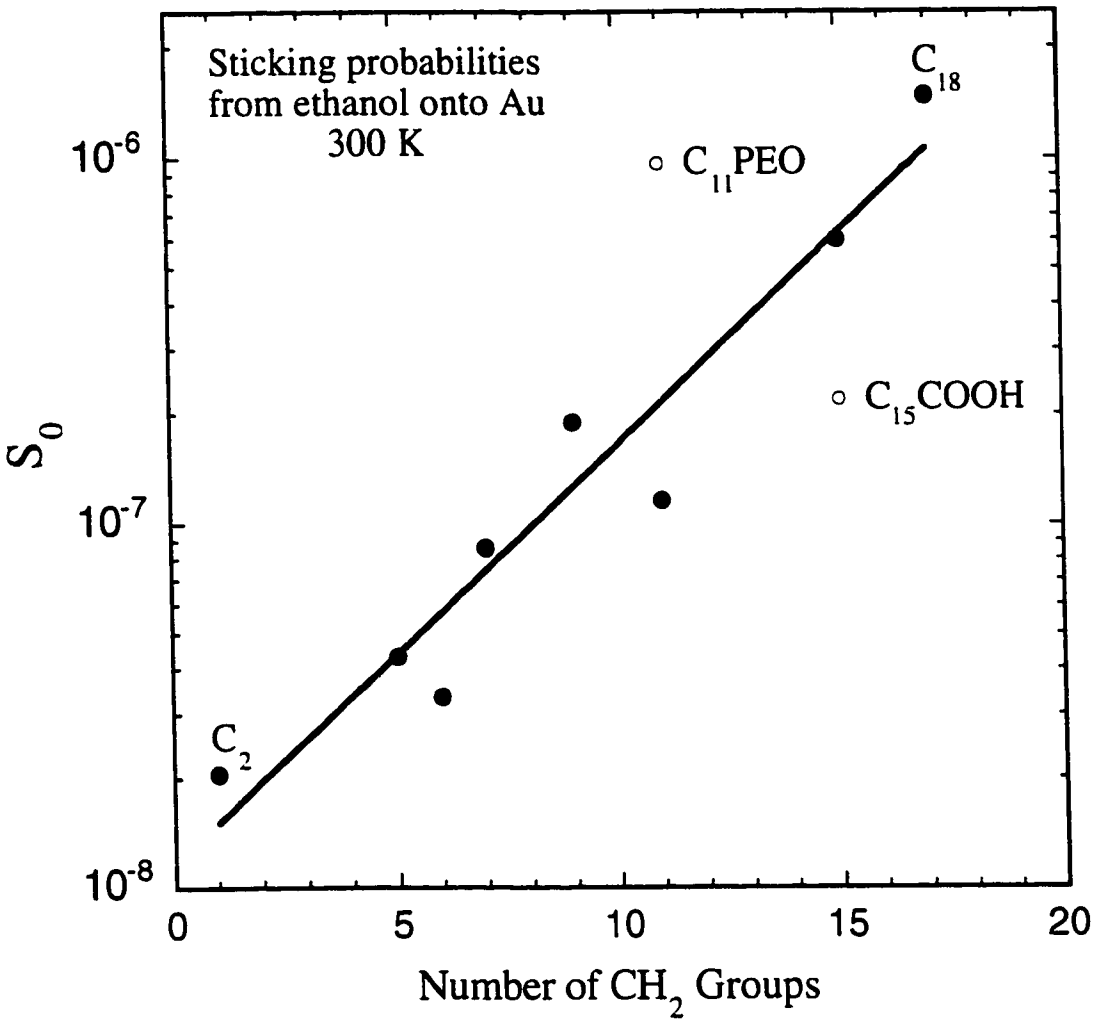


Figure 4.8 The initial sticking probability (S_0) plotted in log format versus chain length for the alkylthiols (filled circles). The open circles show S_0 for the C_{11}PEO and C_{15}COOH thiols, as labeled. The solid line shows the linear best fit to the data for the simple alkylthiols (filled circles), which corresponds to the equation: $S_0 = P_0 \bullet \exp\left(b \bullet N_{CH_2}\right)$, where N_{CH2} is the number of methylene groups in the alkyl chain (not counting the terminal CH_3), and P_0 and b are constants. Their best-fit values are $P_0 = 1.15 \times 10^{-8}$ and b = 0.27 per methylene group.

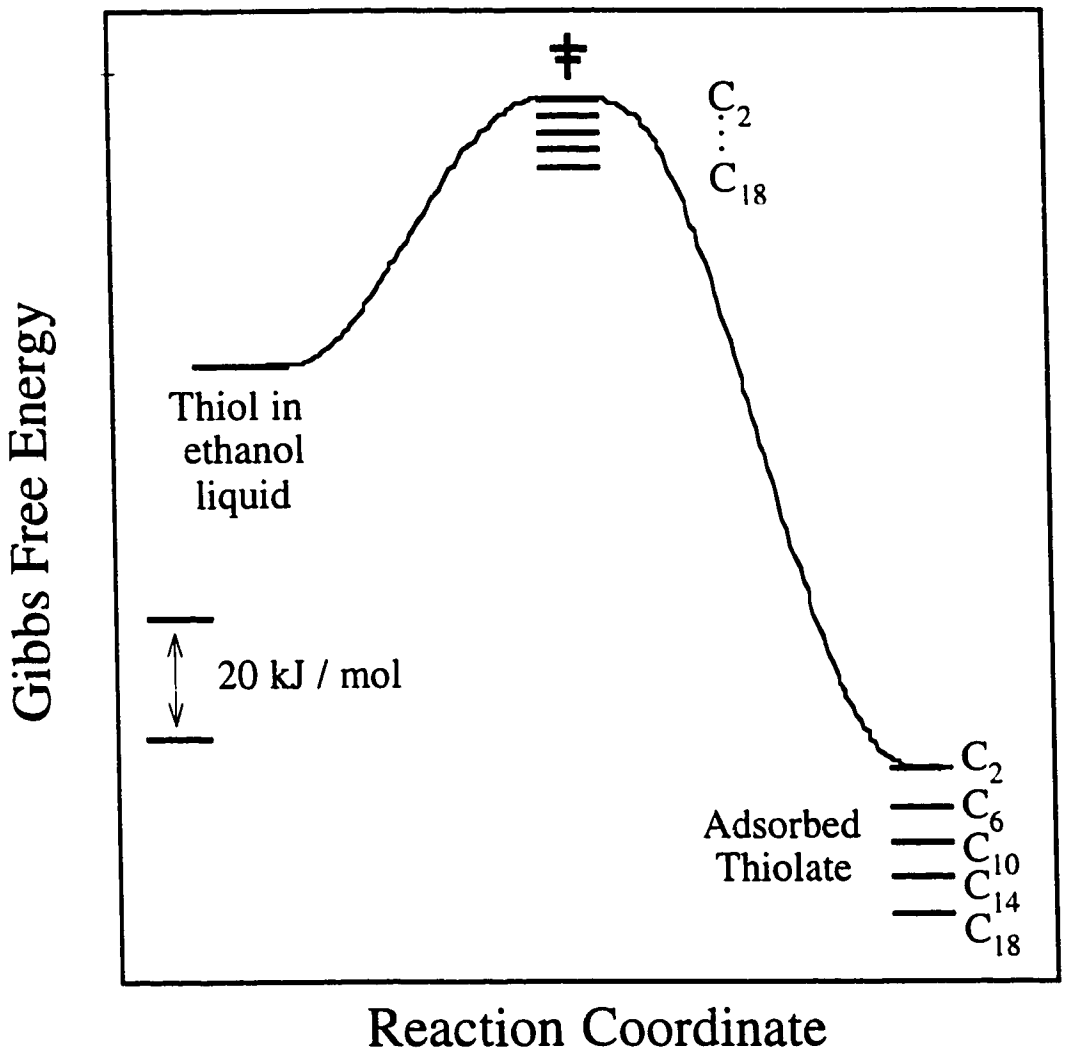


Figure 4.9 Free-energy diagram for the adsorption / desorption process of alkylthiols onto gold.

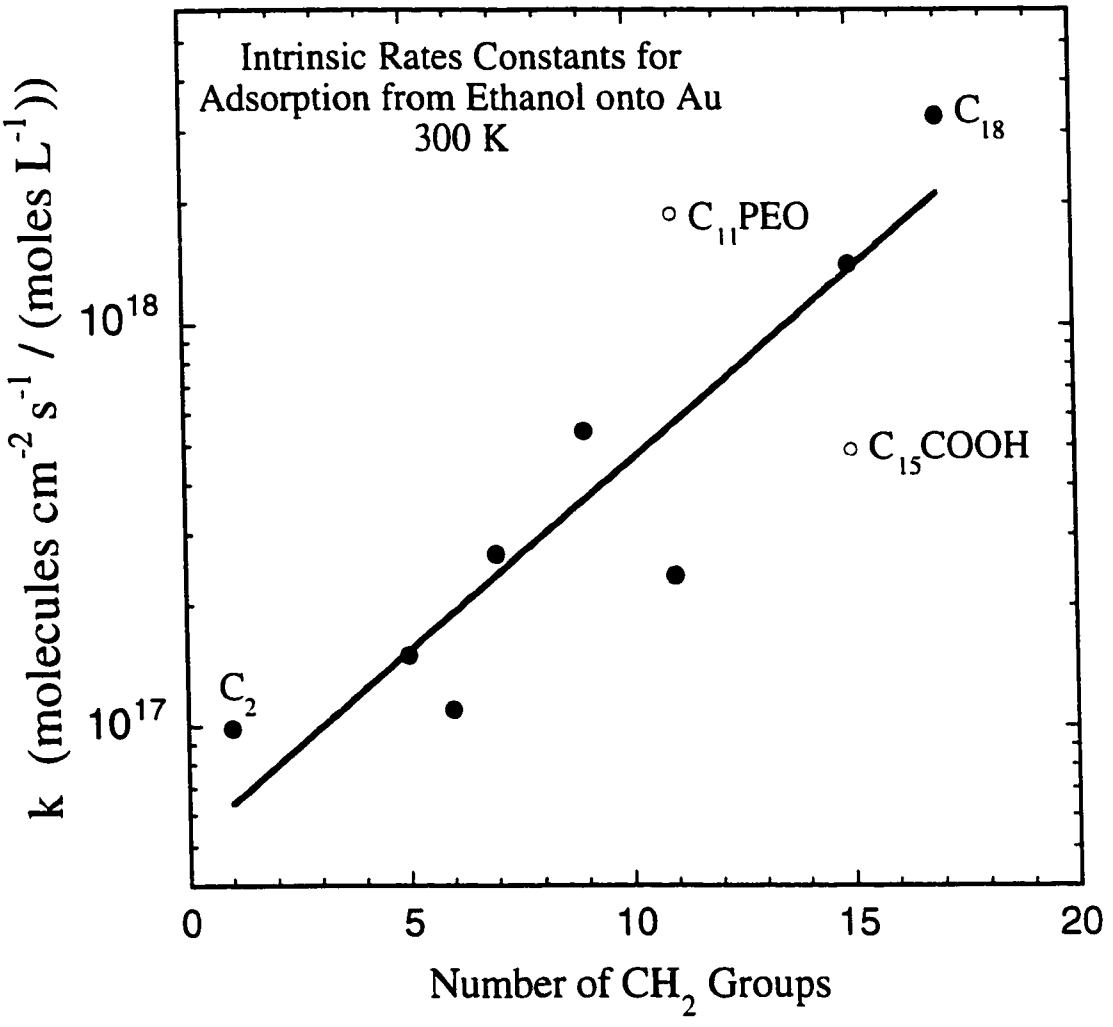


Figure 4.10 Intrinsic rate constants for adsorption, within a first-order Langmuir adsorption model, for the thiols studied here, plotted versus the number of CH_2 groups in the alkyl chains.

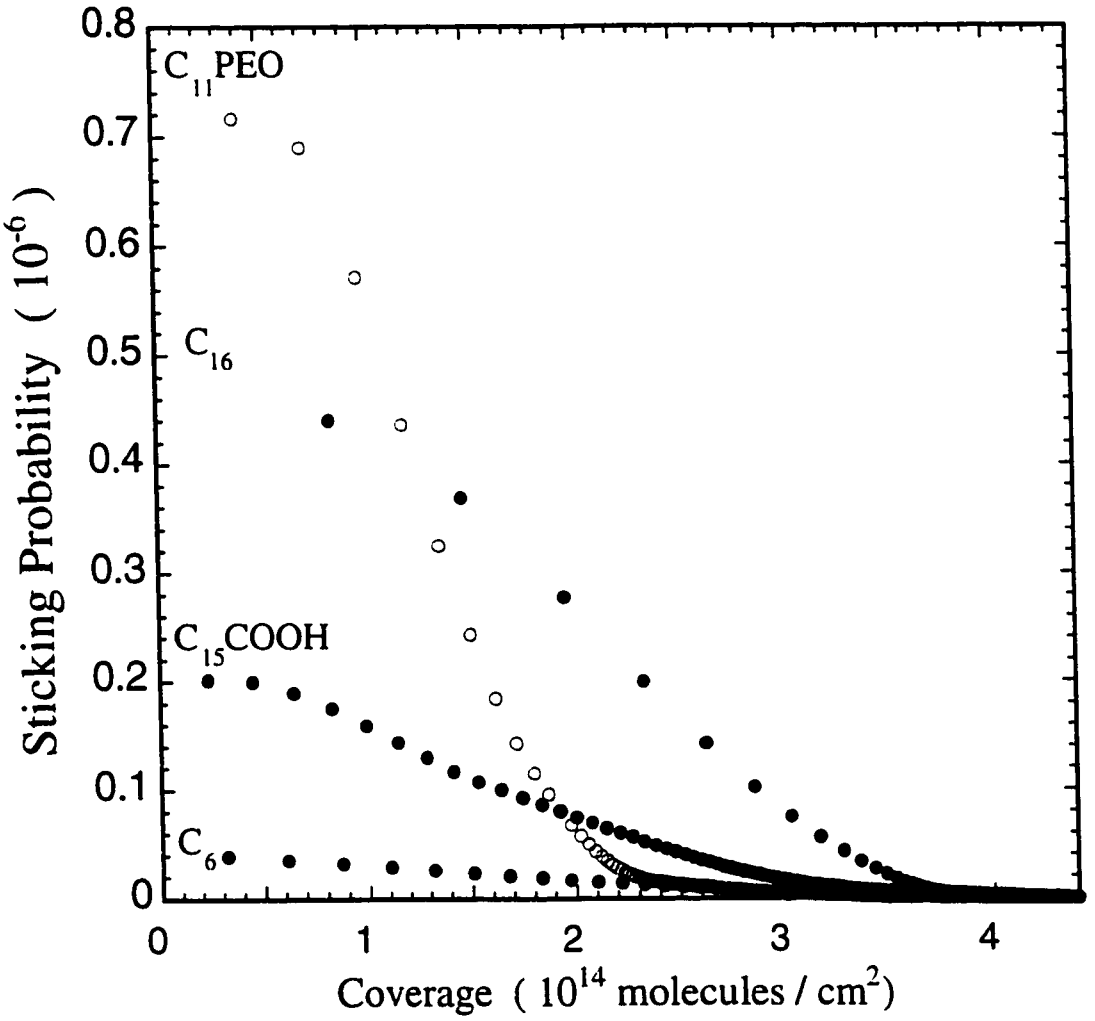


Figure 4.11 The sticking probabilities versus coverage for C_{11}PEO and C_{15}COOH. For comparison, the values for the C_6 and C_{16} alkylthiols are also shown.

CHAPTER 5: SPR MEASUREMENT OF BINDING AND DISSOCIATION OF WILD-TYPE AND MUTANT STREPTAVIDIN ON MIXED BIOTIN-CONTAINING ALKYLTHIOLATE MONOLAYERS

INTRODUCTION

The streptavidin-biotin (SA-biotin) couple is significant due to its extremely high binding affinity, K_a, which is $\sim 10^{13}$ M [1] and the fact that each streptavidin has four equivalent sites for biotins. Because of this, and because of the ease with which a wide variety of molecules can be chemically modified with biotin, the SA-biotin couple is extensively used in many different applications in biotechnology. A recent technique has demonstrated how streptavidin (SA) may be immobilized onto a surface with high coverage, specificity, and activity [2]. This technique entails assembling a biotin-containing alkylthiolate monolayer (BTM) on a Au surface. The biotin in these films is exposed to the liquid and is accessible to SA molecules in solution. SA binds to the BTMs in such as way as to expose two of its binding sites away from the surface (see Fig. 5.1). Secondary molecules chemically modified with biotin can then be rapidly and conveniently immobilized on these SA activated surfaces with minimal impact on the biological activity (e.g. specificity) of the immobilized molecules. The facile immobilization of secondary adsorbates to these prefunctionalized biotin-SA surfaces, in combination with emerging technologies for patterning the primary adsorbates, make these surfaces convenient substrates for the development of biosensors. This system has been investigated in a number of studies using a variety of BTMs [2-5].

The composition of the BTM, hence the surface biotin concentration, can be controlled by varying the percent concentration of the biotin-terminated alkylthiol in the solutions used to prepare these surfaces. The amount of SA that binds to these various composition BTMs is dependent on the percent of biotin in the film [2]. As the

percentage of biotin increases, the amount of SA that binds reaches a maximum at low biotin concentration. It has been hypothesized that a closely packed layer of biotin on the surface could sterically inhibit biotin entry into one of the SA's binding pockets [6]. SA may bind to these surfaces via either one or two such biotin linkages. We postulate that the stability of the SA within the adlayer would be higher and show improved orientation if two biotin linkages are formed. Enhanced stability and orientation could be advantageous in certain applications related to biosensor technologies. Here, we demonstrate that SPR can be used together with genetically engineered SA to interrogate how the stability of the surface-bound SA (and hence the number of SA-biotin links) depends upon the composition of the biotin-containing alkylthiol monolayer. The results further verify that the surface biotin-SA couple is truly specific in nature, as was evidenced earlier using related techniques [7]. The specific alkylthiols used to prepare these mixed monolayers were: biotinylated alkylthiol (BAT), poly(ethylene oxide) (PEO), and methyl-terminated hexadecanethiol (MHD). The monolayers were a binary mixture of BAT with either the PEO or the MHD. Monolayers containing 100% of the PEO almost totally resisted binding of SA, consistent with other reports obtained with similar PEO containing alkylthiol monolayers exposed to a variety of proteins [8,9].

The stability of the surface-bound SA is measured here by studying its desorption kinetics with SPR. Desorption (i.e., cleavage of the surface biotin-SA bond) is initiated in homogeneous solution by the addition of a buffer solution containing an excess of free biotin. The excess of biotin in solution competes with the surface biotin for the SA binding sites. Due to the extremely long half-life of wild-type SA-biotin dissociation, ~63 hours (at 23° C) [10], desorption was measured for both wild-type streptavidin (WT) and an engineered streptavidin binding mutant (W120A), which contains a single amino acid substitution at a residue shown to be important for high-affinity binding[10]. W120A dissociates with a half-life ($t_{1/2}$) of less than one minute in homogenous buffer with excess biotin. (A more accurate $t_{1/2}$ could not be determined for W120A, as it is too rapid to measure with the technique described in [10].) These

engineered SA mutants were developed in our laboratory for the biophysical characterization of the SA-biotin interaction in homogeneous solutions [10,11]. In collaboration with others [4], we have also used these engineered SA mutants in other studies to characterize the binding of SA to biotin functionalized surfaces.

SPR is well suited for these experiments as it provides a means for surface-sensitive, in-situ measurements. The instrument used in these studies can detect protein coverages down to ~ 10^{-3} monolayer (for a system with a typical monolayer packing density of 250 ng / cm^2) [12] with a few seconds time-response, allowing a comparison between the desorption behavior of both the WT and the W120A as a function of the surface biotin composition.

EXPERIMENTAL

The data presented in this chapter was taken using the planar SPR system as described in the Chapter 2. The measured SPR shifts were converted into thicknesses with the formalism presented in Chapter 3.

RESULTS

BAT / PEO MIXED MONOLAYERS

Streptavidin adsorption and removal curves on BAT / PEO mixed alkylthiolate monolayers at two of the biotin concentrations studied and for a pure PEO alkylthiolate layer are shown in Fig. 5.2. Many other concentrations were also studied, as will be discussed in the next chapter. As shown in curve (e), the pure PEO layer resisted

the binding of WT, except for a few percent of a monolayer, which readily rinsed away with buffer. This is consistent with the general resistance for PEO-functionalized surfaces to protein adsorption [8,9]. Both WT and mutant streptavidin adsorbed to the BTMs within seconds after introducing the protein solution into the flow cell. After this initial rapid adsorption, which accounted for ~80% of the maximum, the amount of bound protein more slowly reached an apparent saturation. The half-lives for adsorption were found to be ~5 seconds for WT and ~6 - 17 seconds for W120A (depending on % of BAT on the surface). Details of these adsorption kinetics will be presented and discussed in the next chapter. The amount of adsorbed WT increased with increasing surface biotin concentration and reached a maximum of ~230 ng / cm^2 (or 3800 $Å^2$ / molecules.) when the liquid thiol solution contained ~10% biotin (see Fig. 5.2b). This amount is ~80% of the expected density of a two-dimensional C(2,2,2) crystalline SA monolayer of approximately 280 ng / cm^2 (or 3100 $Å^2$ / molecule) [13]. At very low biotin surface concentration, the amount of W120A that adsorbed was considerably less than the amount of WT that bound to an identical monolayer. For example, a solution containing 1% biotin led to a monolayer which bound only 30% as much W120A as WT (Fig. 5.2c, d). As the biotin composition increased to 10%, the amount of W120A bound was ~90% as much as WT. (The biotin compositions quoted throughout this chapter refer to the percent biotin in the thiol solutions used to assemble the monolayers. The actual amount of biotin on the surface was determined by XPS elsewhere [14] and can be approximated by $y = 1.10x^{0.46}$ for PEO and $y = 1.05x^{1.3}$ for MHD, where x refers to the fraction of thiols in liquid solution which are BAT, and y refers to the fraction of thiolates in the resulting mixed alkylthiolate monolayers which are BAT.)

A dramatic difference in the desorption behavior between W120A and WT is readily observed for all compositions of BTMs studied, as shown for the 1% and 10% BAT / PEO layers in Fig. 5.3. The wild-type SA barely shows any measurable desorption for these adlayers during the time frame of Fig 4.3., but the initial desorption rate is measurable if followed for 30 minutes (not shown). These initial rates correspond

to decay rate constants of $3.6\pm0.5\times10^{-5}$ s^{-1} and $1.1\pm0.9\times10^{-5}$ s^{-1} for 1% and 10% BAT, respectively, assuming first-order exponential decay. In contrast, nearly all the W120A desorbs after the addition of the buffer solution containing 1 mM biotin within only 20-200 seconds. As shown in Fig. 5.3 for W120A, the desorption rate of surface-bound SA also depends strongly on the biotin surface coverage. Half-lives of about 3.0±0.5 and 13±1 s were observed for W120A, with 1% and 10% biotin, respectively. The rate constants, k, mentioned above for WT correspond to half-lives, $t_{1/2}$ = (ln2) / k, of 19,000±2000 and 32,000-320,000 s for 1% and 10% BAT, respectively. The desorption data are better fit using a double exponential kinetic model as shown in Fig. 5.3 where both a single and double exponential fit is shown for curve(c). This suggests that more than one process actually may be occurring, or that lateral SA-SA interactions in the monolayer affect stability. A detailed, quantitative analysis of these trends and of other streptavidin binding mutants will be presented in the next chapter.

BAT / MHD MIXED MONOLAYERS

The behavior was quite different for BTMs containing BAT / MHD as compared with the BAT / PEO alkylthiolate mixtures. SA adsorption onto a BAT / MHD monolayers is no longer dominated by the specific biotin-SA coupling. A typical result is shown in Fig. 5.4, where W120A is dosed to a BTM prepared from a 20% BAT / 80% MHD solution. (This solution gives a *surface* concentration of BAT that is closer to that prepared from a 10% BAT / PEO mixture [14], for comparison.) When MHD replaces PEO in the monolayer, W120A adsorbs to about the same maximum concentration, but much more of the adsorbed W120A can be removed with a biotin-free buffer rinse, as shown after the asterisk (*) in Fig. 5.4. Furthermore, when rinsed with buffer containing excess biotin to initiate competitive desorption (second rinse in Fig. 5.4), a large fraction of the bound SA remains on the surface for long times (note the difference in time scales for Figs. 4.3 and 4.4). Both the SA which rinses in biotin-free buffer and the SA which

cannot be removed with excess biotin correspond to forms of adsorbed SA that are not bound to the surface by specific SA-BAT bonds. For example, as shown in Fig. 5.4, this corresponds to >70% of the adsorbed SA on the 20% BAT / MHD layer, and almost all of the SA on the 100% MHD layer. Clearly, if one desires a specific attachment of SA to the surface for some sensor application, the BAT / PEO monolayer is far superior to the BAT / MHD monolayer. The non-specific binding of the SA to the BAT / MHD monolayers might, in principle, render the SA monolayer less effective for further functionalization which depend on recognition (e.g., attachment of biotinylated antibodies for immunoassay).

DISCUSSION:

Varying the surface concentration of biotin in mixed alkylthiolate monolayers containing PEO terminated alkylthiolates had two marked effects. As the ligand concentration increased (up to 10% biotin), the total amount of protein that adsorbed to these films increased. This is most likely a result of the limited number of surface binding sites at these low surface biotin concentrations. At very low biotin concentrations in the BTM (~1%), significantly less W120A binds relative to wild-type. This observation is tentatively attributed to a rapid equilibrium that can establish for low biotin concentrations between adsorbed and solution-phase mutant SA on the surface. The rapid off-rate kinetics observed upon addition of biotin-containing buffer prove that the mutant binding to such a surface is very weak. Furthermore, a rather substantial fraction (~30%) of the W120A rapidly desorbed from surfaces with very low biotin concentration when rinsed in pure buffer (i.e. even in the absence of solution-phase biotin). Thus, it is reasonable to assume that an equilibrium surface concentration is established upon dosing SA to the 1% BAT / PEO layer which only reaches ~16% of a monolayer. In contrast, the high affinity WT binds essentially irreversibly to such a surface, and therefore establishes an equilibrium surface concentration much closer to a

full monolayer (~60%, probably limited by the number of biotin sites). As the amount of ligand on the surface increases, there is a greater likelihood that SA may bind to the surface via two binding sites. These doubly bound species would be expected to exhibit a slower off-rate in comparison to singly bound species, and indeed slower rates are observed. This additional stabilization provided by two biotin linkages per SA may explain why the amount of adsorbed W120A approaches that of WT at higher biotin coverages. This may also explain the dependence of the competitive desorption rate upon biotin surface coverage which was observed with W120A. Other experiments for elucidating the mechanism of stabilization of surface-bound streptavidin, such as changing the bulk protein concentration, are in progress. The desorption kinetics of other binding mutants are also currently under investigation.

Note that at low biotin concentration, the desorption rate of the surface-bound SA was extremely fast (half-life = 3.0 s) compared to the rates at higher concentrations, such as in Fig. 5.4(c). Thus, the double exponential behavior of Fig. 5.3(c) is clearly not attributable to an additional time constant associated with diffusion of the desorbed SA through the liquid solution. A reaction scheme that could explain the double exponential behavior observed for the dissociation rate will be presented in more detail by Pérez-Luna et al. [4]. In this scheme, it is assumed that initially, there are two populations of adsorbed SA. One fraction of the SA will be attached to the surface through one biotin-SA bond and the other fraction of SA will be attached to the surface through two biotin-SA bonds (See Fig. 5.5). Since the biotin binding pockets are located in pairs at opposite sides of the SA molecule, we can assume that at the most, one SA will bind to two biotin groups on these BTMs. With this scheme, desorption of the adsorbed SA molecules could take place by: a) dissociation of one biotin-SA bond, for those molecules that are attached to the surface through one biotin-SA bond, (step 1 in Fig. 5.5), and b) simultaneous dissociation of two biotin-SA bonds (step 2 in Fig. 5.5), for SA attached to the surface through two biotin groups. Alternatively, SA that is attached to the surface through two biotin-SA bonds could convert to SA that is attached to the

surface through one biotin-SA bond (step A) which may subsequently dissociate by step 1 in Fig. 5.5. If we denote the initial fraction of SA that is attached to the surface through one biotin bond as $[S_1]_0$, and the initial fraction that is attached to the surface through two biotin bonds as $[S_2]_0$, then Pérez-Luna et al. [4] find that the total amount of protein on the surface, [S] is given at time t after the start of the competitive desorption by:

$$[S] = \{[S_1]_0 - \frac{k_A}{k_1 - k_A - k_2}[S_2]_0\}e^{-k_1 t} + \{\frac{k_A}{k_1 - k_A - k_2}[S_2]_0\}e^{-(k_A + k_2)t}$$

This gives the observed double exponential decay. The above mentioned model does not take into account the attractive or repulsive interactions that probably occur between adjacent adsorbed protein molecules, or between proteins and other groups (e.g. unbound biotin) on the surface, which would further complicate the kinetics. The model also does not take into account an equilibrium which might exist between the double bound and single bound SA. In the next chapter, we present a model which incorporates this equilibrium and does not include this alternative desorption pathway which requires the simultaneous dissociation of two biotin linkages to one SA.

From Fig. 5.4 we found that the half-life of the WT was about ~10^4-fold longer than that of the mutant W120A for the lowest biotin concentration (1%) in mixed monolayers with PEO. This low concentration is probably most relevant to addressing the specificity and affinity of the surface biotin-SA coupling by comparison to kinetics in homogenous solution, since here the interaction should be dominated by single biotin-SA bonds, as is probed in the reported off-rate kinetics in homogenous solution. Solution phase experiments showed that W120A has a half-life that is > 3600-fold faster than WT [10], consistent with the ratio of 10^4 measured in this dilute surface monolayer. This further confirms the specific nature of the surface biotin-SA bond. Further details of the composition dependence of the SA off-rates will be presented in the next chapter.

CONCLUSIONS

As a measure of stability of surface-bound SA, SPR was used to measure the adsorption and competitive desorption kinetics of WT and the W120A mutant on mixed alkylthiolate monolayers. The desorption rate was found to decrease with increasing surface coverages of biotinylated alkylthiolate (up to 10% in solution) in a mixed binary monolayer containing poly(ethylene oxide) terminated alkylthiolate. The competitive desorption rates of W120A, compared to that for WT, confirms the specificity of SA binding to these functionalized gold surfaces. The results of SPR experiments show that the stability of SA modified sensors can be tuned through the use of binding mutants and tuned through controlling the percentage of biotin sites in the primary monolayer. This has potential applications in renewable surfaces for biosensors. Furthermore, it is shown that mixed alkylthiolate monolayers containing hydrophobic, methyl-terminated thiolate are much less specific in the way in which they bind adsorbed streptavidin. In this case, attachment to the surface is dominated by bonds which are not of the type used in biotin-SA recognition.

NOTES TO CHAPTER 5

[1] Green, N. M. *Adv. Protein Chem.* **1975**, *29*, 85-133.

[2] Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

[3] Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

[4] Perez-Luna, V. H., O'Brien, M. J., Opperman, K. A., Hampton, P. D., Stayton, P. S., Klumb, L, Lopez, G. P. *(in preparation)* .

[5] Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

[6] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

[7] Muller, W.; Ringsdorf, H.; Rump, E.; Wildburg, G.; Zhang, X.; Angermaier, L.; Knoll, W.; Liley, M.; Spinke, J. *Science* **1993**, *262*, 1706-8.

[8] Wang, R. L. C.; Kreuzer, H. J.; Grunze, M. *J. Phys. Chem. B* **1997**, *101*, 9767-9773.

[9] Prime, K.; Whiteside, G. M. *J. Am. Chem. Soc.* **1993**, *115*, 10714-10721.

[10] Chilkoti, A.; Boland, T.; Ratner, B. D.; Stayton, P. S. *Biophys. J.* **1995**, *69*, 2125-30.

[11] Chilkoti, A.; Tan, P. H.; Stayton, P. S. *Proc. Natl. Acad. Sci. USA.* **1995**, *92*, 1754-8.

[12] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[13] Darst, S. A.; Ahlers, M.; Meller, P. H.; Kubalek, E. W.; Blankenburg, R.; Ribi, H. O.; Ringsdorf, H.; Kornberg, R. D. *Biophys. J.* **1991**, *59*, 387-96.

[14] Nelson, K. E.; Jung, L. S.; Gamble, L.; Boeckl, M.; Naeemi, E.; Campbell, C. T.; Castner, D. G.; Stayton., T. S. P. *Langmuir (submitted)* **1998**.

FIGURES FOR CHAPTER 5

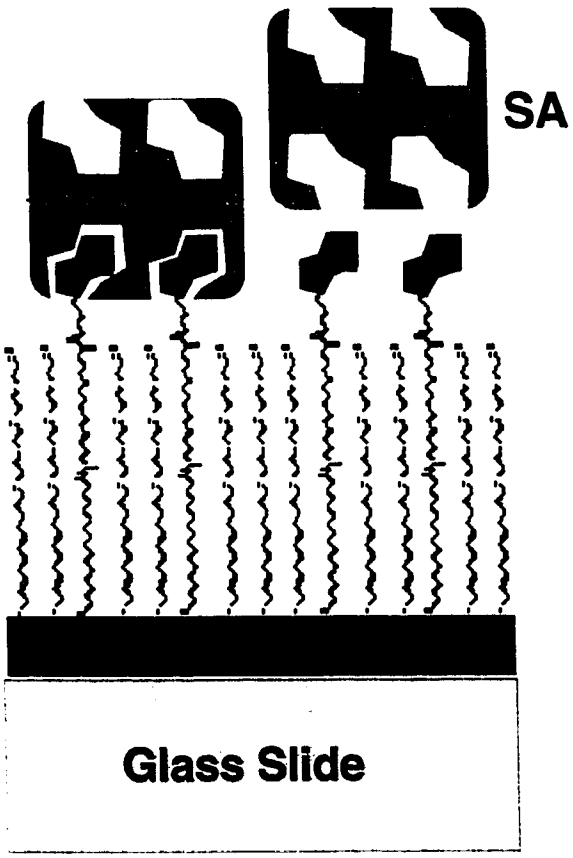


Figure 5.1 Idealized depiction of mixed BAT / PEO BTMs and streptavidin (not to scale). The biotin moiety of the biotinylated thiol inserts ~14 Å into the streptavidin binding pocket. Note that the thiol molecules are not expected to display such high order and rigid orientation.

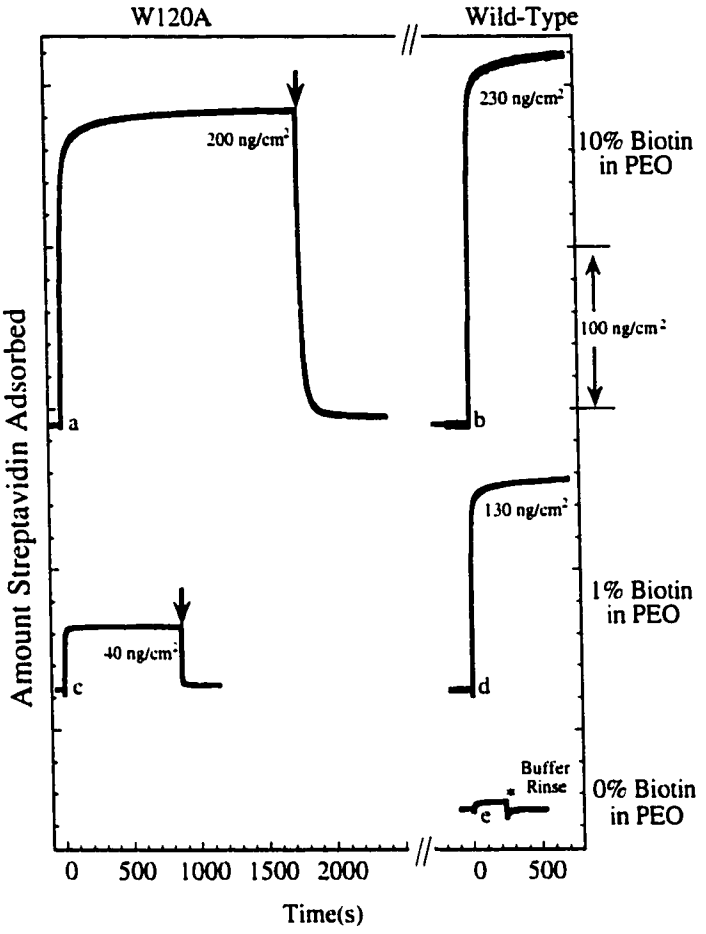


Figure 5.2 Absorption and desorption of W120A and wild-type streptavidin on mixed BTMs, measured by SPR. Arrows indicate when biotin rinses were performed. Numbers below each individual spectrum indicate the maximum amount of protein adsorbed, in ng / cm^2. Mutant and wild-type binding curves for identical BTMs are shown side-by-side. For different surfaces spectra are offset for clarity; zero (adsorbed) is indicated by the baseline prior to injection of protein. (a) W120A on 10% BAT / PEO BTM, (b) WT on 10% BAT / PEO BTM, (c) W120A on 1% BAT / PEO BTM, (d) WT on 1% BAT / PEO BTM. (e) WT on 100% PEO (Note: WT does not appear to appreciably adsorb to this surface, the * indicates a buffer rinse with no free biotin.) See Fig. 5.4 and text for WT rinsing data. (The "% BAT in PEO" here refers to the relative amount of BAT to total thiol (BAT+PEO) in the liquid solution used to assemble the monolayers.)

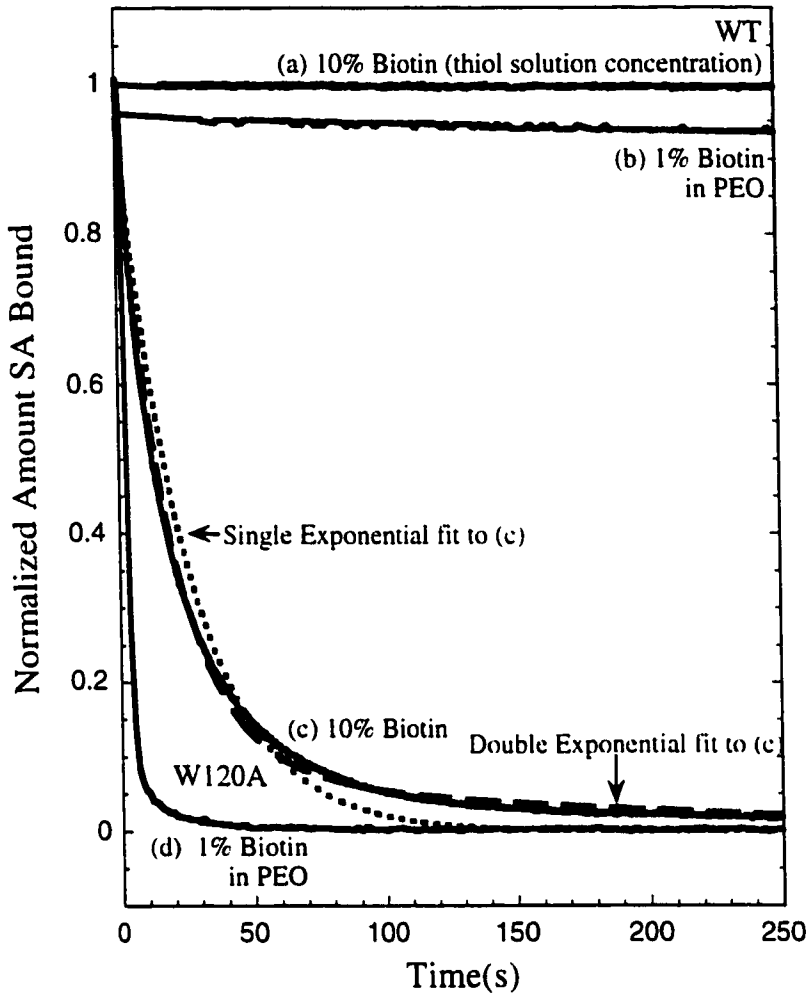


Figure 5.3 Solid curves: Competitive desorption of W120A and WT SA from mixed BTMs, induced by injection biotin-containing buffer at time 0, measured by SPR. The amount of SA adsorbed (upon injection of the biotin-containing buffer) of each sample was normalized to the saturation amount seen in the prior adsorption step. (a) WT on 10% BAT / PEO BTM, (b) WT on 1% BAT / PEO BTM, (c) W120A on 10% BAT / PEO BTM, (d) W120A on 1% BAT / PEO BTM. Note that the WT on 1% BAT appears to start at ~0.96. This is due to the fact that ~4%, presumably non-specifically bound to PEO, and was removed instantly (Ö1 s) upon rinsing. (The "%BAT" here refers to the % of BAT relative to the total thiol in the liquid solutions used to assemble the monolayers.) Dotted curve ( ): single exponential decay fit to curve (c). Dashed curve (- - -): double exponential decay fit to curve (c).

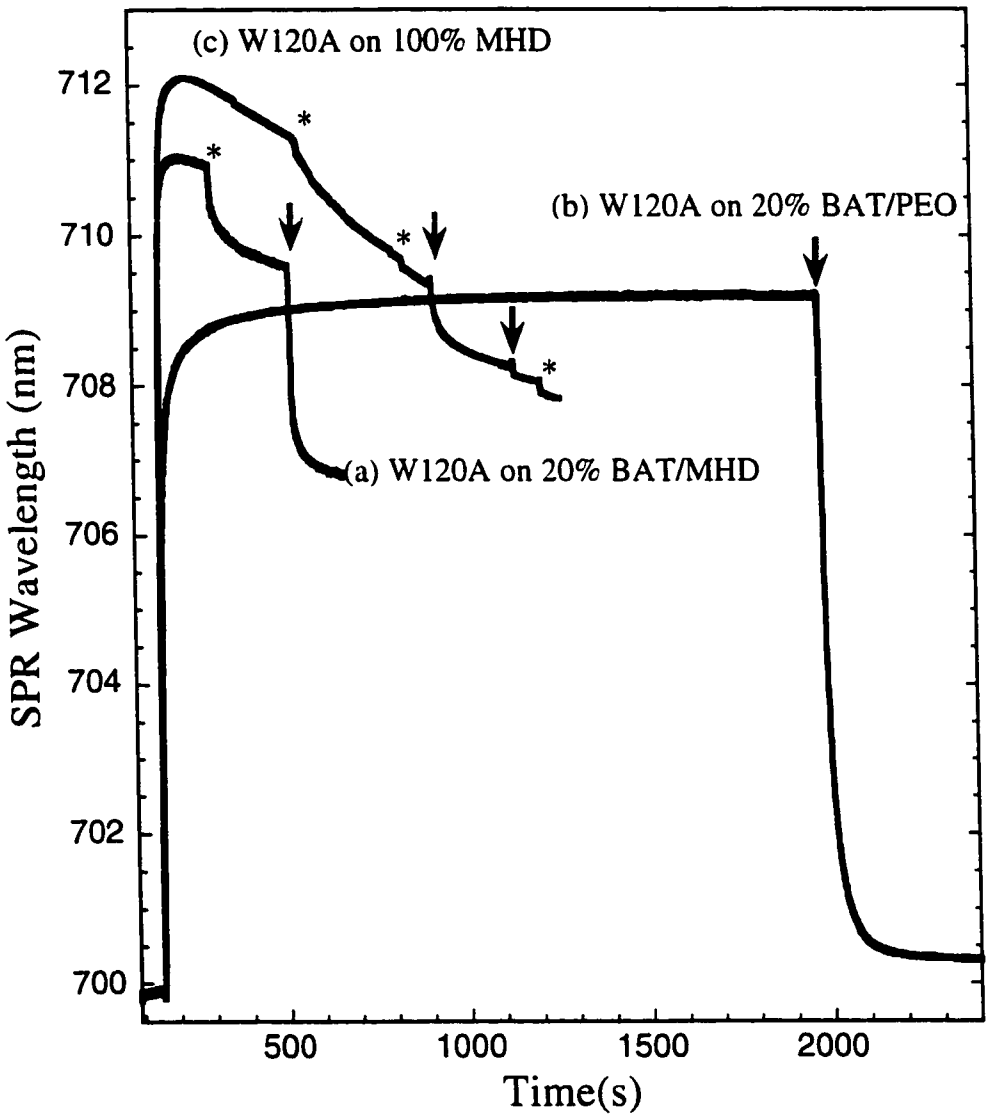


Figure 5.4 Adsorption and desorption of W120A on two different mixed BTMs, and on 100% MHD, measured by SPR. Arrows indicate where biotin-containing buffer was injected and stars (*) indicate a special rinse with PBS buffer without biotin. (a) W120A on 20% BAT / MHD. (b) W120A on 20% BAT / PEO. (c) W120A on 100% MHD. (The "% BAT" here refers to the % of BAT relative to the total thiol in the liquid solutions used to assemble the monolayers.)

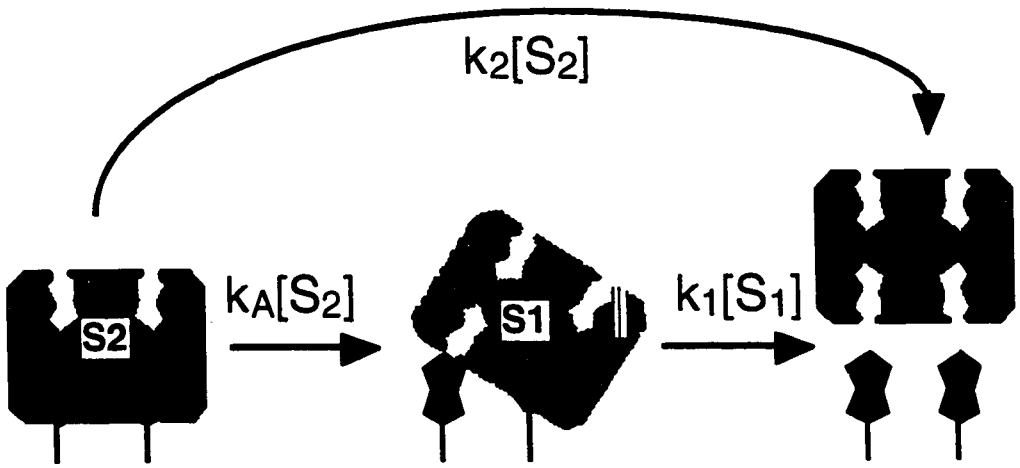


Figure 5.5 Reaction scheme for dissociation of SA from biotin linkages on the surface. (Courtesy of V. Perez-Luna)

CHAPTER 6: KINETIC ANALYSIS OF BINDING AND DISSOCIATION OF WILD-TYPE AND MUTANT STREPTAVIDIN ON MIXED BIOTIN-CONTAINING ALKYLTHIOLATE MONOLAYERS

INTRODUCTION

This is an extension of the work presented in the previous chapter examining the structure and stability of surface bound streptavidin. There, the adsorption and desorption kinetics of wild-type (WT) streptavidin (SA) and its W120A mutant at biotin functionalized surfaces were compared. The desorption (i.e., cleavage of the surface biotin-SA bond) was initiated in homogeneous solution by switching from pure buffer above the surface to a buffer containing an excess of free biotin. This biotin competes with the surface biotin for the SA binding sites, thus initiating a competitive desorption process. In that chapter, the main surface studied contained various concentrations of biotin-terminated alkylthiols (BAT) in mixed monolayers wherein the remainder of the surface was saturated with PEO-terminated alkylthiolates (PEO). Varying the concentration of biotin-terminated alkylthiol (BAT) from 0% - 10% in the thiol solution used to prepare these mixed monolayers had two marked effects. First, the total amount of protein that adsorbed increased from nearly zero as the ligand concentration increased for low concentrations, proving that biotin surface sites are needed to bind the SA, and that SA doesn't bind to the pure PEO surface. Second, as the amount of ligand on the surface increased, the desorption rates decreased. This is expected as there is a greater likelihood that SA will bind to the surface via two binding sites with the increased ligand concentration and such doubly bound SA is expected to be more stable and therefore exhibit a slower off-rate in comparison to singly bound species. In addition, the desorption kinetics demonstrated that the binding of W120A was much, much weaker than the WT. Significantly less W120A binds relative to wild-type (WT) at

lower biotin concentrations, but, at higher biotin coverages, the amount of adsorbed W120A approaches that of WT SA. This was also attributed to the greater likelihood of the SA forming two biotin linkages to the surface at these higher biotin coverages, thus stabilizing it to the surface.

In this chapter, the full range of surface concentrations of BAT from 0 - 100% is examined in greater detail. In addition, two other engineered mutants, N23E and S27A, are included in this study. These two mutants disrupt the hydrogen-bonding between the SA and the biotin, which is considered to contribute significantly to the binding energy [1]. Based on data in previous homogeneous solution experiments [1], these mutants are expected to have an off-rate in solution of 1.2×10^{-3} s^{-1} and 1.6×10^{-3} s^{-1} for S27A and N23E respectively at 25° C [1]. These values are ~10^3 times faster than that expected for the WT, also from those experiments, of 4.0×10^{-6} s^{-1}. The weakest binding of these four variants, W120A, desorbs too quickly for their measurements and therefore must have an off-rate < one minute and has been indirectly estimated to be ~23 s in solution at 25° C [1].

The kinetics of adsorption and the competitive desorption of WT SA and these three SA mutants on mixed BAT / PEO monolayer are compared in order determine if surface layers can be prepared which allow rapid, quantitative testing of the effects of protein mutations on their binding strengths to ligands. In addition, in-depth analysis of the kinetics as a function of surface biotin concentration, X_{BAT}, clarifies how the SA is binding to the surface. A mechanism for the desorption of the SA variants on these monolayers, which explains the kinetics and saturation coverages observed, is proposed here. It involves a single, first-order rate constant for breaking a bond between the surface-bound biotin and a SA, whose value varies with mutation in a very similar way to that found in homogeneous solution. In addition, there is a fast reverse rate for reforming the bond to the surface biotin which was cleaved from a doubly-bound SA, and its rate constant is roughly independent of the mutant, as is the adsorption rate. The ratio of doubly- to singly- bonded SAs varies with increasing biotin surface concentration, from

nearly zero, to a large value, and back down to nearly zero at >60% biotin. These results further support the conclusions of earlier work which suggests that the quality of the bound SA layer is strongly dependent on the composition of the BAT / PEO thiolate monolayer [2,3]. The binding of SA as singly-bound species at the highest biotin concentrations, however, has not been mentioned previously, nor have the rate constants for the elementary steps been previously evaluated, as we do here. The mechanism used here differs substantially from that reported elsewhere [4].

METHODS

The experimental methods are described in the Chapter 2 and are the same as those used in the previous chapters. The measured SPR shifts were converted into coverages or thicknesses with the formalism presented in Chapter 3. The S27A and N23E mutants were prepared as previously reported [1,5] and provided by Patrick S. Stayton.

RESULTS

SATURATION COVERAGES OF SA

As mentioned above, the relative amount of SA bound to the BAT / PEO monolayers varied with the surface biotin concentration. Fig. 6.1a. shows the saturation SPR shift observed for various monolayer compositions upon exposure to the buffer solution containing WT SA at a concentration of 0.05 mg / ml. The monolayer composition here is indicated as the mole fraction of BAT in the original BAT+PEO alkylthiol mixture (in ethanol solution) used to prepare the monolayer. The same data are

shown in Fig. 6.1b, where instead the monolayer composition is expressed in terms of the surface biotin composition (mole fraction in the BAT / PEO mixture on the surface). The solution biotin concentration was converted to surface biotin concentration based on XPS measurements of the surface elemental composition of films prepared in the identical manner, and presented in [2]. Since nitrogen is only present in the biotin moiety of the BAT, and not in the PEO, the mole fraction of nitrogen detected by XPS provides a measurement of the concentration of biotin on the surface. All further references to the biotin composition will refer to the mole fraction of biotin on the surface in the primary BAT / PEO mixed monolayer, X_{BAT}, and not the concentration of biotin of the thiol solution used to prepare these surfaces. The units on the abscissa in Fig. 6.1a has also been changed in Fig. 6.1b, from a simple SPR shift to units of absolute SA coverage (ng / cm^2). This conversion was made as described in [3].

As can be seen in Fig. 6.1b, and like the adsorption results shown in the previous chapter, the amount of SA immobilized on the surface increases from 0 at 0% surface biotin (or 100% PEO, where almost no adsorption of SA is observed), to a maximum of ~230 ng of SA per cm^2 (or ~$2.6x10^{12}$ molecules / cm^2) at 29% surface biotin. As X_{BAT} increases further, the amount of SA bound drops. These trends match those previously observed by others using somewhat different molecules to attach the poly(ethylene oxide) and biotin functionalities to the surface [4,6-8].

The mutants exhibit this same trend. Figure 6.2 shows the saturation SA coverage versus surface biotin composition for W120A (Fig 2a), N23E (Fig 2b) and S27A (Fig 2c). All three of these mutant show less SA adsorption, as compared to the WT, at the lowest surface biotin concentrations presented (above 0%). But, as mentioned above, as X_{BAT} increases to 29% (the composition of maximal WT SA binding shown in Fig 1b.) these differences diminish. As X_{BAT} continues to increase above 30 %, the adsorption of these mutants also start to diminish. The difference in SA coverage compared to the WT are most pronounced with the W120A mutant. The difference

between N23E, S27A and WT at the lowest surface biotin concentrations are nearly negligible, but become more visible at the higher concentration (> 65% surface biotin).

SA ADSORPTION KINETICS

The kinetics of the adsorption of these SA variants to the BAT / PEO surfaces show a variation with biotin surface composition that changes in the same regions as does the saturation coverage. Fig. 6.3 shows the "initial rate of adsorption" (defined here as the rate averaged over the first ~50% of saturation) versus surface biotin concentration. Since the initial rate of adsorption followed similar trends for all four variants, the rates presented here are averaged over all four variants. (The small variations between variants will be discussed below.) At the lowest biotin coverages, adsorption is the slowest, most likely due to the low availability of biotin for binding. Thus, some collisions of SA with the surface are unproductive because they occur in an area where no biotins can be found. As the surface biotin composition increases, the adsorption rate rapidly increases, until ~3 % biotin. It remains constant until the surface biotin composition exceeds ~65%, where the adsorption rate again increases.

Figure 6.4 shows how the average initial adsorption rate depends upon the mutation. For the solid points, we have averaged the initial rate observed for each variant over all biotin concentrations where all four mutants had been measured. The rate is nearly independent of the variant of SA. A better comparison is perhaps made at the lowest biotin coverage (X_{BAT} = 0.34%), where the rates and coverage ranges probed are the lowest, and therefore the diffusion limitations are expected to be the least severe (see below). The initial rates at this biotin concentration are plotted as the open circles in Fig. 6.4. As can be seen, W120A has the slowest rate, WT and N23E are the fastest, and S27A is intermediate. This follows the trend of on-rates reported in homogeneous solution [1].

While Figs. 5.3 and 5.4 are for adsorption from a solution containing 0.05 mg / ml of SA, the kinetics of adsorption of WT SA on surfaces containing ~40% biotin was monitored for several SA concentrations in the buffer. The observed half-lives for adsorption are plotted versus concentration as the solid points in Fig. 6.5. The open circles shows the adsorption half-life that would be observed if the rate were completely diffusion limited. According to Eq. 6 in Chapter 4, this is given by: $t_{1/2,diff} = (1/2) [\theta_{max}^2 / (C^2 D)]$, where C is the concentration and D is the diffusion constant for SA (reported to be 7.4×10^{-7} cm^2 / s (as determined by light scattering measurements) [6]). When compared to the observed half life, this provides a measure of the significance of depletion in that adsorption curve. As can be seen, the measured half life is quite comparable to the diffusion-limited value, especially at low concentrations, where it decreases as $1 / C^2$, as expected for diffusion-limited kinetics and severe depletion. The adsorption rates of SA on similar surfaces have also been measured at various concentrations by Spinke et al. [6] and found to be mainly diffusion-limited kinetics. At high concentrations, on the other hand, the half life in Fig. 6.5 is substantially longer than $t_{1/2,diff}$, and it decreases only linearly with concentration, as expected for reaction-limited adsorption. For example, the half life is reduced by almost exactly 50% as the concentration doubles between the two highest-concentration points, as expected for reaction-limited kinetics.

The concentration used for adsorbing SA in all the other figures in this chapter is 0.05 mg / ml, which, as can be seen in Fig. 6.5, is the lowest concentration studied in this linear regime. This concentration is just high enough that the rate is no longer purely diffusion limited. This was also verified by simulations discussed below. Proof that this is not a strictly diffusion-limited regime at this concentration also comes from the fact that the rate depends upon the surface biotin composition (Fig. 6.3) and the mutant (Fig. 6.4).

Sticking probabilities for numerous SA uptake curves were calculated by a method described in detail in Chapter 4. Briefly, the observed rate of adsorption versus

time from the SPR measurement was applied as a boundary condition to a numerical solution to Fick's Law, which provides the SA concentrations versus distance from the surface as adsorption proceeds. This resulting concentration nearest the surface dictates the actual flux to the surface (collision frequency with the surface), allowing calculation of the sticking probability versus time or coverage. The initial value, S_0, was determined by extrapolating the sticking probability to zero coverage. The value of S_0, averaged over the variants as with the initial adsorption rates of Fig. 6.3, is also shown in Fig. 6.3. It varies with surface biotin composition in a similar way as those initial rates.

A typical result showing the complex dependence of the sticking probability of WT SA upon coverage is shown in Fig. 6.6 for a surface containing ~30% BAT. The apparent slight increase seen at the lowest coverages (shortest times) probably is an artifact of the time response of our flow system (~1 sec). The sticking probability decreases markedly between 4 to 8×10^{11} molecules / cm^2. Thereafter, it decreases more slowly with coverage until it reaches a very low value at ~2.1×10^{12} molecules / cm^2. Adsorption proceeds with a very low probability above this, until saturation is eventually reached at ~2.5×10^{12} molecules / cm^2. Simple first-order Langmuir-type adsorption kinetics predict that $S = S_0 (1-\theta / \theta_{max})$. The sticking probability decreases with coverage less rapidly at low coverages and more rapidly at high coverages than predicted by such a model. This suggests that there are different types of sites that can be filled on the surface, with some showing a much higher adsorption probability than the others, and thus saturating first. This is no doubt associated with the intrinsic heterogeneity of the mixed BAT / PEO monolayer and the induced heterogeneity associated with repulsive lateral interactions between the incoming SA molecule and SA molecules already at the surface, which would inhibit adsorption. This picture is consistent with the desorption kinetics, which show heterogeneity in desorption rate constants (see below).

SA COMPETITIVE DESORPTION KINETICS

The amount of adsorbed SA versus time upon exposure to a buffer containing excess biotin (1.0 mM) is shown in Fig. 6.7 for a variety of BAT / PEO surface compositions, each presaturated with the W120A mutant. The competitive desorption kinetics clearly changes with the surface BAT concentration. The off-rates (slopes of these curves) at 16% and 29% surface BAT are slower than at all other compositions presented in this figure. Also as the surface BAT composition increases from 65% to 100% there appears to be an increase in the amount of W120A which remains on the surface indefinitely. This was made more quantitatively obvious by fitting the curves to a double exponential as described below. The constant (non-exponential) fraction determined in the fits was largest at 100% biotin, not only for W120A but for all the variants. This is possibly due to some non-specific interactions of SA with these surfaces. At such high BAT concentrations, angular-resolved XPS measurements show that more of the BATs tended to lie over on the surface [2], thus exposing their sides (rather than their biotin headgroup) to the incoming SA. This might facilitate non-specific binding of SA to the surface.

The average W120A desorption half-lives determined from competitive desorption curves like those in Fig. 6.7 are plotted in Fig. 6.8 versus surface BAT concentration. An average pseudo-first-order rate constant, k, was also determined from each half life by: $k = (\ln 2) / t_{1/2}$. Its value is also plotted in Fig. 6.8 along the right-hand axis. Although this desorption behavior has been shown to be described better with a double exponential [3,4], which will be discussed further below, this single first-order rate constant gives a single parameter which is useful in showing the trends associated with changing the surface BAT concentration or the SA mutant. As can be seen, the half-life is shortest at the lowest BAT concentration, increases by a factor of ~30 up to ~30% BAT, then decreases by a factor of ~15 by ~60% BAT, and finally increase slowly up to 100% BAT.

The half lives and off-rate constants for the other three SA variants are also presented in Fig. 6.8 and show similar qualitative trends with BAT concentration. For a few of the desorption curves, particularly for the WT SA which desorbs extremely slowly, the desorption curves could not be followed for a long enough time to actually measure the half-life. In these cases, the half life and rate constant for desorption were estimated using only the fractional coverage observed at the longest measured time, assuming that the coverage obeys simple first-order decay kinetics: $\theta / \theta_{max} = \exp(-kt)$, where θ / θ_{max} is the coverage at this longest time, t, and k is the rate constant. It should be noted that the baseline stability of our SPR device limits our ability to accurately measure half-lives $>10^5$ s, so the accuracy of half-life determinations above this is very poor.

The desorption rate constants are the slowest for WT, followed by S27A and N23E, which are quite comparable. Desorption of W120A is by far the fastest. This trend is also reported in homogeneous solution [5]. At 29% surface BAT and above, the differences in the desorption rate constants of S27A and N23E compared to WT are markedly reduced.

As can be seen in Fig. 6.8, the fastest off-rates are seen at 0.4% BAT, the slowest are seen at 29% BAT, and the rates decrease dramatically again by 84% BAT. Figures 6.9 - 6.11 compare the short-time regions of the off-rate curves for each of the SA variants at these three surface BAT concentrations, respectively. As we will discuss further below, the BAT concentration is so low in Fig. 6.9 (0.34%) that one expects the surface SA population to be dominated by the species that are bound to the surface via a single BAT, as shown schematically in the inset. Indeed, these curves are all fairly well fitted by a single exponential decay (plus a constant residue). In contrast, for Fig. 6.10 at 29 % BAT, the biotin concentration is high enough that one expects that most SA species will bind to the surface via two BATs, as shown schematically in the inset. The off rates are the slowest at this BAT concentration, which supports this. Furthermore, each of the

off-rate curves in Fig. 6.10 are well fitted by a single exponential, albeit with much, much smaller rate constants than in Fig. 6.9.

Finally, at 84% BAT (Fig. 6.11), the off-rates are again much faster. This would imply that the binding of SA has reverted to a single BAT linkage to the surface. The reason this might occur is suggested in the cartoon inset of Fig. 6.11. Here, the crowding of BAT on the surface may be too high, so that in many areas of the surface, BATs are so closely surrounded by other BATs that none are accessible to protrude deep enough into the pocket-like binding site of a SA. This would also explain why the saturation coverage decreases at high BAT concentrations (Figs. 5.1 - 5.2). Still, some isolated or protruding BATs exist on the surface that can bind to the sites in SA, although these are rare enough that few SAs bond to the surface via two BATs. This would explain the high off-rates observed, which for the WT become even faster than at 0.34% BAT (Fig. 6.9). However, the curves of Fig. 6.11 are poorly fitted with a single exponential, and at least a double exponential is needed to describe these data. This suggests that there is more complex model required than the competitive desorption of a single type of species from the surface. Similarly, the off rates decrease with BAT concentration between 65% and 100%, which is hard to explain, and the amount of irreversibly adsorbed SA increases to a maximum at 100% BAT. These observations suggest that more complex mechanisms for binding SA to the surface exist at these high BAT concentrations than those solely associated with biotin-SA recognition. Non-specific binding of SA to the sides of the BAT molecules is facilitated at high BAT coverages (since they lie over more on the surface according to XPS). An increasing extent of non-specific binding between 60 and 100% BAT would also explain the weak decrease in SA off-rate (Fig. 6.8) and the increase in SA adsorption rate (Fig. 6.3). Steric repulsions at the binding sites due to nearby BATs may also complicate the binding strengths of SAs to the surface. Also, the binding strengths in Fig. 6.9 and 6.11 may be different since the SAs are surrounded mainly by PEO in Fig. 6.9, but mainly by BATs in Fig. 6.11. These will interact weakly, but differently, with the surfaces of the protein.

DISCUSSION

In general, the desorption curves represented by Figs. 5.8 - 5.11 were not fit well by single exponential decays, except at a few very specific conditions (e.g., at 0.34% biotin, and at 29 % biotin in some cases). However, the curves were all fit well with a linear combination of two exponential decays, each of a different rate constant and scaling factor, plus a constant (time-independent) term, as also pointed out in the previous chapter. We will refer to this fit here as a "double exponential fit". As discussed in the previous chapter, the two exponentials can be attributed to the presence of two species of SA on the surface: one with a faster off-rate due to SA bound to the surface via a single BAT, and a second with a much slower off-rate, due to SA bound to two surface biotins.

Two possible "types" of immobilized SA on the surface were shown schematically in the illustrations in Figs. 5.9 - 5.10. The species bound to two surface biotins (Fig. 6.10) will be represented here as "BSB", and the species bound to the surface by a single biotin (Fig. 6.9) will be represented as "-SB". Since there is a large excess of biotin (of the same concentration) in the aqueous solutions in all of the experiments, the competitive desorption of SA from the single-bound structure, induced by this liquid-phase biotin, can be described simply as a pseudo-first-order decomposition reaction:

$$-SB \xrightarrow{k} B- + S(aq), \qquad (1)$$

where B- is the free surface BAT, S(aq) is the aqueous SA product (an adduct of SA with aqueous biotin), and k is the rate constant associated with the competitive dissociation of one surface biotin - SA linkage. Thus, the time dependence of the coverage of the single-bound SA can be described with a single first-order exponential decay:

$$\theta_{-SB} = \theta_{-SB}^{0} \exp(-kt) \qquad (2)$$

where θ_{-SB}^{0} is the initial coverage of the single-bound species. (It should be noted that, in the simplest case, the value of k would be proportional to the concentration of biotin in the aqueous solution.)

Similarly, the competitive desorption of SA from the doubly-bound structure can be described as occurring in two pseudo-first-order decomposition steps, each with this same rate constant k, where the first step is additionally assumed to be reversible:

$$BSB \underset{k_b}{\overset{k}{\rightleftarrows}} B---SB \xrightarrow{k} 2B- + S(aq). \qquad (3)$$

Note that it is unnecessary to include the reverse of the second step, because once -SB dissociates from the biotin linkage it is released into the solution and diffuses away from the surface faster than it can readsorb. Note also that the species label "B---SB" refers here to a SA bound to the surface via a single BAT, but with another surface BAT available (but unbound) very close to the other (now occupied) biotin binding site on the underside of that same SA. This is to be distinguished from the species "-SB" above which has no nearby surface biotin so readily available for competitive binding to its other binding site. The assumption that both steps have the same k is expected to be valid since multiple biotins bind to a single SA without any apparent cooperativity [9]. We assume here that the B---SB species created in the first step can competitively rebind to the same surface biotin it lost and thus reform the BSB with a pseudo-first-order rate constant, k_b.

We now derive an expression for the time dependence of the quantity we would measure by SPR, assuming we start only with species BSB, at an initial average θ_{BSB}^{0}. Since both BSB and B---SB are detected by SPR, the sum of their surface concentrations, $\theta_{BSB} + \theta_{B---SB}$, divided by θ_{BSB}^{0} give the fraction bound observed by SPR:

$$f = (\theta_{BSB} + \theta_{B\text{---}SB}) / \theta_{BSB}^{0} \quad (4)$$

We will assume that the intermediate in this 2-step mechanism, B---SB, rapidly reaches a steady state concentration. We verify this assumption below by showing that k_b is much faster than k. In this case, the rate of change of its concentration is zero:

$$d\theta_{B\text{---}SB} / dt = k\,\theta_{BSB} - k_b\,\theta_{B\text{---}SB} - k\,\theta_{B\text{---}SB} = 0. \quad (5)$$

Solving for $\theta_{B\text{---}SB}$ gives

$$\theta_{B\text{---}SB} = k\,\theta_{BSB} / (k + k_b) = K\,\theta_{BSB}, \quad (6)$$

where the combined constant K is defined as:

$$K = k / (k + k_b). \quad (7)$$

The rate of change in concentration of BSB is:

$$d\,\theta_{BSB} / dt = k_b\,\theta_{B\text{---}SB} - k\,\theta_{BSB} = (k_b K - k)\,\theta_{BSB}$$

$$= -[k^2 / (k + k_b)]\,\theta_{BSB} = -(k K)\,\theta_{BSB}, \quad (8)$$

which integrates to give:

$$\theta_{BSB} = \theta_{BSB}^{0} \exp(-k K t) \quad (9)$$

Adding θ_{BSB} to both sides of Eq. (6) gives:

$$\theta_{BSB} + \theta_{B\text{---}SB} = (1 + K)\,\theta_{BSB}, \quad (10)$$

so that

$$f = (\theta_{BSB} + \theta_{B\text{---}SB}) / \theta_{BSB}^{0}$$

$$= (1 + K) \exp(-k K t) \quad (11)$$

or

$$\theta_{BSB} = \theta_{BSB}^{0}(1 + K)\exp(-k\,K\,t) \quad (12)$$

We will show below that $K << 1$ (since $k_b >> k$), so that the factor $(1+K)$ is almost unity for the present cases.

In order to account for presence of both -SB and BSB on the surface initially, and since both these species (and B---SB) are detected by SPR, both Eqs. 2 and Eq. 12 are summed to give:

$$\theta(t) = \theta_{-SB}^{0}\exp(-k\,t) + \theta_{BSB}^{0}(1 + K)\exp(-k\,K\,t) + \theta_{irrev} \quad (13)$$

where the term θ_{irrev} has also been added to account for any SA which is irreversibly bonded to the surface (i.e. does not desorb significantly on the experiment's time scale). Thus, the decay can be expressed as a double exponential equation with two apparent rate constants, k and k', where k' is simply

$$k' = k\,K = k^2/(k_b + k) \quad (14)$$

This double exponential behavior agrees well with our observations above and those in the previous chapter.

Perez-Luna et al. [4] also observed a double exponential behavior, but proposed a mechanism to explain it which differs substantially from that proposed here in that: (1) they assumed that a *different* rate constant was needed to describe the loss of the second biotin bond to the surface for a SA species initially bound to two surface biotins, and (2) that a *direct*, single pseudo-first-order step also existed which contributes significantly to the removal of this doubly-bound species from the surface. We believe that this complexity is not necessary, and that the data can be better modeled by simply including the reverse of the first step in the mechanism. The observed non-cooperativity in biotin binding to SA [9] also supports a model like ours where the same rate constant

is used to describe the loss of both biotin bonds to the surface for those species initially bound via two surface biotins, and the same constant also applies for the loss of the biotin bond to the surface for the species initially bound to the surface via a single BAT.

Unfortunately, for a given mutant, the double exponential curve fits at all the different surface biotin concentrations did not give the simple result that they could all be characterized with just the same two decay rate constants. This we attribute this to the fact that there are also more subtle effects at work here, for example: (1) lateral interactions between coadsorbed SAs which stabilize or destabilize each other enough to show up as changes in the off-rate constants by large factors, and (2) non-specific interactions of the SA with the PEG and BAT headgroups which also significantly affect the rates. These effects vary across the surface, and led to double exponential fits even when only a single species (e.g., singly- or doubly-bound SA) dominates on the surface (according to our analysis below). Nevertheless, the curves at all compositions below 30% BAT for a single SA variant were reasonably well fitted by double exponentials where the rate constants for the two off-rates were forced to single constants, independent of the biotin surface composition. The fits, however, were poor enough (factors of ~3 errors in some regions) that the preselection of these two rate constants was rather arbitrary.

Instead, it seems more accurate to determine the two characteristic rate constants by finding those conditions where a single species appears to dominate on the surface and then use the data at those conditions alone to determine the rate constant for that species from a single exponential fit. Figures 6.9 and 6.10 show characteristic off-rate data for those conditions which most clearly represent the singly- and doubly- bound SA species, respectively. In Fig. 6.9 (0.34% BAT), the surface is dominated by the -SB species, which dissociates with rate constant, k. In Fig. 6.10 (at 29% BAT) the surface is initially dominated by the BSB species, which dissociates with a rate constant k'. The average values found for k and k' for each different SA variant by analysis of off-rate curves at these two conditions are listed in Table 6.1. In addition, the values for k_b found

from these values by application of Eq. (14) is also presented in Table 6.1. For comparison, Table 6.1 also lists the pseudo-first-order biotin-SA off-rate constants determined in homogeneous aqueous solution at 25° C by Klumb and Stayton in previous studies of competitive decomposition in 0.05 mM biotin solution [1]. As noted above, these pseudo-first-order rate constants should be proportional to the biotin concentration in the simplest model (i.e., a true second-order reaction). Thus, these rate constants in homogeneous solution possibly should be multiplied by 20 to compare to our surface measurements which were done with 1.0 mM biotin. The level of agreement between k and the value measured in solution is quite good, and comparable in either way, since the pseudo-first-order rates on the surface are ~4-fold faster than in solution, as shown. The difference between the off-rates estimated in solution and the measured off-rate for the surface bound W120A appears to be large. However, the solution phase value was not actually measured, but instead very indirectly estimated from some other types of measurements, so this difference may not be as significant as it appears. We were able to accurately measure fast off-rates (~ few seconds), which allowed measurements of the off-rate for W120A directly. This was not possible in solution-phase measurements.

The large values of k_b compared to k in Table 6.1 confirm the steady-state approximation made in the kinetic derivation above. The values of k_b correlate poorly with the value of k and the solution phase off-rate constant. A strong correlation might be expected, since this reaction step also involves removing one biotin from the SA and replacing it with another biotin, but in this case the latter is also attached to the gold. Note, however, that k_b is in some sense also similar to an adsorption rate constant, except that a biotin must (presumably) be released from that SA site as it binds to the surface biotin. Since the initial adsorption rates of the different variants of SA all were observed to be very similar, this k_b might be expected to be similar in all the variants, if it were not for this necessity to simultaneously remove an existing biotin from the site on the SA. This should make this step much more difficult and cause k_b to reflect to some extent the differences in binding stability of the different mutants seen in the k values of Table 6.1.

This is, however, not obvious. The relative errors on k_b are at least 3-fold larger than those on k or k', since k_b is roughly given by k^2 / k'. Furthermore, k' is so slow for WT that its value can only be limited to $<10^{-5}$ s^{-1}, which allows us to only limit k_b for WT to $>6x10^{-5}$ s^{-1}. These large errors may prevent us from making a more definitive statement about the correlation between k and k_b, than to say that it is not strong. Nevertheless, this suggests that the step: $BSB \underset{k_b}{\overset{k}{\rightleftharpoons}} B---SB$ may not be quite as we have described it above, in that the site on the SA created by the loss of one BAT may not be completely bound with an aqueous biotin competitor. This competitor biotin may only be partially bonded, so that the reverse step (to reattach that SA site to the surface BAT) may be less directly connected to the energetics of the biotin-SA bonding than expected. The fact that k_b is much larger than k may be partially related to this effect. It is certainly occurring because the "effective concentration" of BAT in this B---SB species, which drives competitive rebinding through the value of k_b, is much larger (due to their built-in proximity) than the aqueous biotin concentration of 1 mM used to drive the forward rate constant, through the value of k. It may be that this BAT is so close to the SA site that it has already partially scaled the potential energy barrier to the transition state experienced by an aqueous biotin.

An interesting effect can be seen in Fig. 6.8. At 29% BAT, the differences in off rates between the different mutants seen at 0.34% BAT are greatly amplified for the W120A compared to the S27A or N23A. A similar effect might have been present but masked for the WT species, since off-rate half-lives in excess of $\sim 10^5$ s exceed our dynamic range. This amplification can be easily understood through Eq. (14): $k' = k^2 / (k_b + k) \cong k^2 / k_b$. Since k_b varies weakly with mutation compared to k, the observed off-rate constant varies almost as the square of k. This offers further support to the mechanism proposed here.

CONCLUSIONS

The adsorption and desorption behavior of WT and SA mutants on these mixed BAT / PEO layers confirm that surface layers can be prepared which allow rapid, quantitative testing of the effects of protein mutations on their binding strengths to ligands. Adsorption is nearly diffusion rate limited, but subtle differences in the adsorption rate can be seen between the different variants and as a function of X_{BAT} and SA coverage. The desorption kinetics strongly vary between variants and as a function of X_{BAT}. The dependencies on X_{BAT} suggests strong heterogeneity in the surface layer. At low coverages (0.34% surface biotin), the surface is composed primarily of singly-bound SA, which desorbs with rate constants for the different SA variants which are similar to those from solution measurements. As the concentration of BAT increases to ~29%, the desorption rates slow and the amount bound increases. These observations, in addition to the change in off-rates, point to a surface dominated by doubly-bound species of SA. A mechanism for desorption which models these data well is proposed, and the two key rate constants for elementary steps in this mechanism are determined for each SA variant. They are tabulated in Table 6.1 and correspond to the pseudo-first-order rate constants for (1) the competitive cleavage of a bond between a SA site and a surface-bound biotin (i.e., replacement with solution-phase biotin) and (2) the reforming of this bond in the special case that it was cleaved from a doubly-bonded species whose other site is still bound to a surface-immobilized biotin. For X_{BAT} above 34%, the SA desorption rates increase and the amount adsorbed decreases. This is attributed to close-packed clusters of BAT in the monolayer which, due to greater steric hindrance, are not able to bind into the site of a SA. These prevent SA from forming bonds to two BATs, and it instead appears to link to the surface via a single BAT, as if the BAT concentration were very low. By 100% BAT, more non-specific interactions with the surface appear, which lead to irreversibly adsorbed SA, and correlates with XPS measurements. The latter show less alignment of the surface BAT, so that many of the biotin headgroups are now buried in the adlayer, rather than protruding outward from the surface as observed at lower X_{BAT}. These results

show that X_{BAT} should be about 25% if one is interested in creating the most stable and dense SA linker layers. On the other hand, X_{BAT} of < 1% should be used in hybrid monolayers designed to probe the effects of SA (or biotin) mutations on the intrinsic strength and kinetics of SA-biotin binding.

NOTES TO CHAPTER 6

[1] Klumb, L. A.; Chu, V.; Stayton, P. S. *Biochemistry* **1998**, *37*, 7657-7663.

[2] Nelson, K. E.; Jung, L. S.; Gamble, L.; Boeckl, M.; Naeemi, E.; Campbell, C. T.; Castner, D. G.; Stayton., P.S., *Langmuir (submitted).*

[3] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[4] Perez-Luna, V. H., O'Brien, M. J., Opperman, K. A., Hampton, P. D., Stayton, P. S., Klumb, L, Lopez, G. P. *J. Am. Chem. Soc* **1999**, *121*, 6469-6478.

[5] Chilkoti, A.; Tan, P. H.; Stayton, P. S. *Proc. Natl. Acad. Sci. USA.* **1995**, *92*, 1754-8.

[6] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

[7] Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

[8] Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

[9] Jones, M. L.; Kurzband, G. P. *Biochemistry* **1995**, *34*, 11750-11756.

FIGURES FOR CHAPTER 6

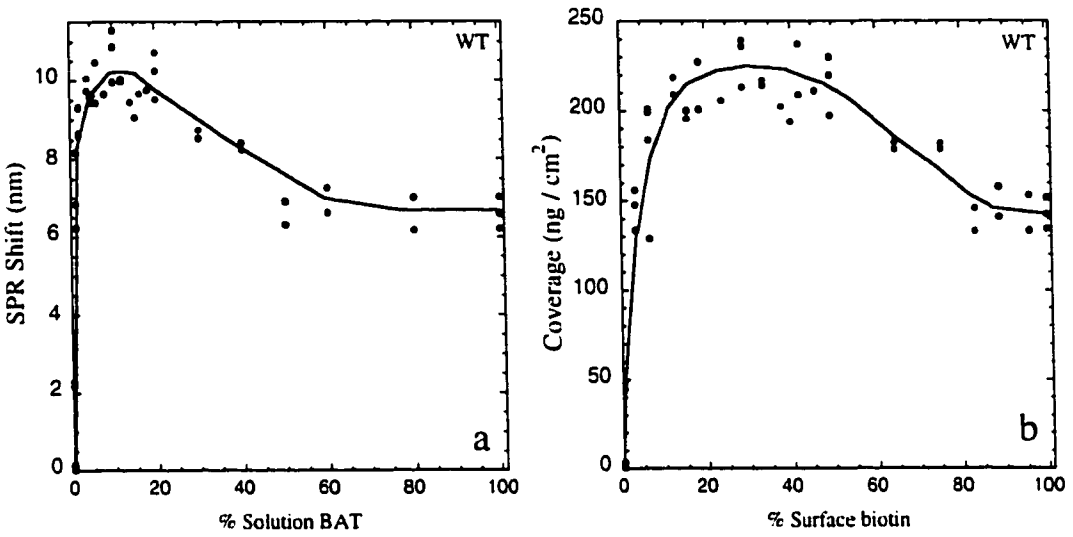


Figure 6.1 The adsorption of wild-type streptavidin from PBS on the mixed BAT / PEO monolayers. a) SPR wavelength shift observed upon exposure to 0.05 mg / ml WT SA in PBs buffer versus mole % of BAT in the original BAT + PEO alkylthiol solution (ethanol) used to prepare the monolayer. b) The same data now presented as coverage (ng / cm^2) versus % surface biotin.

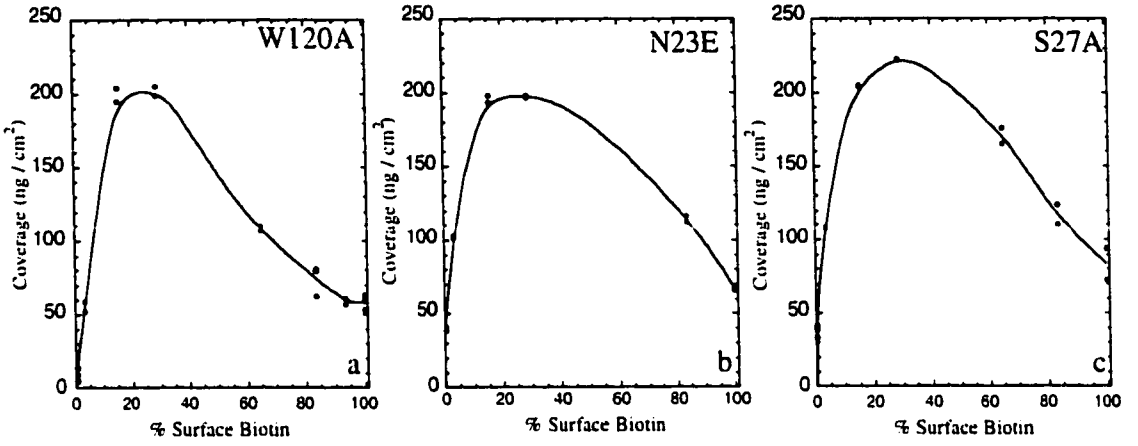


Figure 6.2 The adsorbed coverage versus surface biotin composition for a) W120A, b) N23E and c) S27A.

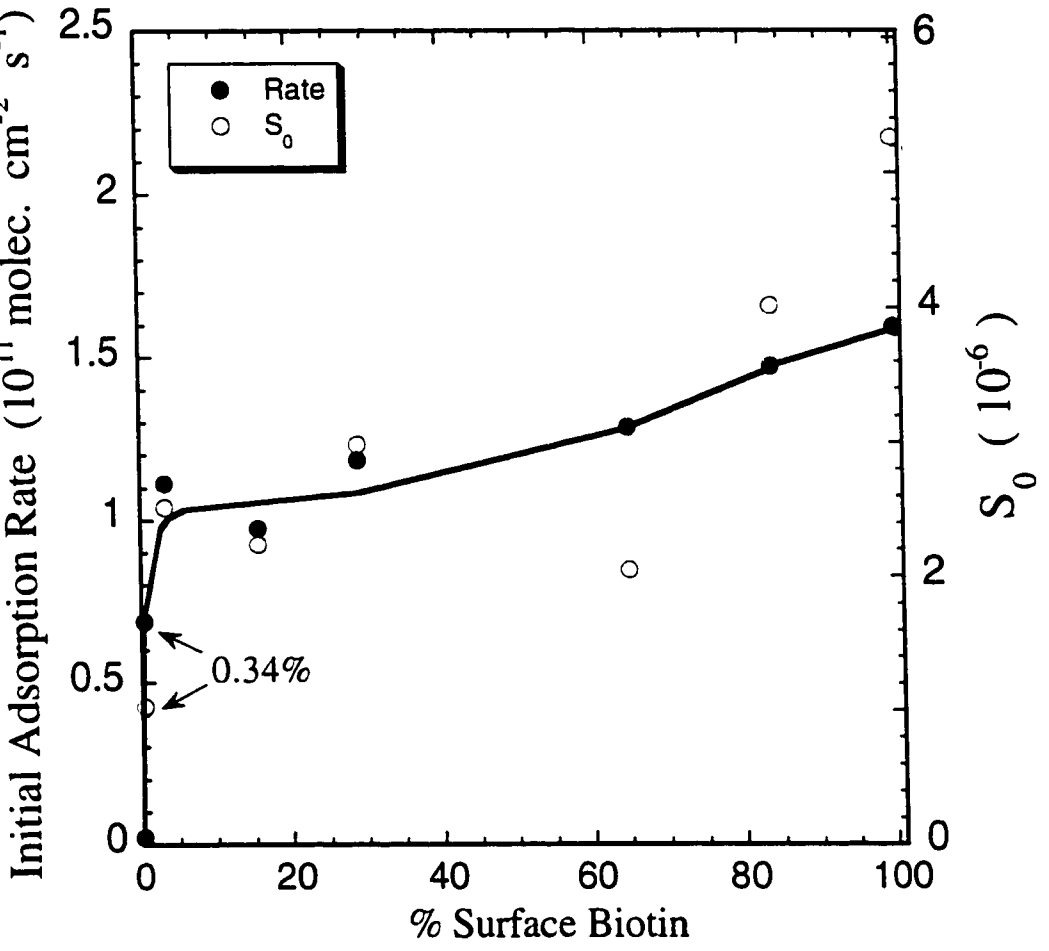


Figure 6.3 "Initial rate of adsorption" (the rate averaged over the first ~50% of saturation) averaged over all four SA variants versus surface biotin concentration (filled circles). The open circles are S_0 averaged over all four SA variants. Both the adsorption rate and the S_0 are the lowest at the lowest biotin coverages. As the surface biotin composition increases, the rate and S_0 increase to a stable value until the composition exceeds ~65% surface biotin where both start to increase again.

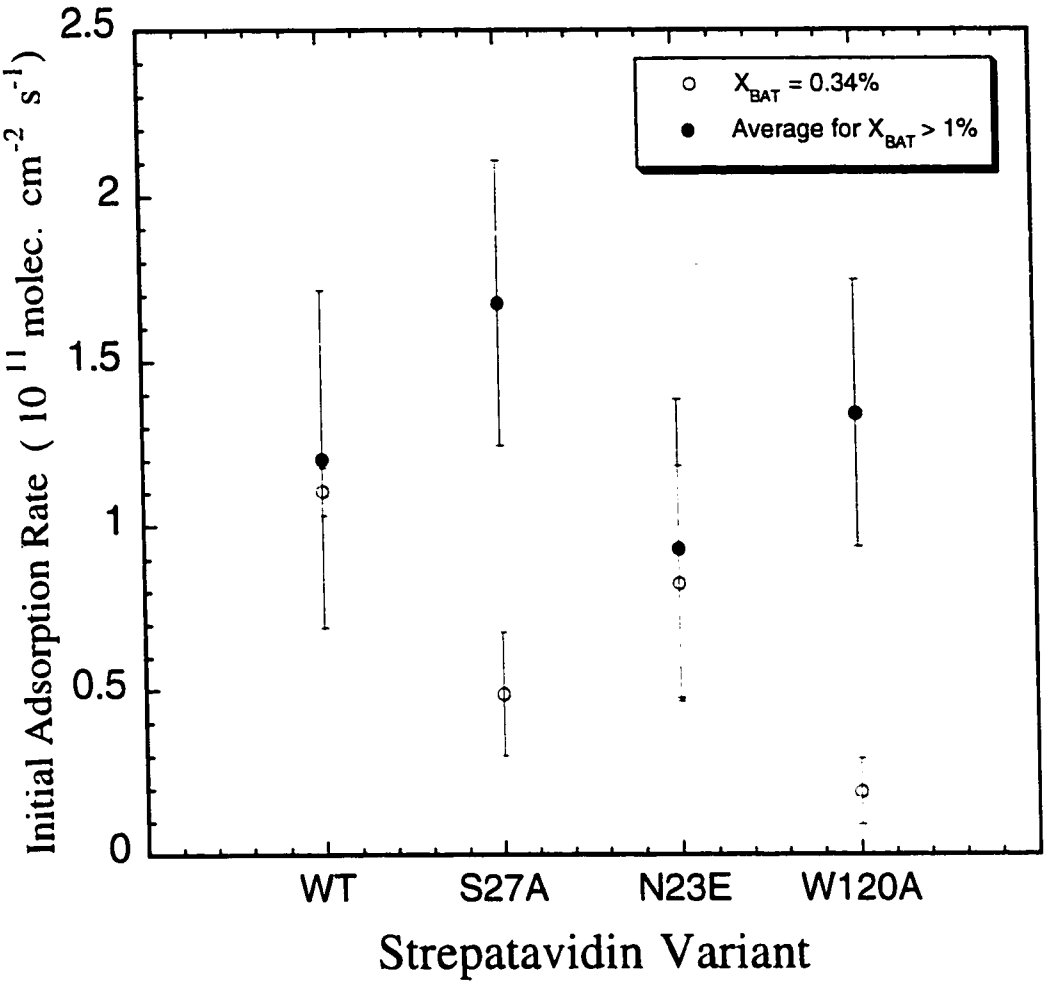


Figure 6.4 Average initial rate observed for each SA variant over all biotin concentrations where all four variants had been measured (filled circles). These rates appear nearly independent of SA variant. The open circles are the initial rates at the 0.34% surface biotin concentration which may provide a better comparision of the rates (see text).

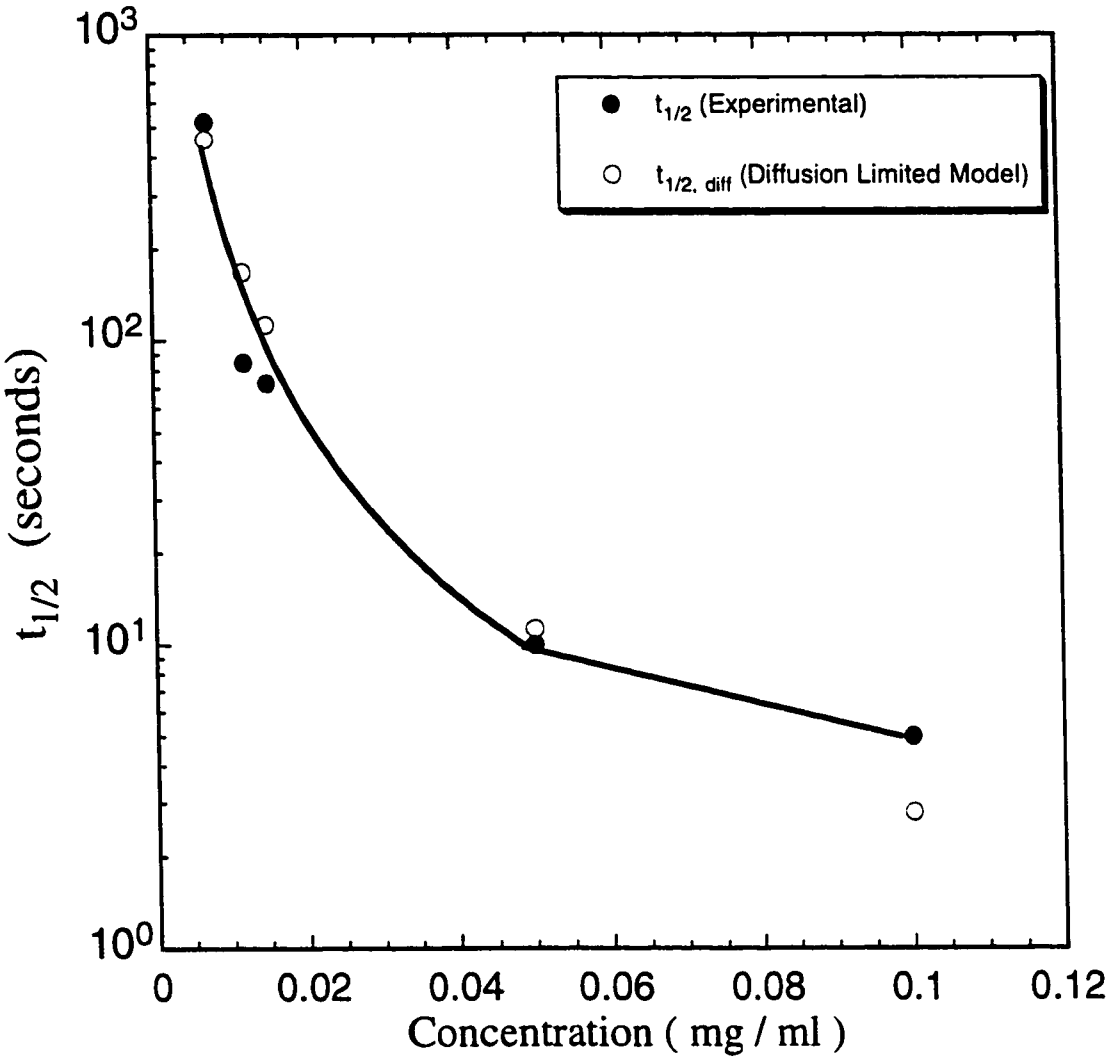


Figure 6.5 Adsorption half-life for WT SA at various SA solution concentrations (filled circles). The line represents the trend in the $t_{1/2}$ with concentration. The measured half-life at lower concentrations decreases as $1/ C^2$ as expected for diffusion - limited adosrption, while the higher concentrations decrease linearly as expected for reaction - limited adsorption. The open circles shows the adsorption half-life that would be observed if the rate were completely diffusion limited ($t_{1/2,diff}$).

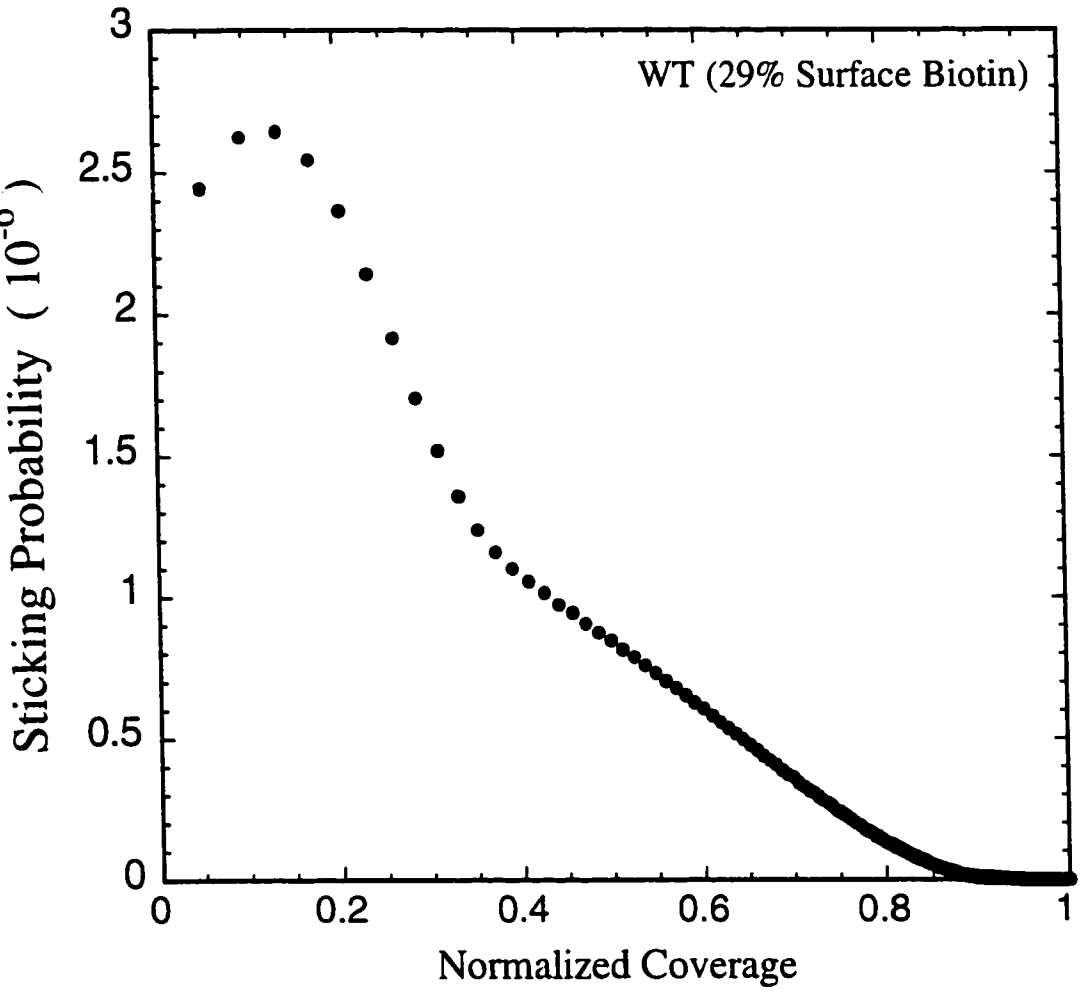


Figure 6.6 Sticking probability of WT SA versus coverage for a surface containing ~29% surface biotin. The apparent slight increase seen at the lowest coverages (shortest times) is probably is an artifact of the time response of our flow system (~1 sec). The sticking probability decreases markedly at low (< 40%) coverages and and decreases much more slowly after that until ~85% of the saturation coverage is reached. Above 85%, adsorption proceeds with a very low probability until saturation coverage is finally reached.

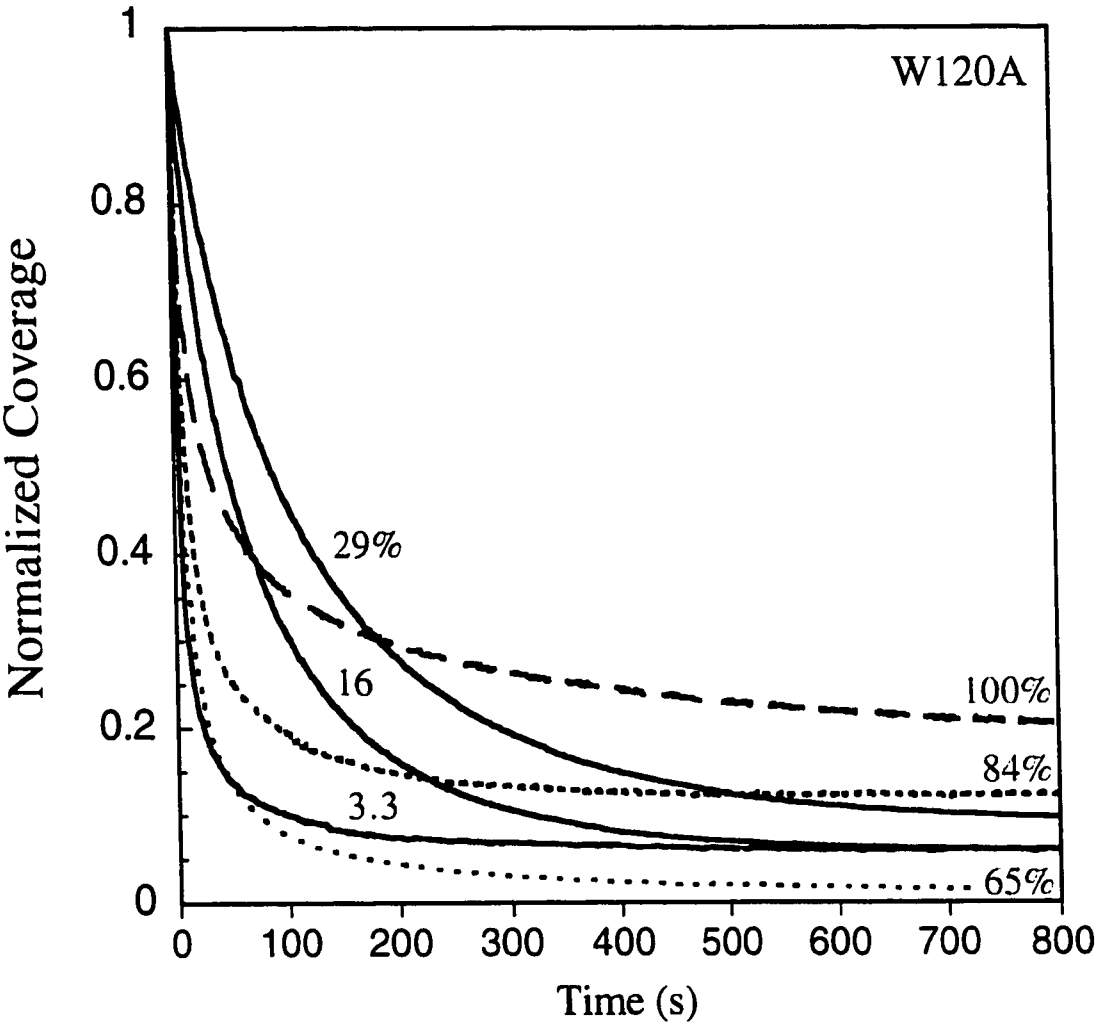


Figure 6.7 Competitive desorption of W120A from mixed BAT / PEO monolayers (3.3%, 16%, 29%, 65%, 85% and 100%) induced by injection biotin-containing buffer at time 0, measured by SPR. The amount of SA adsorbed (upon injection of the biotin-containing buffer) of each sample was normalized to 1. Each curve is labeled with the appropriate surface biotin composition. Concentrations < 65% are shown with solid lines, while concentrations ≥ 65% are shown with dashed gray curves.

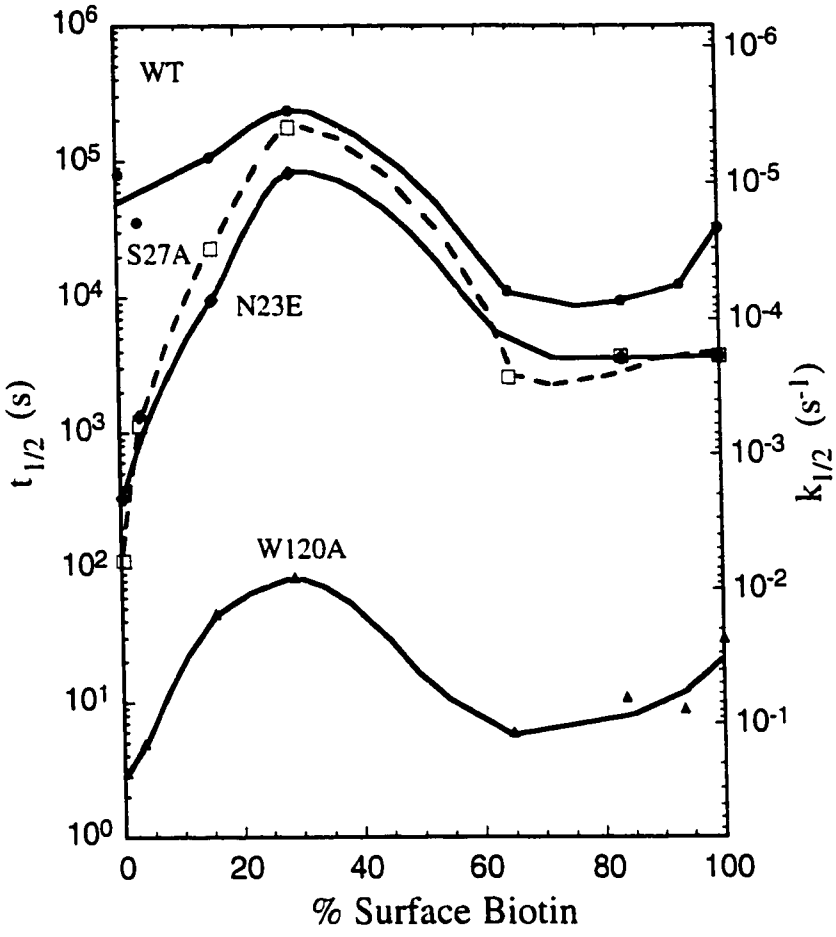


Figure 6.8 The half-lives and $k_{1/2}$ of all four SA variants versus % surface biotin. $k_{1/2}$ is the off-rate constant determined by the observed half-life, $t_{1/2}$, of desorption, where $k = \ln 2 / t_{1/2}$. The desorption rate constants for W120A are significantly faster than the other variants. WT is the slowest, followed by S27A and N23E, which are similar. Around 29% surface biotin, the differences between the S27A and N23E as compared to WT are markedly reduced.

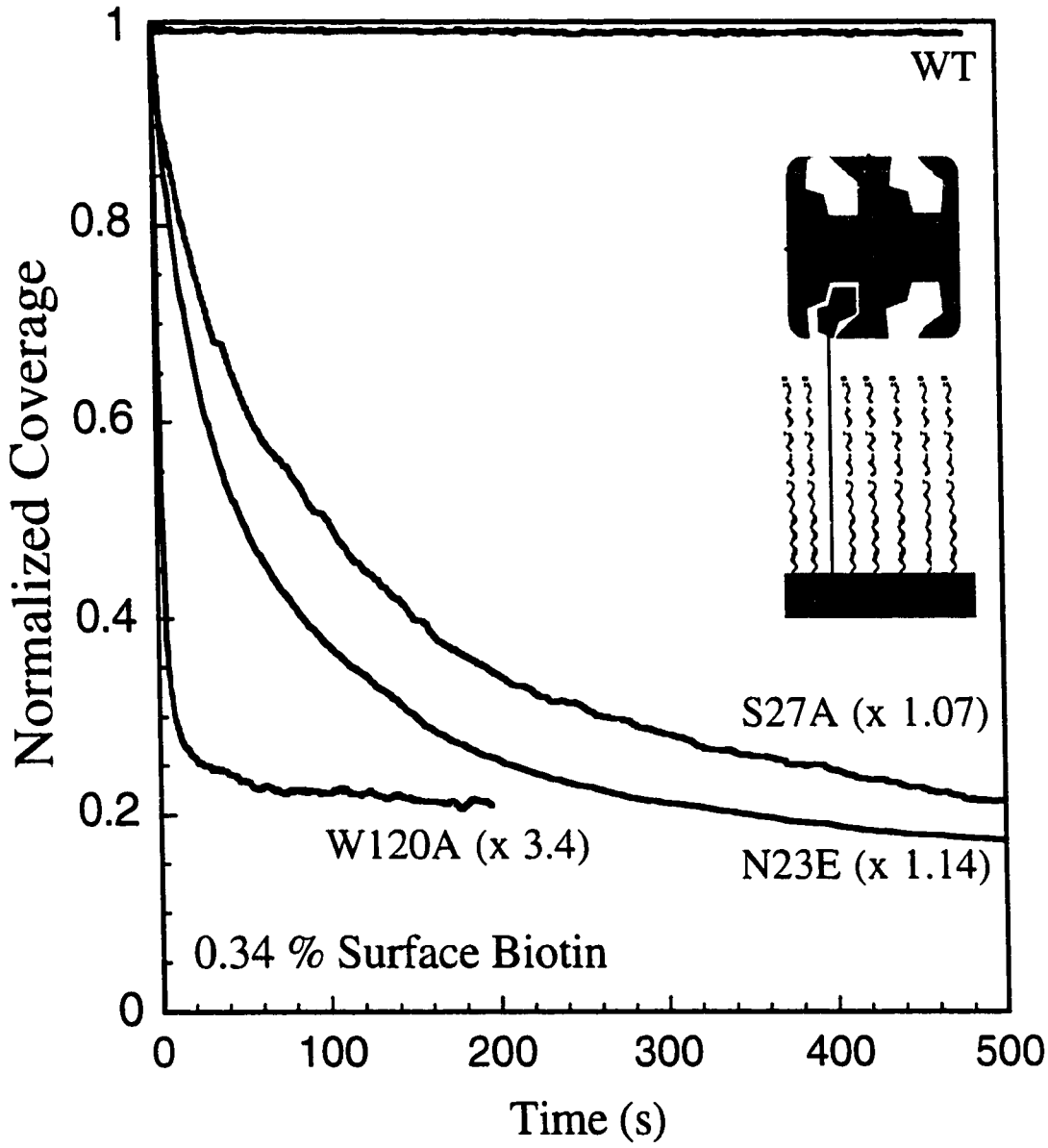


Figure 6.9 Normalized short-time regions of the desorption curves of the SA variants at 0.33% surface biotin. The curves were scaled by the amount shown in the (). The inset illustration depicts SA bound to the surface by a single biotin linkage due low BAT concentration. At this low surface biotin composition the surface is expected to be dominated by such species.

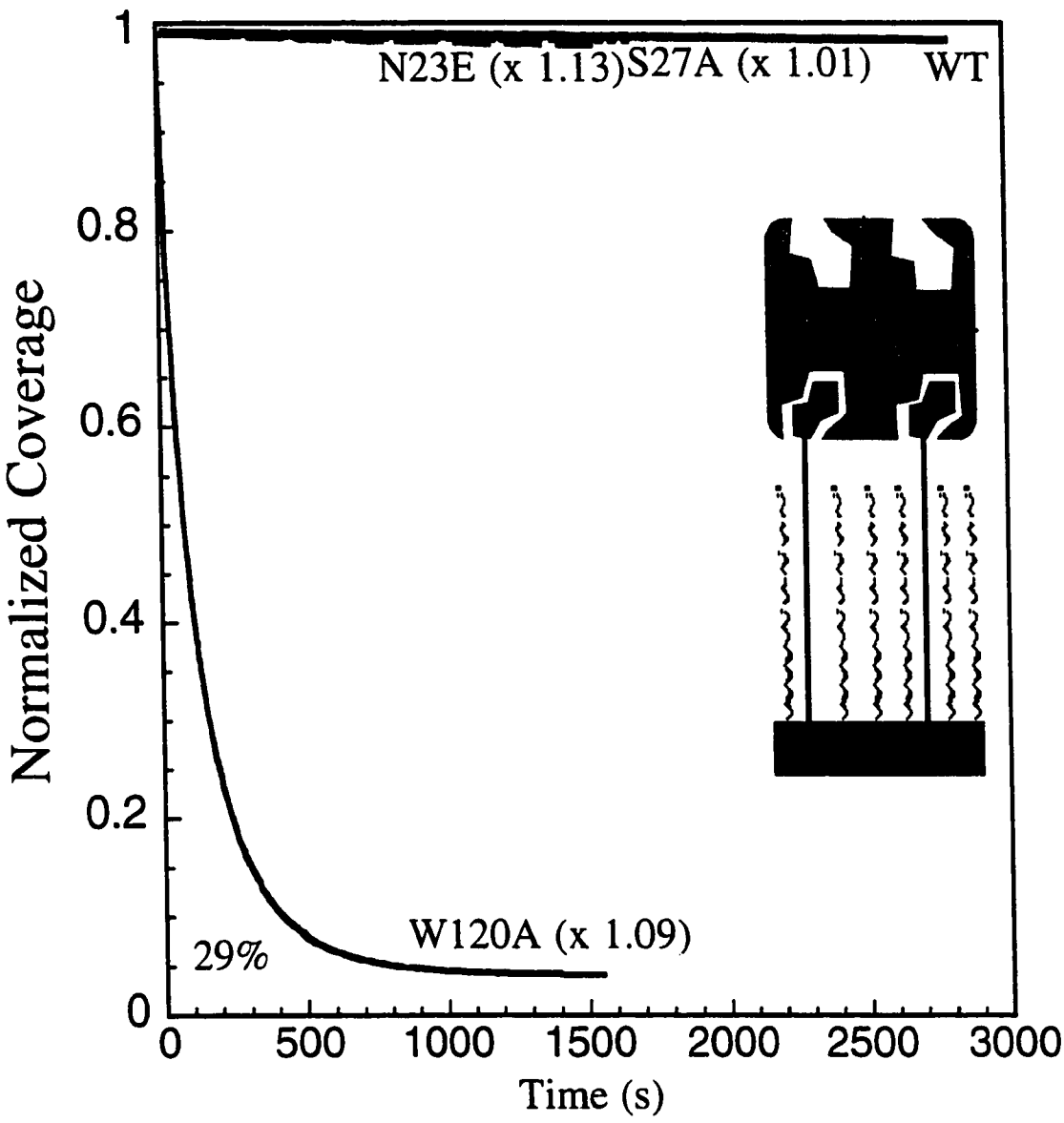


Figure 6.10 Normalized desorption curves at 29% surface biotin. The curves were scaled by the amount shown in the (). Since WT, S27A and N23E are similar, S27A is shown with the dotted curve and N23E is shown with the lower grey dashed curve. The inset illustration depicts SA bound to the surface by two biotin linkages. At this surface biotin composition the surface is expected to be dominated by such species.

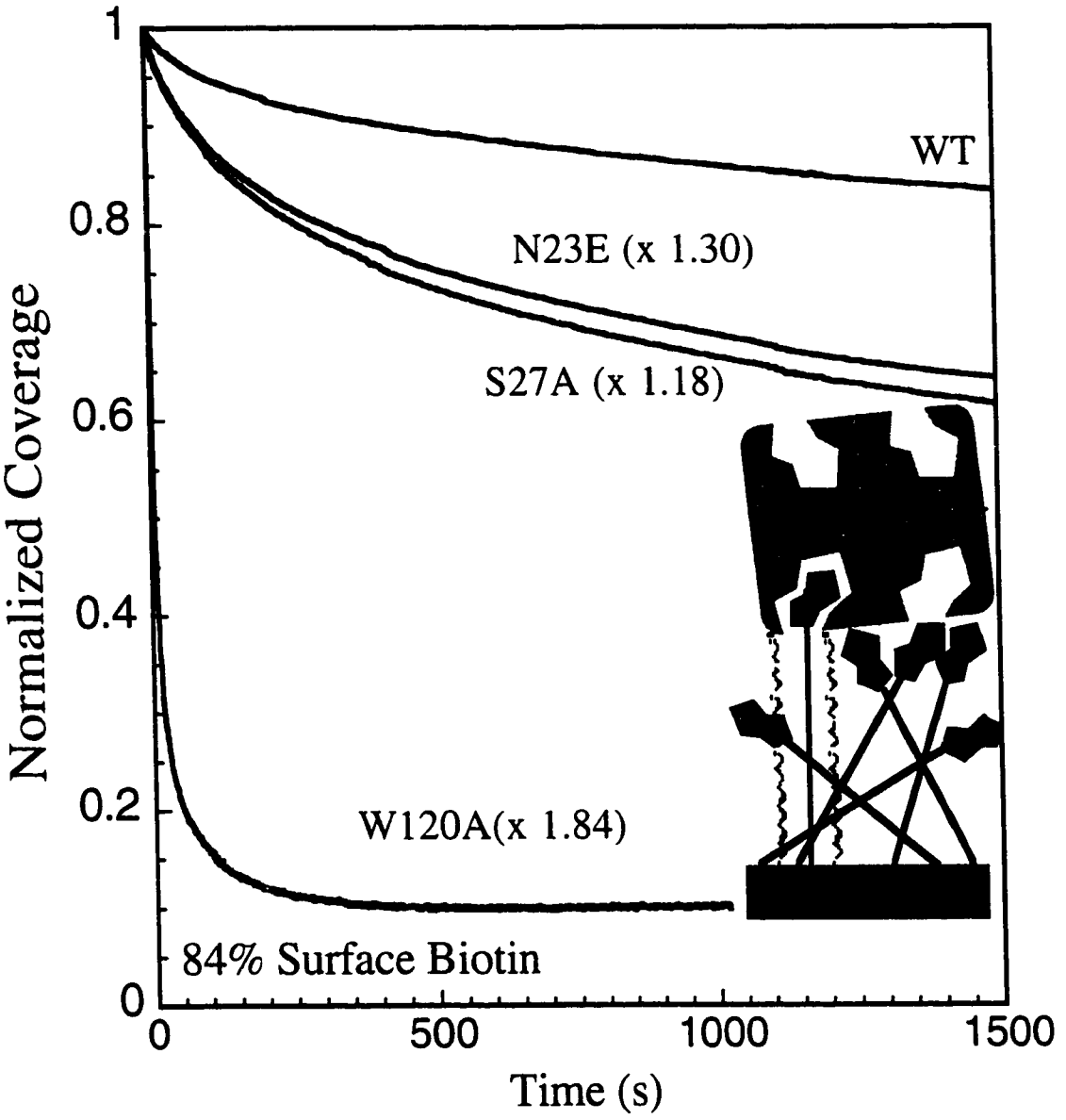


Figure 6.11 Normalized short - time regions of the desorption curves at 84% surface biotin. The curves were scaled by the amount shown in the (). The inset illustration depicts a SA bound to the surface by one biotin linkage due to the fact that the BATs are so closely surrounded by other BATs that none are accessible to protrude deep enough into the other pocket-like binding site of the SA. At this surface biotin composition the surface is expected to be dominated by such species.

TABLE FOR CHAPTER 6

Table 6.1 Rate constants for competitive desorption of SA-biotin for different SA variants. The pseudo-first-order biotin-SA off-rate constants determined in homogeneous aqueous solution at 25° C by Klumb and Stayton in previous studies of competitive decomposition in 0.05 mM biotin solution [1] along with the average values found for k and k` for each different SA variant by analysis of off-rate curves at 0.34% and 29% surface biotin and the values for k_b found from these values by application of Eq. (14).

SA Variant	k (solution) (literature) s^{-1}	k (0.35% BAT) s^{-1}	k' (29% BAT) s^{-1}	k_b (29% BAT) s^{-1}
WT	4.0×10^{-6}	1.4×10^{-5}	2.9×10^{-6}	1.7×10^{-5} *
S27A	1.2×10^{-3}	6.1×10^{-3}	3.9×10^{-6}	9.6
N23E	1.6×10^{-3}	2.1×10^{-3}	8.4×10^{-6}	5.2×10^{-1}
W120A	2.3×10^{1} *	2.3×10^{-1}	8.2×10^{-3}	6.4

* Unable to accurately determine these values due to limitations of the techniques.

CHAPTER 7: SURFACE PLASMON RESONANCE BASED QUANTITIFICATION OF TIGHT BINDING OF PHOSPHOLIPASE A_2 TO SURFACE-IMMOBILIZED PHOSPHOLIPID VESICLES

INTRODUCTION

A large variety of species bind to the phospholipid interfaces of biological membranes and initiate a variety of physiological effects. Therefore, it is of great interest to have a sensitive method for measuring the equilibrium binding constants and kinetics for such binding. Here, we present a new approach for this, and demonstrate it by measuring the equilibrium constant for the binding of a phospholipase to a phospholipid membrane.

Enzymes such as phospholipases that operate on highly water-insoluble components of biological membranes must be bound to the phospholipid interface to gain access to their substrates. Many of these enzymes are peripheral membrane proteins that can exist in both membrane bound and water soluble states. Such is the case for a large family of enzymes called phospholipase A_2 that hydrolyze the *sn*-2 ester of glycerophospholipids [1]. The affinity of these enzymes for vesicles depends dramatically on the molecular composition of the vesicle bilayer [2]. The role of specific amino acid residues on the membrane binding surface of 14-kDa secreted phospholipases A_2 that control interfacial binding selectivity is being delineated by site-directed mutagenesis studies [3-6]. Such studies are important because they offer clues about the function of secreted phospholipases A_2. For example, the high affinity of human group IIA phospholipase A_2 for phosphatidylglycerol vesicles is consistent with the proposed role of this enzyme as an anti-bacterial agent since bacterial membranes, but not mammalian cell membranes, are highly enriched in phosphatidylglycerol [7].

It is thus important to develop an efficient, accurate, and generally applicable method to quantify the binding of peripheral membrane proteins to phospholipid vesicles. Although several methods have been developed to obtain equilibrium dissociation constants for enzymes bound to phospholipid interfaces, they have limitations especially when the enzyme engages in high affinity interaction with vesicles. Resonance energy transfer has been used to detect the close proximity of an enzymatic tryptophan with a fluorescent membrane probe [2,8]. This method suffers from the fact that the enzyme must contain a fluorescent residue such as tryptophan. Introduction of a tryptophan residue on the membrane binding surface of the protein can greatly influence interfacial binding [5,9]. Furthermore, the energy transfer efficiency depends dramatically on the lipid composition of the vesicle [5], so that low fluorescent signals do not prove that there is little interfacial bonding. Many methods for measuring protein-vesicle dissociation constants involve separating vesicles from bulk solution by centrifugation followed by analysis of the amount of protein in the supernatant and pellet fractions. Vesicles composed only of lipids do not sediment well unless they are loaded with sucrose [10]. However, sucrose loaded phosphatidylcholine vesicles still resist sedimentation probably due to sucrose leakage [9]. In addition, recovery of protein from the centrifuge tube is often low unless detergent is present [10], and this problem is especially difficult when low concentrations of components must be used to measure dissociation constants in the sub-micromolar range. Vesicles containing a small amount of biotinylated phospholipid can be precipitated by crosslinking with solution phase streptavidin [11], but efficient precipitation occurs only in the presence of high micromolar to millimolar concentrations of lipids, so that dissociation constants below $\sim 10^{-6}$ M cannot be measured.

In this study, we have developed a simple method, based on intact vesicles immobilized on a surface plasmon resonance (SPR) sensor, to obtain the dissociation equilibrium constant, K_d, for the reaction Enzyme•Vesicle $\leftrightarrow$ Enzyme + Vesicle. It is applicable to lipids or lipid mixtures that form bilayered vesicles and to proteins that

undergo reversible association to their surface. We have used SPR to directly measure the maximal binding and equilibrium fractional surface coverage as the concentration of enzyme in a recirculating solution is stepped up and down in a manner similar to the stepwise surface titration method previously described for the study of protein-protein and protein-DNA interactions [12,13]. The surface coverage is plotted versus concentration as an adsorption isotherm to determine the equilibrium dissociation constant, K_d, for the enzyme-vesicle interaction. The sensitivity of this method is limited only by the sensitivity of the assay for the aqueous enzyme, and not by the sensitivity of the SPR system. Because of this, K_d values $< 10^{-10}$ M can be measured, in principle, which previously has not been possible. Furthermore, non-specific binding of protein to non-vesicle surfaces in the flow system is of no consequence since such binding does not contribute to the SPR signal.

SPR has been used to determine K_d of several toxins and enzymes on lipid surfaces [14-17]. However, most of those studies relied on the determination of the association and dissociation rate constants by fitting adsorption and / or desorption curves, which is not as accurate as measuring true equilibrium binding values and correlating the actual fractional coverage with the actual solution concentration. Also, the ratio of these rate constants gives the equilibrium dissociation constant only when a simple kinetic model applies: Enzyme•Vesicle $\leftrightarrow$ Enzyme + Vesicle, with only a single vesicle-bound enzyme species. Those fits often do not properly consider depletion of the solution near the surface due to adsorption and slow diffusion, which can lead to an inaccurate rate constant. A number of those studies were performed on biotinylated lipid vesicles immobilized using SA or avidin bound within the dextran coating of a BIACore SPR Chip [14,17], which might introduce further complications to the kinetic analysis due to other mass transfer issues specific to these hydrogel coatings [18,19]. In addition, those studies did not consider the significant decrease in solution-phase protein concentration due to nonspecific adsorption to the system (i.e. walls of tubing, flow cell, etc.), unlike our method which takes any loss of enzyme into account by measuring the

actual free enzyme concentration. Finally, none of those studies demonstrated that the probe vesicles were bound *intact* into the gel coatings. Here, we use a planar array of vesicles, which we prove to be intact and are not complicated by any surrounding gel.

An alternative method which has been commonly used for immobilization of phospholipids results in hybrid bilayer membranes (HBMs). Many HBMs have been previously described and well characterized [20-25]. Phospholipid vesicles or micelles adsorb, rupture and assemble onto a substrate with an alkylthiol monolayer, creating a single, planar monolayer of phospholipids with their hydrocarbon tails oriented towards the chains of the alkylthiol monolayer and their polar heads oriented up towards the buffer as shown in Fig. 7.1a. Although these surfaces are well-defined [20-24], there may be problems when using them as model membrane systems, primarily due to the decreased fluidity of the phospholipid monolayer caused by its constrainment by the underlying solid-like alkyl monolayer. This and other issues with HBMs will be discussed in more detail below.

The more physiological alternative we use here is shown schematically in Fig. 7.1b. The foundation of the substrate is a gold surface functionalized with a mixed monolayer of biotin-terminated alkylthiols (BAT) and poly(ethylene)oxide-terminated alkythiols (PEO). The PEO-terminated alkylthiols resist non-specific adsorption of solution components [26], while the BAT immobilizes a streptavidin (SA) linker monolayer onto the surface. These monolayers can immobilize SA onto a Au surface with high coverage, specificity, and activity [27-31].

The amount of SA that binds to these biotin - containing monolayers is dependent on the mole fraction of biotin in the film [27,28]. A monolayer with 33 mole percent BAT (which results if the dosing solution composition has a BAT / PEO ratio of 1/9) has been shown to lead to an optimal density of approximately 230 ng of SA / cm^2 (or 3800 $Å^2$ / SA) [28,29], which is ~80% of the expected density of a two-dimensional C(2,2,2) crystalline SA monolayer [32].

The streptavidin-biotin (SA-biotin) couple was chosen because of its high binding constant, ~10^{13} M^{-1} [33], and the fact that each streptavidin has four equivalent sites for biotin and is known to bind to these biotin - containing monolayers in such a way as to expose two of its sites away from the surface as shown in Fig 7.1b. For these reasons and the easy availability of biotinylated molecules, SA-functionalized surfaces have been utilized as a linker monolayer for attachment of antibodies, DNA, proteins and of their binding partners to gold surfaces [27,29-31]. Since, it has been demonstrated that phospholipid vesicles can be bound intact to aqueous-phase SA using the SA-biotin linkage without aggregation, stress, strain, fusion or lysing and with their fundamental properties maintained [14,34-36], it is expected that vesicles would make good physiological models if bound to a solid via SA. Here, we have used this SA linker monolayer to attach *intact* phospholipid vesicles containing a small amount of phospholipid with a biotinylated headgroup to these gold surfaces.

These surface-bound vesicles form the surface sites with which the K_d values of membrane-binding proteins have been determined using SPR. Because these immobilized vesicles are quite fluid and can swell in response to protein or ligand binding, they offer a more physiologically relevant model of membranes than systems based on planar monolayers or bilayers attached directly to solid surfaces, as we show below.

Here, this technique for measuring K_d is demonstrated for phospholipase A_2 from cobra venom (PLA_2) binding to zwitterionic phosphatidylcholine vesicles. Since PLA_2 selectively hydrolyzes the phosphatidylcholine *sn*-2 ester linkage when bound to vesicles, we have used vesicles of 1,2-dioleoyl-*sn*-glycero-3-phosphocholine ($DO_{et}PC$), where the fatty acyl ester linkages have been replaced with ether linkages to render them unreactive.

EXPERIMENTAL

The index of refraction for the proteins used was 1.57, with a specific volume of 0.77 mL / g [37,38]. For the $DO_{et}PC$, the index of refraction was estimated to be 1.49 [24,39-41] with a specific volume of 0.92 mL / g lipids [42]. For the intact vesicles and for protein adsorption on them, the depth-related attenuation of the surface plasmon intensity (~30%) was properly considered in applying the formalism of [38].

The data was taken using the planar SPR system as described in the Chapter 2. The circulating flow system is used for these experiments (except where noted). The solutions were recirculated to ensure that binding of the enzyme to all surfaces had reached steady state. Thereafter, analysis of a sample of solution from the reservoir should provide the concentration of free enzyme at equilibrium.

BIOTINYLATED VESICLES

Phospholipid vesicles were biotinylated by including during their preparation a small amount of phospholipid with biotin covalently attached to the headgroup as previously demonstrated [11,34]. Biotin-DOPE (in chloroform) was added to $DO_{et}PC$ (in chloroform) to yield 0.3 mol% of biotin-DOPE. The resulting solution was dried with N_2 and placed in a dessicator for ~30 minutes in vacuo to remove traces of chloroform. Buffer (50 mM Tris-HCl, 50 mM KCl, 1 mM $CaCl_2$, pH 7.6) was added to the dried lipids to yield ~1.0 - 0.1 mM of total lipid. After being subjected to six freeze-thaw / vortex cycles using a dry ice / acetone mixture and a 30°C water bath, the lipid solution was passed through two polycarbonate membranes (100 nm pore, Avestin Industries, Ottawa, Ontario) at least 19 times using the Lipofast extruder (Avestin Industries, Ottawa, Ontario), to produce the desired solution containing unilamellar vesicles with 0.3% biotin-DOPE lipids. As shown previously, the procedure produces vesicles of 80±25 nm diameter [43]. This diameter corresponds to 4.7×10^4 lipids (total)

per vesicle, or $2.9x10^4$ outer lipid head groups per vesicle, assuming that the packing density of lipids is the same as in Langmuir-Blodgett films (~0.7 nm^2 per lipid [44] at the reported surface pressure of the vesicle of ~30 dynes / cm [45]).

PREPARATION OF SURFACE IMMOBILIZED VESICLES

Samples with PEO / BAT monolayers were dosed with ~0.05 mg / mL of SA in 50 mM Tris buffer as previously described [28] until the adsorption appeared to reach a saturation coverage of SA as observed with SPR. After rinsing with pure buffer to remove any physisorbed SA, the above solution containing biotinylated vesicles was diluted ~50% with buffer and introduced in the same buffer and allowed to bind to the immobilized SA until equilibrated.

MEASURING K_D FOR PLA_2 BINDING

A solution of PLA_2 (in 50 mM Tris buffer) was circulated over the immobilized vesicles, and the adsorption was monitored with SPR. Additional PLA_2 was added to the solution to increase the concentration until the addition of more PLA_2 did not result in an increase in adsorption. This coverage was assumed to be saturation. The PLA_2 solution was then diluted by addition of buffer to the reservoir to remove a fraction of the bound PLA_2, and a new equilibrium was allowed to establish. This was repeated multiple times until all of the bound PLA_2 was removed from the surface as observed by SPR. In order to determine the PLA_2 concentration at each step, samples of the PLA_2 solution were collected from the reservoir (after establishing equilibrium) into containers with Tris buffer (25 mM Tris-HCl, 100 mM KCl, 0.1 mM $CaCl_2$, pH = 8.5) containing 1

mg / mL bovine serum albumin to decrease protein loss to the walls of the container. These collected samples were assayed for enzyme concentration as described below.

PLA_2 ASSAYS

A thiolester assay was used to determine the concentration of PLA_2 in samples from the SPR experiment [46]. The catalytic rate was determined by monitoring the change in absorbance at 412 nm as the free thiol formed on hydrolysis of the thiolester bond reacted with Ellman's reagent. The concentration of PLA_2 was found by calibration of the PLA_2 activity with standard solutions. Substrate for PLA_2 assays was prepared by drying a mixture of stock solutions of DTPM and dithio-$DC_{10}PM$ in chloroform with a stream of N_2. Remaining solvent was removed in a Speed-Vac (Savant Inc.) for 30 minutes. Buffer (1 mL of 25 mM Tris buffer) was added to make 90 mol% DTPM and 10 mol% dithio-$DC_{10}PM$, containing a total lipid concentration of 1.1 mM. Vesicles were formed by sonication [47]. Stock solutions of PLA_2 were diluted with 1 mg / mL BSA in order to prevent loss of enzyme to the walls of the tubes.

The final assay solution consisted of 0.8 mM Ellman's reagent, 100 μM DTPM, 10 μM dithio-$DC_{10}PM$, and 1 to 20 ng / mL PLA_2. First, the spectrophotometer thermostatted at 30° C was zeroed at 412 nm with just the substrate and buffer (25 mM Tris buffer stored in a 30° C water bath) in a cuvette. Then, Ellman's reagent was added, and, after equilibration for 2 minutes, a stable baseline was recorded. Finally, the PLA_2 was added and its activity determined by measuring the increase in absorption at 412 nm. Between each assay, the cuvette was rinsed with a 5% solution of sodium hypochlorite (Clorox), 0.1 M HCl and deionized water. Enzyme activity was linear with respect to amount of enzyme within the concentration range used.

DYE EXPERIMENTS

Extruded $DO_{et}PC$ vesicles were prepared in 50 mM Tris buffer containing 5 mM 5-carboxyfluorescein (Molecular Probes) as described above. The vesicles with entrapped dye were separated from the remaining dye using a size-exclusion column with G-25 Sephadex (Superfine, Pharmacia). The vesicles were allowed to adsorb on an immobilized SA layer as described previously. Rinses subsequent to adsorption were assayed for traces of dye to ensure that all non-bound vesicles and residual dye were removed from the flow cell. The surface-immobilized vesicles were exposed to a known volume of 1% Triton X100 solution to burst the vesicles, and the resulting solution was assayed. The amount of dye in the detergent rinse was determined using a standard curve with known dye concentrations in detergent-containing buffer. Fluorescence intensity was monitored with a fluorimeter using excitation and emission wavelengths of 492 nm and 518 nm, respectively.

HYBRID BILAYER MEMBRANES (HBMS)

The method used to prepare HBMs on Au was based on previous literature [22,25]. A 0.1 mM solution of $DO_{et}PC$ in buffer was sonicated as above, resulting in small unilamellar vesicles. The vesicles were injected into the SPR flow cell onto a Au substrate previously functionalized with hexadecanethiol as described above. The adsorption and assembly of the lipids were followed with SPR for 6-8 hours until a stable HBM was formed. PLA_2 was injected onto the rinsed HBM and its binding monitored.

RESULTS

The SPR response for a typical experimental sequence to study PLA_2 binding to immobilized, *intact* vesicles is shown in Figure 7.2. The experiment starts with the BAT / PEO monolayer already on the Au substrate in the SPR flow cell. At arrow a, SA was injected, resulting in a typical SPR wavelength shift of ~10.0 nm due to SA immobilization. The SPR shift for six experiments averaged 9.6±0.9 nm, which corresponds to the adsorption of $2.4\pm0.2\times10^{12}$ SA / cm^2 onto the surface (using the methods outlined above for converting SPR shifts to absolute surface coverages). This agrees with the known saturation coverage of SA, 2.6×10^{12} SA / cm^2, on this particular composition of BAT / PEO as previously observed [28].

At arrow b in Fig. 7.2, the biotinylated vesicles were injected into the flow cell. The SPR shift shown of ~30 nm was typical of the average shift observed in six similar experiments (30.4±1.9 nm). Doubling the concentration of vesicles did not lead to larger shifts, showing that saturation is reached (not shown). This saturation SPR shift corresponds to $8.9\pm0.6\times10^{14}$ lipids / cm^2 or 1.89×10^{10} vesicles / cm^2. This corresponds to about 1 vesicle for every 127 SA, or 5300 nm^2 per vesicle. This is close to the cross-sectional area of these ~80 nm diameter vesicles injected (~5030 nm^2), which proves that a dense layer of vesicles is formed on the chip surface. The total surface area of exposed lipid available for enzyme binding on one such vesicle is $\sim2.01\times10^{4}$ nm^2, which gives a total exposed vesicle area of 3.7 cm^2 per cm^2 of Au sensor surface.

To explore if *intact* vesicles were binding to the SA, vesicles with entrapped dye were adsorbed to an immobilized SA layer in the same manner as described above. After adsorption, the surface was rinsed, and no dye was found to be present in the rinse, indicating the vesicles were not leaking or bursting. The SA bound vesicles were then rinsed with a 1% Triton X100 solution, bursting the vesicles and releasing 2.0×10^{-11} moles of entrapped dye from the SPR flow cell, as measured by

fluorescence analysis. Based on the concentration of dye loaded into vesicles (5 mM) and the Au surface area in the flow cell (1.6 cm^2), this corresponds to $4.0x10^{-6}$ cm^3 of trapped volume, or $1.8x10^{10}$ vesicles per cm^2 of Au area. The vesicle surface density measured by SPR is within 5% of this value, confirming that the vesicles are adsorbed intact.

Arrow c in Fig. 7.2 indicates the time when the solution of PLA_2 was first injected across the chip surface containing the immobilized vesicles. The concentration of PLA_2 flowing through the flow cell was increased (indicated by successive arrows) until a saturation coverage was reached at a PLA_2 concentration ~$6x10^{-5}$ M. Four such experiments gave a saturation SPR shift of 6.3±1.4 nm for PLA_2, which corresponds to $6.9\pm1.5x10^{12}$ PLA_2 per cm^2 of Au (or $1.9x10^{12}$ PLA_2 per cm^2 of outer lipid surface) or approximately 1 PLA_2 per 74±16 outer lipid molecules (since each lipid occupies ~0.7 nm^2, see above). This packing density is lower than the value of 1 PLA_2 per 35-40 lipid headgroups reported for free vesicles [2], but the difference is small considering error bars. It can be attributed the proximity of the vesicles to the SA layer, the presence of biotin headgroups, and the close packing between vesicles, which probably renders some of their outer lipid surface inaccessible to protein. When no vesicles were immobilized on the surface (SA on the mixed BAT / PEO monolayer only), the PLA_2 adsorption was < 3% of that observed with vesicles present, as shown in Fig. 7.4, curve c (dotted curve). This proves that PLA_2 binding is specific to the vesicles.

By subsequently lowering the concentration of the PLA_2 in the solution (arrow d in Fig. 7.2.), the equilibrium fractional amount of PLA_2 bound diminished. Further decreases in PLA_2 concentration (subsequent arrows) led to greater decreases in bound PLA_2. This is continued until no more PLA_2 appears to rinse from the surface. This (which may not match the baseline observed prior to PLA_2 adsorption due non-specifically bound PLA_2 which may remain on the surface) is considered the baseline

defining zero PLA_2 bound to the surface. The fractional amounts of PLA_2 bound are plotted versus the log of concentrations (as determined by enzyme activity assay) at each of these steps in Fig. 7.3. The dissociation constant is defined as:

$$K_d = \frac{[\text{Sites}]_{free} \bullet [E]_{free}}{[\text{Sites}]_{bound}} \quad (1)$$

where $[\text{Sites}]_{bound}$ is the number of PLA_2 binding sites occupied by bound enzymes, $[\text{Sites}]_{free}$ is the number of sites free for binding and $[E]_{free}$ is the concentration of enzyme in solution, all at equilibrium. This can be rearranged and simplified to give:

$$[E]_{free} = \frac{f \bullet K_d}{1 - f} \quad (2)$$

where f is the fraction of PLA_2 binding sites on the vesicle surfaces that are filled (at equilibrium) at each specific PLA_2 concentration in the aqueous phase. We assume here that K_d is independent of f (i.e., no lateral interactions between bound enzymes). The amount of bound PLA_2 was determined by SPR and normalized to the amount of PLA_2 binding sites that are filled at saturation (f = 1.0). A binding site can be thought of as the number of lipids to which one PLA_2 binds (~40). The value of K_d for PLA_2 dissociating from $DO_{et}PC$ vesicles was determined by fitting the data of Fig. 7.3 to Eq. 2, as indicated by the solid curve through the data. The best-fit value of $6\pm2\times10^{-7}$ M is similar to the K_d of 2×10^{-7} M reported for phospholipase A_2 from a different cobra species binding to 1,2-ditetradecyl-phosphatidylcholine vesicles [47]. The data increase somewhat less steeply with concentrations than the fit to Eq. 2, suggesting that K_d actually increases slightly with coverage. This is typical of weakly repulsive lateral interactions between adsorbed species.

Measurements of PLA_2 binding were also performed on HBMs of $DO_{et}PC$. In these experiments, small unilamellar vesicles of $DO_{et}PC$ were first allowed

to fuse and assemble onto a pre-assembled hexadecanethiol monolayer on Au as described in Methods. Six such experiments produced an average of 9.9±2.4 nm SPR shift which corresponds to ~2.1±0.5x10^{14} lipids / cm^2 or 0.48±0.11 nm^2 per lipid, close to the value of ~0.7 nm^2 per lipid in a Langmuir Blodgett monolayer (see above). In these initial experiments, a 2x10^{-5} M PLA_2 solution in 50 mM Tris buffer was injected into the flow cell in a stop-flow manner.

Curve a (dashed curve) of Fig. 7.4 shows a 1.2 nm SPR shift as the PLA_2 binds to the HBM surface, typical of the average (1.2±0.2 nm) observed in such experiments. This SPR signal corresponds to 1.0±0.2x10^{12} PLA_2 / cm^2 or about 1 PLA_2 for every 208±35 lipid headgroups. Curve b (solid curve) of Fig. 7.4 shows the same experiment performed with the intact vesicles immobilized on the chip surface instead. In this case, a SPR shift of ~6.4 nm results, corresponding to 7.0x10^{12} PLA_2 / cm^2 (or 1.8x10^{12} PLA_2 / cm^2 of outer lipid surface). This is a packing density of one PLA_2 for every ~73 lipid headgroups, 3-fold larger than on the HBM.

DISCUSSION

There are a number advantages of using the intact immobilized vesicle surface as a model membrane for studying peripheral protein - membrane interactions or the interactions of other species with membranes or with membrane - bound species. The experiments presented here have clearly demonstrated that the biotinylated vesicles can be bound *intact* onto the surface of an SPR sensor chip without significant changes in the properties of the vesicles. The vesicles bind PLA_2 with a similar equilibrium constant and packing density as aqueous-phase vesicles. Our results demonstrate that a dense layer of vesicles can be adsorbed onto the surface, so that signals (like those due to protein binding in SPR) are large for this geometry. In addition, nonspecific binding of

the protein to the SPR chip was not a problem with this geometry, at least with the cobra venom PLA_2.

The HBMs offer an advantage over the intact vesicle surface in that their surface is entirely planar and orientational effects could be studied. However, the PLA_2 binding results clearly indicate there is a difference in the quality of the lipid layer in vesicles and HBMs which also has also been observed by others [48]. Even though the vesicles have nearly four times more surface area, for the same concentration of enzyme, the vesicles still allow almost three times higher enzyme-to-lipid binding ratio than the HBM surface. The net effect leads to > 6-fold signal enhancement and a more physiological environment. Recently, it has been generally suspected that HBMs are more constrained than true vesicle membranes [23,49]. This difference is attributed to the changes in the physical properties of the lipid layer in HBMs, such as density or fluidity, which may in turn restrict an enzyme's ability to adopt a favored conformation [50]. More importantly perhaps, when a peripheral membrane protein such as the PLA_2 studied here adsorbs to the surface, it partially inserts partially between the lipids [51], and thus squeezes into the lipid layer to some extent. With an HBM, this leads to very energetically unfavorable compression of the lipid monolayer, or an effectively higher surface pressure. (The surface pressure and energy of a lipid bilayer increases dramatically with packing density.) This destabilization may shut down enzyme adsorption at the unusually low saturation coverage observed here on HBMs (one PLA_2 per ~208 lipid headgroups).

This is apparently not a problem with the immobilized, intact vesicles which bind a higher density of PLA_2 (one per ~74 lipids). We note that free vesicles are considered in such respects as very good models of real biological cell membranes. Since water can easily pass through the bilayer of intact vesicles, their diameter will simply increase as the adsorbing protein squeezes into the bilayer, so there will be no marked increase in surface pressure or bilayer energy upon protein adsorption to the

intact vesicle surfaces. This allows supported vesicles to more closely mimic the behavior of free vesicles or real cells and, therefore, provide more meaningful saturation coverages and binding constants for protein adsorption. However, HBMs are better from a non-specific adsorption standpoint as there are no other surfaces besides exposed lipids, unlike the intact vesicle monolayer which potentially offers sites for non-specific binding to the underlying SA or BAT / PEO. Much work has been done to prepare improved model membranes on planar solid surfaces that mimic biological membranes. For example, Heyse et al. [52] have recently used unique types of planar trilayer model membranes and obtained an equilibrium binding constant for transducin binding to the membrane that is only ~50% below the value obtained for disc membranes [53].

The surface bound intact vesicles used here conveniently provide a direct way to measure protein-membrane affinities, whose values should be more physiologically relevant than those of such planar systems. The method avoids some previously described problems [12,13] that are inherent in measuring association and dissociation rate constants to obtain K_d.

The stepwise titration shown in Fig. 7.2 allows adsorption measurement at a number of protein concentrations on the same surface instead of multiple measurements on different surfaces, which may result in error due to differences in the availability of binding sites for the different experiments. Also, when analyzing adsorption or dissociation kinetic curves, mass transfer must be considered, particularly in cases of low concentration or high affinity interactions. Since the present technique measures the amount of bound enzyme at equilibrium, such issues are avoided.

By adding the direct assay of free enzyme in solution to this SPR method, the accuracy of this method is improved. In previous studies, it was assumed that protein loss due to non-specific binding to the flow or injection tubes was negligible compared to the initial concentration of protein in solution, and in some cases the free protein concentration was calculated by subtracting the amount of protein bound to the vesicles

from the total protein in the system initially [12,13]. This assumption may be valid when the protein concentration in the bulk solution is high, but non-specific binding is a common problem at the low concentrations needed (< 10^{-6} M) for strongly-binding protein. With the present method, assaying the solution at equilibrium for each titration step avoids these issues.

In the method reported here, it is the sensitivity of the free-protein enzyme assay, not the SPR measurement, which limits the ability to measure low values of K_d in the case of tight protein-vesicle interaction. The thiolester phospholipid assay used here allows the detection of ~10^{-9} M of enzyme [46]. In order to measure higher affinity interactions, a more sensitive assay method could be employed. In the case of PLA_2, a radioactivity assay would allow for measurement of K_d <10^{-10} M [54]. For such strong binding, the off rates would be so slow that the waiting time for equilibrium to be established might be prohibitively long when decreasing the concentration of enzyme (i.e. amount bound). The drift rate of the SPR baseline might become comparable to the real signal, but this can be minimized by inclusion of an internal reference channel [55,56]. One could also determine the equilibrium bound enzyme coverage after extremely long equilibration times (over which the baseline drifts too much for direct measurement) by a difference method, whereby the number of remaining free sites is measured from the rapid SPR response to a post-equilibrium, step-wise increase in the enzyme concentration to a large value, which quickly saturates the remaining free sites. To overcome long waiting times, one could do the experiment with increasing concentration increments, whereby the difference method also would not be necessary.

There are still some issues that need to be further examined, so it is not yet clear that this technique can be applied to all enzymes and lipids. Some anomalous behavior of anionic vesicles in the presence of Ca^{2+}, even when the vesicles are bound prior to exposure to Ca^{2+}, were observed (experiment not presented). Until the exact nature of these problems is clarified, which are most likely due to vesicle aggregation

facilitated by Ca^{2+}, this method is not yet recommended for highly charged vesicles in the presence of Ca^{2+}. In addition, although nonspecific binding to the sensor surface was not a problem with cobra venom PLA_2, we did encounter enough nonspecific binding of the human group IIA PLA_2 to our SPR chip surface in the absence of adsorbed vesicles, such that control experiments were necessary to measure and correct for it. (This protein non-specifically sticks strongly to many types of surfaces including various types of plastic and glass [57].) Finally, we point out that the method also will allow study of the kinetics of formation and dissociation of complexes formed by proteins (and other ligands) with the surfaces of vesicles. Note that Fig. 7.4 (curve a), for example, also contains this kinetic information for PLA_2 / vesicle interactions.

CONCLUSIONS

A new method for studying the equilibrium (and kinetic) constants for the binding of species to biological membranes has been demonstrated. We have bound ~80 nm diameter phospholipid vesicles intact onto a gold SPR sensor surface using a streptavidin linker layer with biotinylated alkyl thiols, and biotinylated lipids incorporated at low concentrations (~0.3%) in the vesicles. An equilibrium titration approach was used to measure the dissociation constant for PLA_2 binding to such biotinylated-$DO_{et}PC$ vesicles, using SPR to monitor the amount of PLA_2 which bound to these vesicles. The concentration of free enzyme at equilibrium was directly determined by a catalytic assay to insure accuracy and sensitivity. A value of $K_d = 6\pm2\times10^{-7}$ M was found, with a saturation coverage of one PLA_2 for every ~74±16 outer lipid headgroups, similar to the values expected based on studies of free vesicles. Because the method is limited by the sensitivity of the assay for the free enzyme, K_d values as low as 10^{-10} M should be measurable, which has not been previously possible with the more commonly used fluorescence- or centrifugation-based measurements. This will facilitate the study

of the binding of a variety of mutant phospholipases and other species to membranes for which the determination of K_d has been difficult with other methods.

NOTES TO CHAPTER 7

[1] Verheij, H. M.; Slotboom, A. J.; De Haas, G. H. *Rev. Physiol. Biochem. Pharmacol.* **1981**, *91*, 91-203.

[2] Ramirez, F.; Jain, M. K. *Proteins: Structure, Function, and Genetics* **1991**, *9*, 229-239.

[3] Han, S. K.; Yoon, E. T.; Scott, D. L.; Sigler, P. B.; Cho, W. *J. Biol. Chem.* **1997**, *272*, 3573-3582.

[4] Liu, X.; Zhu, H.; Huang, B.; Rogers, J.; Yu, B. Z.; Kumar, A.; Jain, M. K.; Sundaralingam, M.; Tsai, M. D. *Biochemistry* **1995**, *34*, 7322-34.

[5] Baker, S. F.; Othman, R.; Wilton, D. C. *Biochemistry* **1998**, *37*, 13203-13211.

[6] Snitko, Y.; Koduri, R. S.; Han, S. K.; Othman, R.; Baker, S. F.; Molini, B. J.; Wilton, D. C.; Gelb, M. H.; Cho, W. H. *Biochemistry* **1997**, *36*, 14325-14333.

[7] Foreman-Wykert, A. K.; Weinrauch, Y.; Elsbach, P.; Weiss, J. *J Clin Invest* **1999**, *103*, 715-21.

[8] Nalefski, E. A.; Sultzman, L. A.; Martin, D. M.; Kriz, R. W.; Towler, P. S.; Knopf, J. L.; Clark, J. D. *J. Biol. Chem.* **1994**, *269*, 18239-18249.

[9] Cho, W.; Gelb, M. H. *Unpublished observations* .

[10] Buser, C. A.; Sigal, C. T.; Resh, M. D.; McLaughlin, S. *Biochemistry* **1994**, *33*, 13093-13101.

[11] Tortorella, D.; Ulbrandt, N. D.; London, E. *Biochemistry* **1993**, *32*, 9181-9188.

[12] Hall, D. R.; Cann, J. R.; Winzor, D. J. *Analytical Biochemistry* **1996**, *235*, 175-184.

[13] Schuck, P.; Millar, D. B.; Kortt, A. A. *Analytical Biochemistry* **1998**, *265*, 79-91.

[14] MacKenzie, C. R.; Hirama, T.; Lee, K. K.; Altman, E.; Young, N. M. *Journal of Biological Chemistry* **1997**, *9*, 5533-5538.

[15] Masson, L.; Mazza, A.; Brousseau, R.; Tabashnik, B. *Journal of Biological Chemistry* **1995**, *270*, 11887-11896.

[16] Stachowiak, O.; Dolder, M.; Wallimann, T. *Biochemistry* **1996**, *35*, 15522-15528.

[17] Lange, C.; Koch, K.-W. *Biochemistry* **1997**, *36*, 12019-12026.

[18] Glaser, R. W. *Analytical Biochemistry* **1993**, *213*, 152-161.

[19] Schuck, P. *biophysical Journal* **1996**, *70*, 1230-1249.

[20] Meuse, C. W.; Niaura, G.; Lewis, M. L.; Plant, A. L. *Langmiur* **1998**, *15*, 1604-1611.

[21] Meuse, C. W.; Krueger, S.; Majkrzak, C. F.; Dura, J. A.; Fu, J.; Connor, J. T.; Plant, A. L. *Biophysical Journal* **1998**, *74*, 1388-1398.

[22] Plant, A. L. *Langmuir* **1993**, *9*, 2764-2767.

[23] Plant, A. L.; Gueguetchkeri, M.; Yap, W. *Biophysical Journal* **1994**, *67*, 1126-1133.

[24] Plant, A. L.; Brigham-Burke, M.; Petrella, E. C.; O'Shannessy, D. J. *Analytical Biochemistry* **1995**, *226*, 342.

[25] Lingler, S.; Rubinstein, I.; Knoll, W.; Offenhausser, A. *Langmuir* **1997**, *13*, 7085-7091.

[26] Prime, K.; Whiteside, G. M. *J. Am. Chem. Soc.* **1993**, *115*, 10714-10721.

[27] Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

[28] Jung, L. S.; Nelson, K. E.; Campbell, C. T.; Stayton, P. S.; Yee, S. S.; Perez-Luna, V.; Lopez, G. P. *Sensors and Actuators B* **1999**, *54*, 137-144.

[29] Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

[30] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

[31] Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

[32] Darst, S. A.; Ahlers, M.; Meller, P. H.; Kubalek, E. W.; Blankenburg, R.; Ribi, H. O.; Ringsdorf, H.; Kornberg, R. D. *Biophys. J.* **1991**, *59*, 387-96.

[33] Green, N. M. *Adv. Protein Chem.* **1975**, *29*, 85-133.

[34] Bayer, E. A.; Rivnay, B.; Skutelsky, E. *Biochimica et Biophysica Acta* **1979**, *550*, 464-473.

[35] Chiruvolu, S.; Walker, S.; Israelachvili, J.; Schmitt, F.-J.; Leckband, D.; Zasadzinski, J. A. *Science* **1994**, *264*, 1753-1756.

[36] Pashkov, V. N.; Tsurupa, G. P.; Griko, N. B.; Skopinskaya, S. N.; Yarkov, S. P. *Analytical Biochemistry* **1992**, *207*, 341-347.

[37] Darnell, J. E.; Lodish, H.; Baltimore, D. *Molecular Cell Biology*; Scientific American Books: New York, 1990.

[38] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[39] Huang, W.-T.; Levitt, D. G. *Biophysical Journal* **1977**, *17*, 111-128.

[40] Engelsen, D. D. *Surface Science* **1976**, *56*, 272-280.

[41] Florin, E. L.; Gaub, H. E. *Biophysical Journal* **1993**, *64*, 375-383.

[42] Small, D. M. *The Physical Chemistry of Lipids: From Alkanes to Phospholipids*; Plenum Press: New York, 1986; Vol. 4.

[43] MacDonald, R. C.; MacDonald, R. I.; Menco, B. P. M.; Rakeshita, K.; Subbarao, N. K.; Hu, L. *Biochimica et Biophysica Acta* **1991**, *1061*, 297-303.

[44] Lund-Katz, S.; Laboda, H. M.; McLean, L. R.; Phillips, M. C. *Biochemistry* **1988**, *27*, 3416-3423.

[45] Demel, R. A.; Kessel, W. S. M. G. V.; Zwaal, R. F. A.; Roelofsen, B.; Deenen, L. L. M. V. *Biochimica et Biophysica Acta* **1975**, *406*, 97-107.

[46] Hendrickson, S. H.; Hendrickson, E. K.; Dybvig, R. H. *J. Lipid. Res.* **1983**, *24*, 1532.

[47] Jain, M. K.; Egmond, M. R.; Verheij, H. M.; Apitz-Castro, R.; Dijkman, R.; Haas, G. H. D. *Biochim. Biophys. Acta* **1982**, *688*, 341-348.

[48] Wilkening, G.; Linke, T.; Sandhoff, K. *Journal of Biological Chemistry* **1998**, *273*, 30271-30278.

[49] Plant, A. L. *Langmuir* **1999**, *15*, 5128-5135.

[50] Dahmen-Levison, U.; Brezesinski, G.; Mohwald, H. *Thin Solid Films* **1998**, *327-329*, 616-620.

[51] Heyse, S.; Ernst, O. P.; Dienes, Z.; Hofmann, K. P.; Vogel, H. *Biochemistry* **1998**, *37*, 507-522.

[52] Schleicher, A.; Hofmann, K.-P. *J. Membrane Biol.* **1987**, *95*, 271-281.

[53] Reynolds, L. J.; Washburn, W. N.; Deems, R. A.; Dennis, E. A. *Methods in Enzymology* **1991**, *197*, 3.

[54] O'Brien, M. J.; Brueck, S. R. J.; Perez-Luna, V. H.; Tender, L. M.; Lopez, G. P. *Biosensors and Bioelectronics* **1999**, *14*, 145-154.

[55] Homola, J.; Lu, H. B.; Yee, S. S. *Electronics Letters* **1999**, *35*, 1105-1106.

[56] Gelb, M. H. *Unpublished observations* .

[1] Verheij, H. M.; Slotboom, A. J.; De Haas, G. H. *Rev. Physiol. Biochem. Pharmacol.* **1981**, *91*, 91-203.

[2] Ramirez, F.; Jain, M. K. *Proteins: Structure, Function, and Genetics* **1991**, *9*, 229-239.

[3] Han, S. K.; Yoon, E. T.; Scott, D. L.; Sigler, P. B.; Cho, W. *J. Biol. Chem.* **1997**, *272*, 3573-3582.

[4] Liu, X.; Zhu, H.; Huang, B.; Rogers, J.; Yu, B. Z.; Kumar, A.; Jain, M. K.; Sundaralingam, M.; Tsai, M. D. *Biochemistry* **1995**, *34*, 7322-34.

[5] Baker, S. F.; Othman, R.; Wilton, D. C. *Biochemistry* **1998**, *37*, 13203-13211.

[6] Snitko, Y.; Koduri, R. S.; Han, S. K.; Othman, R.; Baker, S. F.; Molini, B. J.; Wilton, D. C.; Gelb, M. H.; Cho, W. H. *Biochemistry* **1997**, *36*, 14325-14333.

[7] Foreman-Wykert, A. K.; Weinrauch, Y.; Elsbach, P.; Weiss, J. *J Clin Invest* **1999**, *103*, 715-21.

[8] Nalefski, E. A.; Sultzman, L. A.; Martin, D. M.; Kriz, R. W.; Towler, P. S.; Knopf, J. L.; Clark, J. D. *J. Biol. Chem.* **1994**, *269*, 18239-18249.

[9] Cho, W.; Gelb, M. H. *Unpublished observations* .

[10] Buser, C. A.; Sigal, C. T.; Resh, M. D.; McLaughlin, S. *Biochemistry* **1994**, *33*, 13093-13101.

[11] Tortorella, D.; Ulbrandt, N. D.; London, E. *Biochemistry* **1993**, *32*, 9181-9188.

[12] Hall, D. R.; Cann, J. R.; Winzor, D. J. *Analytical Biochemistry* **1996**, *235*, 175-184.

[13] Schuck, P.; Millar, D. B.; Kortt, A. A. *Analytical Biochemistry* **1998**, *265*, 79-91.

[14] Lange, C.; Koch, K.-W. *Biochemistry* **1997**, *36*, 12019-12026.

[15] MacKenzie, C. R.; Hirama, T.; Lee, K. K.; Altman, E.; Young, N. M. *Journal of Biological Chemistry* **1997**, *9*, 5533-5538.

[16] Masson, L.; Mazza, A.; Brousseau, R.; Tabashnik, B. *Journal of Biological Chemistry* **1995**, *270*, 11887-11896.

[17] Stachowiak, O.; Dolder, M.; Wallimann, T. *Biochemistry* **1996**, *35*, 15522-15528.

[18] Glaser, R. W. *Analytical Biochemistry* **1993**, *213*, 152-161.

[19] Schuck, P. *biophysical Journal* **1996**, *70*, 1230-1249.

[20] Meuse, C. W.; Niaura, G.; Lewis, M. L.; Plant, A. L. *Langmiur* **1998**, *15*, 1604-1611.

[21] Meuse, C. W.; Krueger, S.; Majkrzak, C. F.; Dura, J. A.; Fu, J.; Connor, J. T.; Plant, A. L. *Biophysical Journal* **1998**, *74*, 1388-1398.

[22] Plant, A. L. *Langmuir* **1993**, *9*, 2764-2767.

[23] Plant, A. L.; Gueguetchkeri, M.; Yap, W. *Biophysical Journal* **1994**, *67*, 1126-1133.

[24] Plant, A. L.; Brigham-Burke, M.; Petrella, E. C.; O'Shannessy, D. J. *Analytical Biochemistry* **1995**, *226*, 342.

[25] Lingler, S.; Rubinstein, I.; Knoll, W.; Offenhausser, A. *Langmuir* **1997**, *13*, 7085-7091.

[26] Prime, K.; Whiteside, G. M. *J. Am. Chem. Soc.* **1993**, *115*, 10714-10721.

[27] Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

[28] Jung, L. S.; Nelson, K. E.; Campbell, C. T.; Stayton, P. S.; Yee, S. S.; Perez-Luna, V.; Lopez, G. P. *Sensors and Actuators B* **1999**, *54*, 137-144.

[29] Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

[30] Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

[31] Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

[32] Darst, S. A.; Ahlers, M.; Meller, P. H.; Kubalek, E. W.; Blankenburg, R.; Ribi, H. O.; Ringsdorf, H.; Kornberg, R. D. *Biophys. J.* **1991**, *59*, 387-96.

[33] Green, N. M. *Adv. Protein Chem.* **1975**, *29*, 85-133.

[34] Bayer, E. A.; Rivnay, B.; Skutelsky, E. *Biochimica et Biophysica Acta* **1979**, *550*, 464-473.

[35] Chiruvolu, S.; Walker, S.; Israelachvili, J.; Schmitt, F.-J.; Leckband, D.; Zasadzinski, J. A. *Science* **1994**, *264*, 1753-1756.

[36] Pashkov, V. N.; Tsurupa, G. P.; Griko, N. B.; Skopinskaya, S. N.; Yarkov, S. P. *Analytical Biochemistry* **1992**, *207*, 341-347.

[37] Darnell, J. E.; Lodish, H.; Baltimore, D. *Molecular Cell Biology*; Scientific American Books: New York, 1990.

[38] Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

[39] Huang, W.-T.; Levitt, D. G. *Biophysical Journal* **1977**, *17*, 111-128.

[40] Engelsen, D. D. *Surface Science* **1976**, *56*, 272-280.

[41] Florin, E. L.; Gaub, H. E. *Biophysical Journal* **1993**, *64*, 375-383.

[42] Small, D. M. *The Physical Chemistry of Lipids: From Alkanes to Phospholipids*; Plenum Press: New York, 1986; Vol. 4.

[43] MacDonald, R. C.; MacDonald, R. I.; Menco, B. P. M.; Rakeshita, K.; Subbarao, N. K.; Hu, L. *Biochimica et Biophysica Acta* **1991**, *1061*, 297-303.

[44] Lund-Katz, S.; Laboda, H. M.; McLean, L. R.; Phillips, M. C. *Biochemistry* **1988**, *27*, 3416-3423.

[45] Demel, R. A.; Kessel, W. S. M. G. V.; Zwaal, R. F. A.; Roelofsen, B.; Deenen, L. L. M. V. *Biochimica et Biophysica Acta* **1975**, *406*, 97-107.

[46] Hendrickson, S. H.; Hendrickson, E. K.; Dybvig, R. H. *J. Lipid. Res.* **1983**, *24*, 1532.

[47] Jain, M. K.; Egmond, M. R.; Verheij, H. M.; Apitz-Castro, R.; Dijkman, R.; Haas, G. H. D. *Biochim. Biophys. Acta* **1982**, *688*, 341-348.

[48] Wilkening, G.; Linke, T.; Sandhoff, K. *Journal of Biological Chemistry* **1998**, *273*, 30271-30278.

[49] Plant, A. L. *Langmuir* **1999**, *15*, 5128-5135.

[50] Dahmen-Levison, U.; Brezesinski, G.; Mohwald, H. *Thin Solid Films* **1998**, *327-329*, 616-620.

[51] Verger, R.; Rietsch, J.; Dam-Mieras, M. C. E. v.; Haas, G. H. D. *J. Biol. Chem.* **1976**, *251*, 3128-3133.

[52] Heyse, S.; Ernst, O. P.; Dienes, Z.; Hofmann, K. P.; Vogel, H. *Biochemistry* **1998**, *37*, 507-522.

[53] Schleicher, A.; Hofmann, K.-P. *J. Membrane Biol.* **1987**, *95*, 271-281.

[54] Reynolds, L. J.; Washburn, W. N.; Deems, R. A.; Dennis, E. A. *Methods in Enzymology* **1991**, *197*, 3.

[55] O'Brien, M. J.; Brueck, S. R. J.; Perez-Luna, V. H.; Tender, L. M.; Lopez, G. P. *Biosensors and Bioelectronics* **1999**, *14*, 145-154.

[56] Homola, J.; Lu, H. B.; Yee, S. S. *Electronics Letters* **1999**, *35*, 1105-1106.

[57] Gelb, M. H. *Unpublished observations* .

FIGURES FOR CHAPTER 7

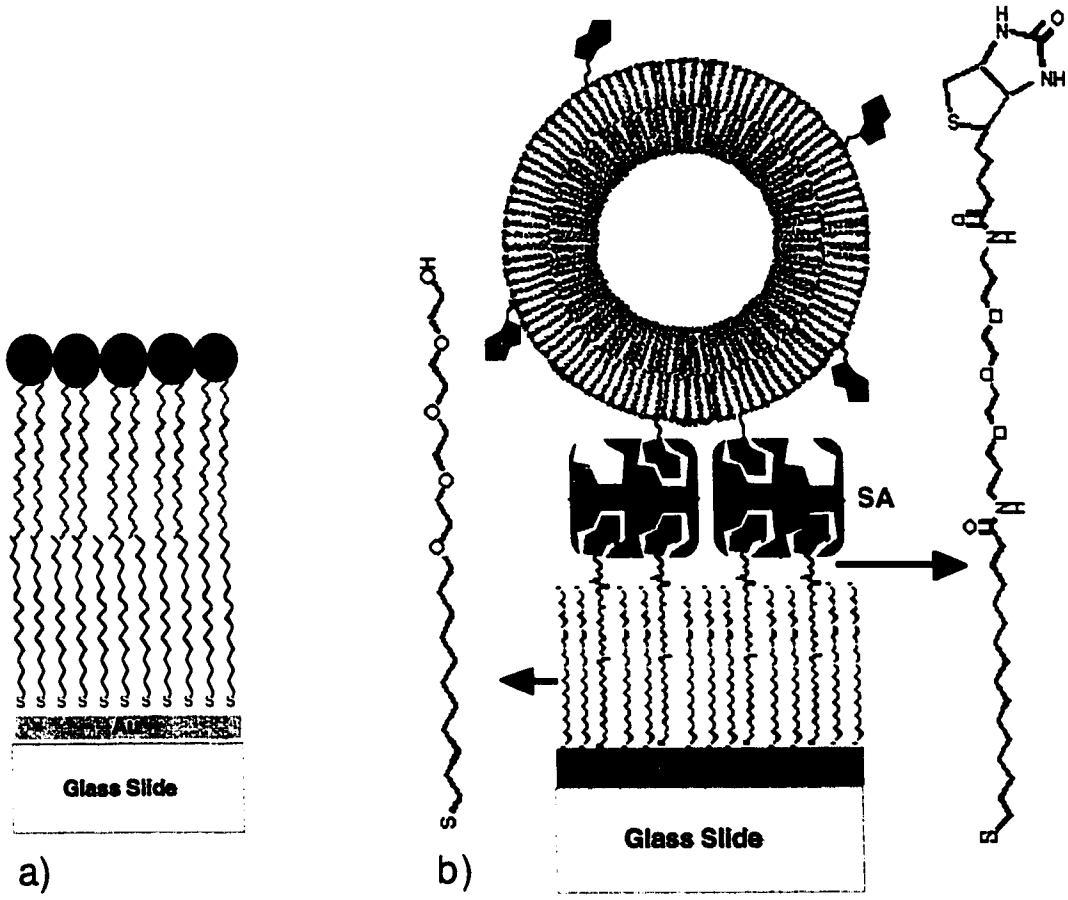


Figure 7.1a). Schematic of a hybrid bilayer membrane (HBM) (not to scale). The top layer represents the lipids with their polar heads (filled circles), while the lower layer represents the alkylthiols bound to the Au substrate. 1b). Schematic of the intact immobilized biotinylated vesicle monolayer (not to scale). The biotin in a mixed $DO_{et}PC$ - biotinylated vesicle is bound to free binding sites on streptavidin which is itself bound to biotin in a mixed BAT / PEO monolayer adsorbed on the Au substrate.

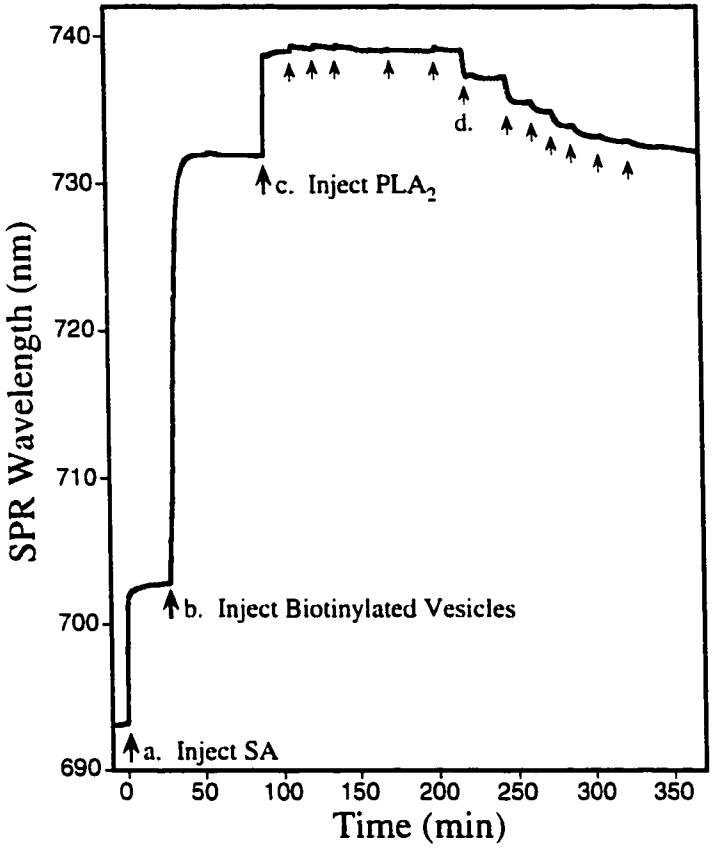


Figure 7.2 Construction of a biotinylated-$DO_{et}PC$ vesicle substrate (starting with the mixed BAT / PEO layer on Au) and determination of K_d for PLA_2 as observed with SPR. a) A recirculating mixture of 0.05 mg / mL SA in 50 mM Tris buffer was introduced to the flow cell, resulting in an increase in the SPR wavelength due to binding of SA to the biotin in the mixed BAT / PEO layer. After ~20 minutes, the sample was rinsed with pure buffer. b) A 0.8 mM solution of biotinylated-$DO_{et}PC$ vesicles was introduced to the flow cell and an increase in SPR wavelength was seen due to binding of the vesicles to the SA layer. After ~30 minutes the adsorption appeared to reach saturation, and the sample was rinsed with buffer. c) A $2.5x10^{-5}$ M recirculating solution of PLA_2 was introduced to the flow cell, and the PLA_2 bound to the vesicles. Additional PLA_2 was then added (indicated by arrows) to the solution to increase its concentration until a saturation coverage was obtained. d) The PLA_2 solution was diluted seven times (indicated by arrows), resulting in different $[PLA_2]_{bound}$ for each solution concentration, and was eventually replaced with buffer to completely rinse (see text) the PLA_2 from the surface. Solution-phase aliquots were taken to assay for free PLA_2 concentration just before each arrow (i.e., before changing the PLA_2 concentration but after establishing equilibrium).

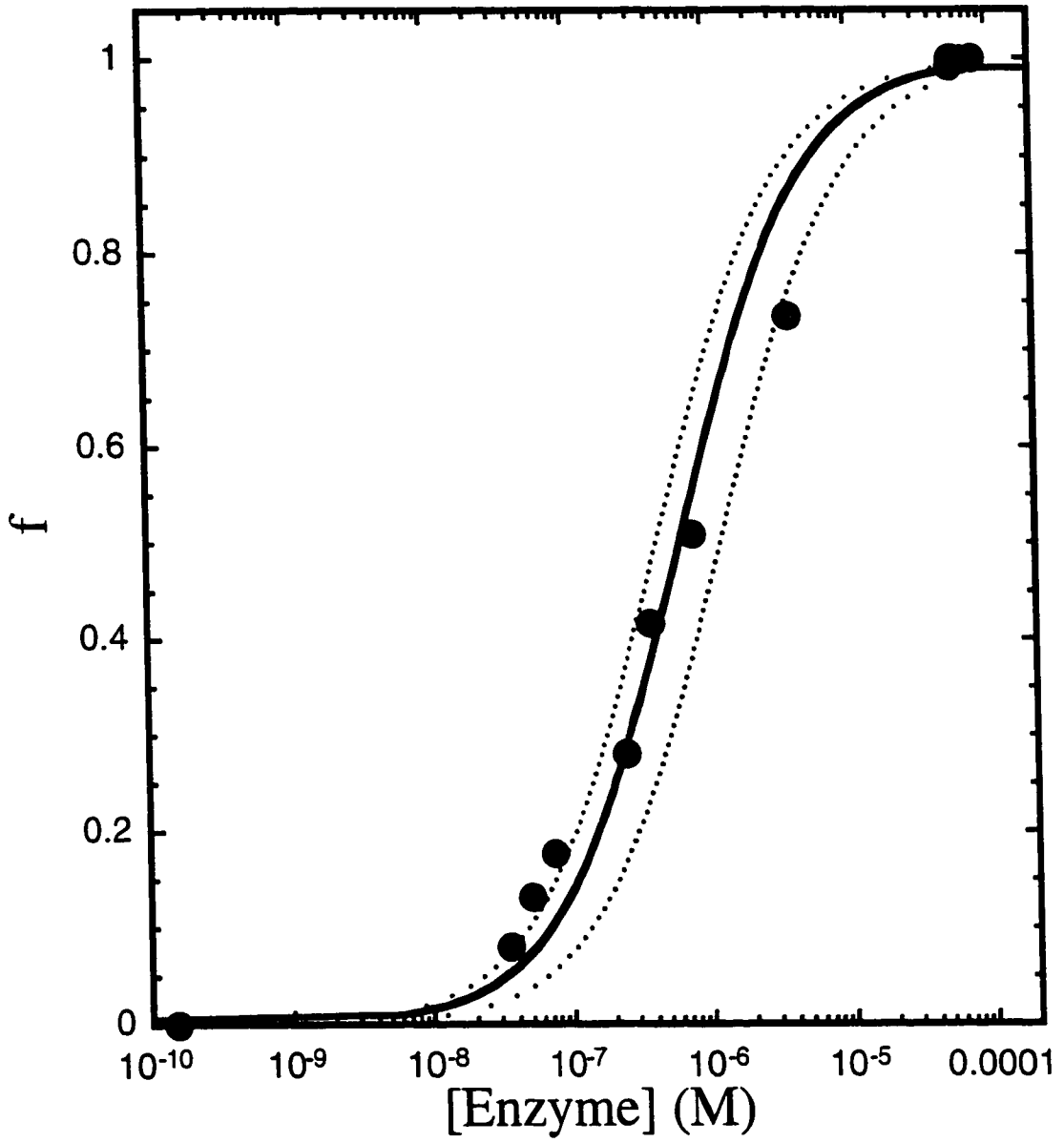


Figure 7.3 Plot of the fraction of PLA_2 binding sites on the vesicles that are filled (at equilibrium) by PLA_2 (f) versus the log of the solution-phase PLA_2 concentration. The solid curve represents the regression fit to Eq. 2 giving a K_d of $6\pm2\times10^{-7}$ M. (The dotted curves represent plots for K_d at these error-bar limits.)

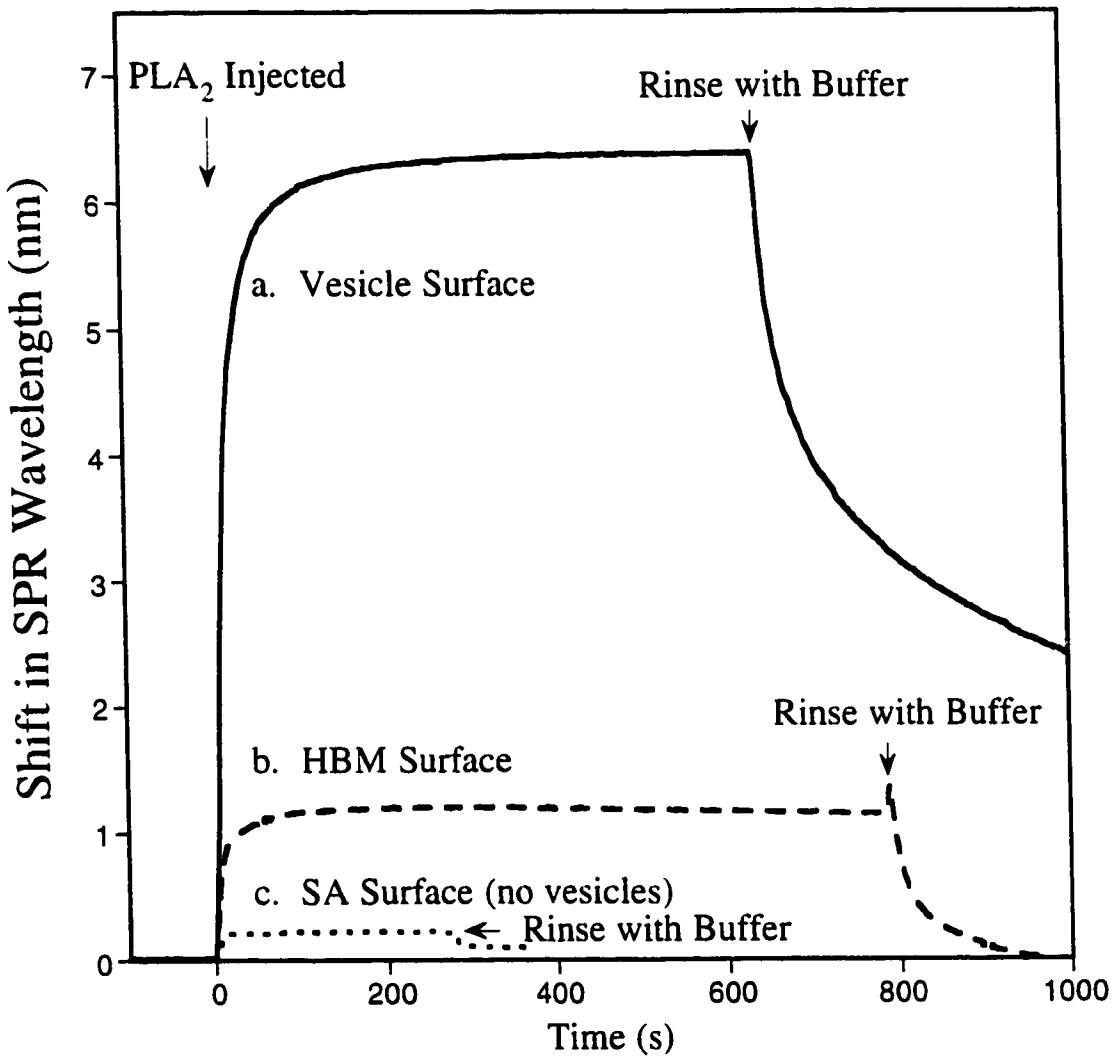


Figure 7.4 Stop-flow SPR experiments of PLA_2 adsorption on lipid surfaces. a) Adsorption of PLA_2 from a solution of the same PLA_2 concentration onto intact biotinylated $DO_{et}PC$ vesicles immobilized to the SPR chip surface through a SA monolayer. b) Adsorption of PLA_2 from a $2x10^{-5}$ M solution onto a HBM of $DO_{et}PC$. c) Control experiment showing the adsorption of PLA_2 on the same surface as (b) but without vesicles (SA on the BAT / PEO monolayer).

BIBLIOGRAPHY

Allara, D. L. *Biosensors & Bioelectronics* **1995**, *10*, 771-783.

Andrade, J. D. *Protein Adsorption*; Plenum Press: New York, 1985; Vol. 2.

Andrade, J. D.; Hlady, V.; Wei, A. P. *Pure and Appl Chem.* **1992**, *4*, 1777-1781.

Armstrong, S. H.; Budka, J. J. E.; Morrison, K. C.; Hasson, M. *J. Am. Chem. Soc.* **1947**, 1747.

Atkins, P. W. *Physical Chemistry*; 4th ed.; W. H. Freeman and Company: New York, 1990.

Bain, C. D.; Whitesides, G. M. *Science* **1988**, *240*, 62-63.

Bain, C. D.; Whitesides, G. M. *J. Am. Chem. Soc* **1989**, *111*, 7164-7175.

Bain, C. D.; Troughton, E. B.; Tao, Y.-T.; Evall, J.; Whitesides, G. M.; Nuzzo, R. G. *J. Am. Chem. Soc* **1989**, *111*, 321-335.

Bain, C. D.; Evall, J.; Whitesides, G. M. *J. Am. Chem. Soc.* **1989**, *111*, 7155-7164.

Baker, S. F.; Othman, R.; Wilton, D. C. *Biochemistry* **1998**, *37*, 13203-13211.

Barker, A. S. *Physical Review B* **1973**, *8*, 5418-5426.

Bayer, E. A.; Rivnay, B.; Skutelsky, E. *Biochimica et Biophysica Acta* **1979**, *550*, 464-473.

Berger, C. E. H.; Beumer, T. A. M.; Kooyman, R. P. H.; Greve, J. *Anal. Chem.* **1998**, *70*, 703-706.

Biebuyck, H. A.; Whitesides, G. M. *Langmuir* **1993**, *9*, 1766-1770.

Biebuyck, H. A.; Bain, C. D.; Whitsides, G. M. *Langmuir* **1994**, *10*, 1825-1831.

Bond, W. N.; Puls, H. O. *Phil. Mag.* **1937**, *7*, 864-888.

Buck, M.; Eisert, F.; Fischer, J.; Grunze, M.; Trager, F. *Appl. Phys. A* **1991**, *552*, 556.

Bunjes, N.; Schmidt, E. K.; Jonczyk, A.; Rippmann, F.; Beyer, D.; Ringsdorf, H.; Graber, P.; Knoll, W.; Naumann, R. *Langmuir* **1997**, *13*, 6188-6194.

Buser, C. A.; Sigal, C. T.; Resh, M. D.; McLaughlin, S. *Biochemistry* **1994**, *33*, 13093-13101.

Chilkoti, A.; Tan, P. H.; Stayton, P. S. *Proc. Natl. Acad. Sci. USA.* **1995**, *92*, 1754-8.

Chilkoti, A.; Boland, T.; Ratner, B. D.; Stayton, P. S. *Biophys. J.* **1995**, *69*, 2125-30.

Chiruvolu, S.; Walker, S.; Israelachvili, J.; Schmitt, F.-J.; Leckband, D.; Zasadzinski, J. A. *Science* **1994**, *264*, 1753-1756.

Cho, W.; Gelb, M. H. *Unpublished observations* .

Corsel, J. W.; Willems, G. M.; Kop, J. M. M.; Cuypers, P. A.; Hermens, W. T. *Journal of Colloid and Interface Science* **1986**, *111*, 544-554.

Dahmen-Levison, U.; Brezesinski, G.; Mohwald, H. *Thin Solid Films* **1998**, *327-329*, 616-620.

Dannenberger, O.; Weiss, K.; Himmel, H. J.; Jager, B.; Buck, M.; Woll, C. *Thin Solid Films* **1997**, *307*, 183-191.

Dannenberger, O.; Buck, M.; Grunze, M. *J. Phys. Chem. B* **1999**, *103*, 2202-2213.

Darnell, J. E.; Lodish, H.; Baltimore, D. *Molecular Cell Biology*; Scientific American Books: New York, 1990.

Darst, S. A.; Ahlers, M.; Meller, P. H.; Kubalek, E. W.; Blankenburg, R.; Ribi, H. O.; Ringsdorf, H.; Kornberg, R. D. *Biophys. J.* **1991**, *59*, 387-96.

DeBono, R. F.; Loucks, G. D.; Mann, D. D.; Krull, U. J. *Can. J. Chem.* **1996**, *74*, 677-688.

Delamarche, E.; Sundarababu, G.; Biebuyck, H.; Michel, B.; Gerber, C.; Sigrist, H.; Wolf, H.; Ringsdorf, H.; Xanthopoulos, N.; Mathieu, M. J. *Langmuir* **1996**, *12*, 1997-2006.

Demel, R. A.; Kessel, W. S. M. G. V.; Zwaal, R. F. A.; Roelofsen, B.; Deenen, L. L. M. V. *Biochimica et Biophysica Acta* **1975**, *406*, 97-107.

Dixon-Warren, S. J.; Bondzie, V.; Burson, N.; Lucchesi, L.; Yu, Y.; Zhang, L. *J. Vac. Sci. Technol. A* **1999**, *17*, 2982-2986.

Doty, P.; Geiduschek, E. P. *The Proteins*; Academic Press: New York, 1953.

Dubois, L. H.; Nuzzo, R. G. *Annu. Rev. PHys. Chem* **1992**, *43*, 437-63.

Ehler, T. T.; Malmberg, N.; Noe, L. J. *J. Phys. Chem. B.* **1997**, *101*, 1268-1272.

Engelsen, D. D. *Surface Science* **1976**, *56*, 272-280.

Fair, B. D.; Jamieson, A. M. *J. of Coll. and Int. Sci* **1980**, *77*, 525-534.

Feldberg, S. W. *Digital Simulation: A General Method for Solving Electrochemical Diffusion-Kinetic Problems*; Bard, A. J., Ed.; Marcel Dekker Inc.: New York, 1969; Vol. 3, pp 199-296.

Florin, E. L.; Gaub, H. E. *Biophysical Journal* **1993**, *64*, 375-383.

Folkers, J. P.; Laibinis, P. E.; Whitesides, G. M. *Langmuir* **1992**, *8*, 1330-1341.

Foreman-Wykert, A. K.; Weinrauch, Y.; Elsbach, P.; Weiss, J. *J Clin Invest* **1999**, *103*, 715-21.

Frey, B. L.; Claire, E. J.; Kornguth, S.; Corn, R. M. *Anal. Chem* **1995**, *67*, 4452-4457.

Gardner, W. C. *Rate and Mechanisms of Chemical Reactions*; W. A. Benjamin Inc.: Menlo Park, CA, 1972.

Glaser, R. W. *Analytical Biochemistry* **1993**, *213*, 152-161.

Golander, C. G.; Kiss, E. J. *J. Coll. Int. Sci* **1988**, *121*, 240.

Grabar, K. C.; Smith, P. C.; Musick, M. D.; Davis, J. A.; Walter, D. G.; Jackson, M. A.; Guthrie, A. P.; Natan, M. J. *J. Am. Chem. Soc.* **1996**, *118*, 1148-1153.

Green, N. M. *Adv. Protein Chem.* **1975**, *29*, 85-133.

Hall, D. R.; Cann, J. R.; Winzor, D. J. *Analytical Biochemistry* **1996**, *235*, 175-184.

Han, S. K.; Yoon, E. T.; Scott, D. L.; Sigler, P. B.; Cho, W. *J. Biol. Chem.* **1997**, *272*, 3573-3582.

Harder, P.; Grunze, M.; Dahint, R.; Whitesides, G. M.; Laibinis, P. E. *J. Phys. Chem. B* **1998**, *102*, 426-436.

Haussling, L.; Ringsdorf, H. *Langmuir* **1991**, *7*, 1837-1840.

Heavens, O. S. *Optical Properties of Thin Films* New York, 1955.

Hegner, M.; Dreier, M.; Wagner, P.; Semenza, G.; Güntherodt, H. J. *J. Vac. Sci. Technol. B* **1996**, *14*, 1418-1421.

Heister, K.; Allara, D. L.; Bahnck, K.; Frey, S.; Zharnikov, M.; Grunze, M. *Langmuir* **1999**, *15*, 5440-5445.

Hendrickson, S. H.; Hendrickson, E. K.; Dybvig, R. H. *J. Lipid. Res.* **1983**, *24*, 1532.

Heyse, S.; Ernst, O. P.; Dienes, Z.; Hofmann, K. P.; Vogel, H. *Biochemistry* **1998**, *37*, 507-522.

Hixon, M. S.; Ball, A.; Gelb, M. H. *Biochemistry* **1998**, *37*, 8516.

Homola, J.; Lu, H. B.; Yee, S. S. *Electronics Letters* **1999**, *35*, 1105-1106.

Huang, W.-T.; Levitt, D. G. *Biophysical Journal* **1977**, *17*, 111-128.

Innes, R. A.; Sambles, J. R. *J. Phys. F: Met Phys* **1987**, *17*, 277-287.

Jain, M. K.; Egmond, M. R.; Verheij, H. M.; Apitz-Castro, R.; Dijkman, R.; Haas, G. H. D. *Biochim. Biophys. Acta* **1982**, *688*, 341-348.

Johnston, K. S.; Karlsen, S. R.; Jung, C. C.; Yee, S. S. *material Chemistry and Physics* **1995**, *42*, 242-246.

Jones, M. L.; Kurzband, G. P. *Biochemistry* **1995**, *34*, 11750-11756.

Jordan, C. E.; Frey, B. L.; Kornguth, S.; Corn, R. M. *Langmuir* **1994**, 3642-3648.

Jordan, C. E.; Corn, R. M. *Anal. Chem.* **1997**, *69*, 1449-1456.

Jordan, C. E.; Frutos, A. G.; Thiel, A. J.; Corn, R. M. *Anal. Chem.* **1997**, *69*, 4939-4947.

Jorgenson, R. C.; Yee, S. S. *Sens. Actuators B* **1993**, *12*, 213.

Jung, C. C. *Surface Plasmon Resonace Light Modulators Using Electro-optic Polymers*; University of Washington, 1994.

Jung, L. S.; Campbell, C. T.; Chinowsky, T. M.; Mar, M. N.; Yee, S. S. *Langmuir* **1998**, *14*, 5636-5648.

Jung, L. S.; Nelson, K. E.; Campbell, C. T.; Stayton, P. S.; Yee, S. S.; Perez-Luna, V.; Lopez, G. P. *Sensors and Actuators B* **1999**, *54*, 137-144.

Karpovich, D. S.; Blanchard, G. J. *Langmuir* **1994**, *20*, 3315-3322.

Kim, H. J.; Kwak, S.; Kim, Y. S.; Seo, B. I.; Kim, E. R.; Lee, H. *Thin Solid Films* **1998**, *327-329*, 191-194.

Klumb, L. A.; Chu, V.; Stayton, P. S. *Biochemistry* **1998**, *37*, 7657-7663.

Knoll, W.; Angermaier, L.; Batz, G.; Fritz, T.; Fujisawa, S.; Furuno, T.; Guder, H. J.; Hara, M.; Liley, M.; Niki, K.; Spinke, J. *Synthetic Metal* **1993**, *61*, 5-11.

Kooyman, R. P. H.; Kiolkman, H.; Gent, J. V.; Greve, J. *Analytica Chimica Acta* **1988**, 35-45.

Kunz, U.; Katerkamp, A.; Renneberg, R.; Spener, F.; Cammann, K. *Sensors and Actuators B* **1996**, *32*, 149-155.

Kurosawa, K.; Pierce, R. M.; Ushioda, S.; Hemminger, J. C. *Phys. Rev. B* **1986**, *33*, 789.

Lang, H.; Duschl, C.; Vogel, H. *Langmuir* **1994**, *10*, 197-210.

Lange, C.; Koch, K.-W. *Biochemistry* **1997**, *36*, 12019-12026.

Lavrich, D. J.; Wetterer, S. M.; Bernasek, S. L.; Scoles, G. *J. Phys. Chem. B* **1998**, *102*, 3456-3465.

Leslie, T. E.; Lilley, T. H. *Biopolymers* **1985**, *24*, 695.

Levich, V. G. *Physicochemical Hydrodynamics*; Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1962.

Levine, I. N. *Physical Chemistry*; McGraw-Hill: New York, 1988.

Li, Y.; Huang, L.; Jr., R. T. M.; Hemminger, J. C. *J. Am. Chem. Soc.* **1992**, *114*, 2428-2432.

Lide, D. R. *Handbook of Chemistry and Physics*; 71st ed.; CRC Press: Boston, 1990.

Liedberg, B.; Nylander, C.; Lundstrom, I. *Sensors and Actuators* **1983**, *4*, 299-304.

Liedberg, B.; Nylander, C.; Lundstrom, I. *Biosensors & Bioelectronics* **1995**, *10*, i-ix.

Lingler, S.; Rubinstein, I.; Knoll, W.; Offenhausser, A. *Langmuir* **1997**, *13*, 7085-7091.

Liu, X.; Zhu, H.; Huang, B.; Rogers, J.; Yu, B. Z.; Kumar, A.; Jain, M. K.; Sundaralingam, M.; Tsai, M. D. *Biochemistry* **1995**, *34*, 7322-34.

Lofas, S.; Malmqvist, M.; Ronnberg, I.; Stenberg, E.; Liedberg, B.; Lunstrom, I. *Sensors and Actuators B* **1991**, *5*, 79-84.

Lukosz, W. *Biosensors & Bioelectronics* **1991**, *6*, 215-225.

Lukosz, W. *Biosensors & Bioelectronics* **1997**, *12*, 175-184.

Lund-Katz, S.; Laboda, H. M.; McLean, L. R.; Phillips, M. C. *Biochemistry* **1988**, *27*, 3416-3423.

Lundstrom, I. *Biosens. Bioelectron.* **1994**, *9*, 725.

MacDonald, R. C.; MacDonald, R. I.; Menco, B. P. M.; Rakeshita, K.; Subbarao, N. K.; Hu, L. *Biochimica et Biophysica Acta* **1991**, *1061*, 297-303.

MacKenzie, C. R.; Hirama, T.; Lee, K. K.; Altman, E.; Young, N. M. *Journal of Biological Chemistry* **1997**, *9*, 5533-5538.

Masson, L.; Mazza, A.; Brousseau, R.; Tabashnik, B. *Journal of Biological Chemistry* **1995**, *270*, 11887-11896.

Matsuura, K.; Ebara, Y.; Okahata, Y. *Thin Solid Films* **1996**, *273*, 61-65.

McMeekin, T. L.; Groves, M. L.; Hipp, N. J. *Refractive Indicies of Amino Acids, Proteins and Related Substances*; 44 ed., 1964, pp 54-66.

Melendez, J.; Carr, R.; Bartholomew, D.; Taneja, H. *Development of a Surface Plasmon Resonance Sensor for Commerical Applications*: Washington, D. C., 1996.

Melendez, J.; Carr, R.; Bartholomew, D. U.; Kukanskis, K.; Elkind, J.; Yee, S.; Furlong, C.; Woodbury, R. *Sens. Acutators, B* **1996**, *35*, 212.

Meuse, C. W.; Niaura, G.; Lewis, M. L.; Plant, A. L. *Langmiur* **1998**, *15*, 1604-1611.

Meuse, C. W.; Krueger, S.; Majkrzak, C. F.; Dura, J. A.; Fu, J.; Connor, J. T.; Plant, A. L. *Biophysical Journal* **1998**, *74*, 1388-1398.

Motschmann, H.; Stamm, M. *Macromolecules* **1991**, *24*, 3681-3688.

Mrksich, M.; Sigal, G. B.; Whitesides, G. M. *Langmuir* **1995**, *11*, 4383-4385.

Mrksich, M.; Grunwell, J. R.; Whitesides, G. M. *J. Am. Chem. Soc* **1995**, *117*, 12009-12010.

Mrksich, M.; Whitesides, G. M. *Annu. Rev. Biophys. Biomol. Struct* **1996**, *25*, 55-78.

Muller, W.; Ringsdorf, H.; Rump, E.; Wildburg, G.; Zhang, X.; Angermaier, L.; Knoll, W.; Liley, M.; Spinke, J. *Science* **1993**, *262*, 1706-8.

Nalefski, E. A.; Sultzman, L. A.; Martin, D. M.; Kriz, R. W.; Towler, P. S.; Knopf, J. L.; Clark, J. D. *J. Biol. Chem.* **1994**, *269*, 18239-18249.

Nelson, K. E.; Jung, L. S.; Gamble, L.; Boeckl, M.; Naeemi, E.; Campbell, C. T.; Castner, D. G.; Stayton., P.S., *Langmuir (submitted)* .

Nir, S.; Stein, W. D. *J. Chem. Phys.* **1971**, *55*, 1598-1603.

Nuzzo, R. G. *J. Am. Chem. Soc.* **1987**, *109*, 2358-2368.

Nuzzo, R. G.; Zegarski, B. R.; Dubois, L. H. *J. Am. Chem. Soc.* **1987**, *109*, 733-740.

O'Brien, M. J.; Brueck, S. R. J.; Perez-Luna, V. H.; Tender, L. M.; Lopez, G. P. *Biosensors and Bioelectronics* **1999**, *14*, 145-154.

O'Shannessy, D. J. *Curr. Opin. Biotechnol.* **1994**, *6*, 65.

Offord, D. A.; Griffin, J. H. *Langmuir* **1993**, *9*, 3015-3025.

Palik, E. D. *Handbook of Optical Constants of Solids*; Academic Press: Orlando, FL, 1985.

Pan, W.; Durning, C. J.; Turro, N. J. *Langmuir* **1996**, *12*, 4469-4473.

Park, K.; Simmons, S. R.; Albrecht, R. M. *Scanning Microscopy* **1987**, *1*, 339-350.

Park, K.; Park, H.; Albrecht, R. M. *Quantitative Characterization of Colloidal Gold Statining*; Hyatt, M. A., Ed.; Academic Press: San Diego, 1989; Vol. Chapter 28.

Pashkov, V. N.; Tsurupa, G. P.; Griko, N. B.; Skopinskaya, S. N.; Yarkov, S. P. *Analytical Biochemistry* **1992**, *207*, 341-347.

Perez-Luna, V. H., O'Brien, M. J., Opperman, K. A., Hampton, P. D., Stayton, P. S., Klumb, L, Lopez, G. P. *J. Am. Chem. Soc* **1999**, *121*, 6469-6478.

Persson, B. *Analytical Biochemistry* **1997**, *246*, 34-44.

Pertsin, A. J.; Grunze, M.; Garbuzova, I. A. *J. Phys. Chem. B.* **1998**, *102*, 4918-4926.

Peterlinz, K. A.; Georgiadis, R. *Langmuir* **1996**, *12*, 4731-4740.

Peterlinz, K. A.; Georgiadis, R. *J. Phys. Chem. B.* **1997**, *191*, 8041-8042.

Plant, A. L. *Langmuir* **1993**, *9*, 2764-2767.

Plant, A. L.; Gueguetchkeri, M.; Yap, W. *Biophysical Journal* **1994**, *67*, 1126-1133.

Plant, A. L.; Brigham-Burke, M.; Petrella, E. C.; O'Shannessy, D. J. *Analytical Biochemistry* **1995**, *226*, 342.

Plant, A. L. *Langmuir* **1999**, *15*, 5128-5135.

Poirier, G. E.; Pylant, E. D. *Science* **1996**, *272*, 1145-1148.

Poirier, G. E. *Langmuir* **1999**, *15*, 1167-1175.

Prater, K. B.; Bard, A. J. *J. Electrochem. Soc.* **1970**, *117*, 209-213.

Prime, K.; Whiteside, G. M. *J. Am. Chem. Soc.* **1993**, *115*, 10714-10721.

Rahn, J. R.; Hallock, R. B. *Langmuir* **1995**, *11*, 650-654.

Ramirez, F.; Jain, M. K. *Proteins: Structure, Function, and Genetics* **1991**, *9*, 229-239.

Ramsden, J. J.; Li, S. Y.; Prenosil, J. E.; Heinzle, E. *Biotechnology and Bioengineering* **1994**, *43*, 939-945.

Reynolds, L. J.; Washburn, W. N.; Deems, R. A.; Dennis, E. A. *Methods in Enzymology* **1991**, *197*, 3.

Rickert, J.; Gopel, W.; Beck, W.; Jung, G.; Heiduschka, P. *Biosensors & Bioelectronics* **1996**, *11*, 757-768.

Ruzic, I.; Feldberg, S. *Electroanalytical Chemistry and Interfacial Electrochemistry* **1974**, *50*.

Sambles, J. R.; Bradbery, G. W.; Yang, F. *Contemp. Phys* **1991**, *32*, 173.

Savitzky, A.; Golay, M. J. E. *Analytical Chemistry* **1964**, *36*, 1627-1639.

Schessler, H. M.; Karpovich, D. S.; Blanchard, G. J. *J. Am Chem. Soc.* **1996**, *118*, 9645-9651.

Schleicher, A.; Hofmann, K.-P. *J. Membrane Biol.* **1987**, *95*, 271-281.

Schlenoff, J. B.; Li, M.; Ly, H. *J. Am Chem. Soc.* **1995**, *117*, 12528-12536.

Schonenberger, C.; Sondag-Huethorst, J. A. M.; Jorritsma, J.; Fokkink, L. G. J. *Langmuir* **1994**, *10*, 611-614.

Schuck, P. *biophysical Journal* **1996**, *70*, 1230-1249.

Schuck, P.; Millar, D. B.; Kortt, A. A. *Analytical Biochemistry* **1998**, *265*, 79-91.

Schurr, J. M. *Biophysical Journal* **1970**, *10*, 700-716.

Sigal, G. B.; Barndad, C.; Barberis, A.; Strominger, J.; Whitesides, G. M. *Anal. Chem.* **1996**, *68*, 490-497.

Silin, V. I.; Blachytis, G. A. *Optics Communications* **1993**, *97*, 19-24.

Silin, V.; Weetall, H.; Vanderah, D. J. *Journal of Colloid and Interface Science* **1997**, *185*, 94-103.

Sjolander, S.; Urbaniczky, C. *Anal. chem.* **1991**, *63*, 2338-2345.

Small, D. M. *The Physical Chemistry of Lipids: From Alkanes to Phospholipids*; Plenum Press: New York, 1986; Vol. 4.

Snitko, Y.; Koduri, R. S.; Han, S. K.; Othman, R.; Baker, S. F.; Molini, B. J.; Wilton, D. C.; Gelb, M. H.; Cho, W. H. *Biochemistry* **1997**, *36*, 14325-14333.

Spinke, J.; Liley, M.; Schmitt, F. J.; Guder, H. J.; Angermaier, L.; Knoll, W. *J. Chem. Phys* **1993**, *99*, 7012-7019.

Spinke, J.; Liley, M.; Guder, H. J.; Angermaier, L.; Knoll, W. *Langmuir* **1993**, *9*, 1821-1825.

Stachowiak, O.; Dolder, M.; Wallimann, T. *Biochemistry* **1996**, *35*, 15522-15528.

Stenberg, E.; Persson, B.; Roos, H.; Urbaniczky, C. *Journal of Colloid and Interface Science* **1990**, *142*, 513-526.

Sundgren, J. E.; Bodo, P.; ivarsson, B.; Lundstrom, I. *J. Coll. Int. Sci.* **1986**, *113*, 530.

Swalen, J. D. *Annu. Rev. Mater. Sci* **1991**, *21*, 373-408.

Tengvall, P.; Lestelius, M.; Leidberg, B.; Lundstrom, I. *Langmuir* **1992**, *8*, 1236-1238.

Terrettaz, S.; Stora, T.; Duschl, C.; Vogel, H. *Langmuir* **1993**, *9*, 1361-1369.

Thomas, R. C., (personal communication..

Thomas, R. C.; Yang, H. C.; DiRubio, C. R.; Ricco, A. J.; Crooks, R. M. *Langmuir* **1996**, *12*, 2239-2246.

Tortorella, D.; Ulbrandt, N. D.; London, E. *Biochemistry* **1993**, *32*, 9181-9188.

Ulman, A. *An Introduction to Ultrathin Organic Films From Langmuir-Blogestt to Self-Assembly*; Academic Press: San Diego, 1991.

Ulman, A. *Chem. Rev.* **1996**, *96*, 1533-1554.

Uvdal, K.; Bodo, P.; Liedber, B. *Journal of Colloid and Interface Science* **1991**, *149*, 162-173.

Verger, R.; Rietsch, J.; Dam-Mieras, M. C. E. v.; Haas, G. H. D. *J. Biol. Chem.* **1976**, *251*, 3128-3133.

Verheij, H. M.; Slotboom, A. J.; De Haas, G. H. *Rev. Physiol. Biochem. Pharmacol.* **1981**, *91*, 91-203.

Walczak, M. M.; Alves, C. A.; Lamp, B. D.; Porter, M. D. *J. of Electroanalytical Chemistry* **1995**, *396*, 103-114.

Wang, R. L. C.; Kreuzer, H. J.; Grunze, M. *J. Phys. Chem. B* **1997**, *101*, 9767-9773.

Ward, A. F. H.; Tordai, L. *J. Chem. Phys.* **1946**, *14*, 453-461.

Wilkening, G.; Linke, T.; Sandhoff, K. *Journal of Biological Chemistry* **1998**, *273*, 30271-30278.

Xu, S.; Cruchon-Dupeyrat, S.; Garno, J. C.; Liu, G.-Y.; Jennings, G. K.; Yong, T.-H.; Laibinis, P. E. *J. Chem. Phys.* **1998**, *108*, 1-11.

Yan, L.; Marzolin, C.; Terfort, A.; Whitesides, G. M. *Langmuir* **1997**, *13*, 6704-6712.

Yang, H. C.; Dermody, D. L.; Xu, C.; Ricco, A. J.; Crooks, R. M. *Langmuir* **1996**, *12*, 726-735.

Yeatman, E. M. *Biosensors & Bioelectronics* **1996**, *11*, 635-649.

Zangwill, A. *Physics at Surfaces*; Cambridge University Press: Cambridge, 1990.

Zhdanov, V. P.; Kasemo, B. *Proteins: Structure, Function, and Genetics* **1998**, *30*, 177-182.

APPENDIX A: THE PLANAR SPR MANUAL

The details about this system have already been presented in the following papers:

- Quantification and Basic System: L.S. Jung, C.T. Campbell, T. Chinowsky, M.N. Mar, S.S. Yee, Langmuir, 14 (1998), 5636-5648

- Flow System: L.S. Jung, K.E. Nelson, C.T. Campbell, P.S. Stayton, S.S. Yee, V. Perez-Luna, G.P. Lopez, Sensors and Actuators B, 54 (1999) 137-144

THE SPR TEN COMMANDMENTS

YOU must follow these ten rules or you will NOT be allowed to use the SPR system.

1. ONLY DI water, buffer or ethanol shall ever touch the syringe pump (no proteins etc.)
2. ONLY DI shall remain in the tubing, syringe or valves, meaning everything must be thoroughly cleaned after use. The manufacturer recommends the syringe be rinsed out at least 10 times. At least once a month the syringe must be cleaned with an acid/base cleaning.
3. The syringe must never be run dry/empty.
4. Do not touch the optics
5. Never alter the flow system or any software with consulting person in charge of the system, as well as informing all other users.
6. Use your own glassware only (or free use glassware above sink at your own risk). This applies to all other supplies in the lab also unless you already have an agreement with other users to share.
7. You must clean up the area and all the glassware you have used after your experiments.
8. Properly dispose of your own waste (use your own waste bottles, which should be labeled with your name and it's contents, put disposable glass items in the broken glass box, etc.)
9. Disk space on the SPR data acquisition is limited, please remove (preferred) or at least compress your data files.
10. All gas cylinders must be completely closed after use, do not rely on the regulators, at least one of them is known to leak.

THE SYSTEM

- 1 planar SPR system (prism, collimator, polarizer, lenses, Al stand, and optical fibers) enclosed in a blue thermal insulated box
- 2 computers (one for the SPR data acquisition and the other to control the syringe pump)
- 1 Syringe pump with power supply
- 2 injection valves with controls (each switch between 2 positions known as red (sample) or green (buffer))
- 1 Oriel White Light Source
- 1 S1000 Ocean Optics Spectrometer
- Lots of tubing (2 tubing loops of ~equal size attached to the valves (one known as red (for samples), one known as green (for buffer) and one attached to the syringe pump (safety loop))

THE SPR SOFTWARE:

Multiplexed SPR Fiber Probe System Software Version 1.3 developed by EBI Sensors Inc. It can not be modified.

THE PUMP SOFTWARE:

This is a Quick Basic program originally supplied by the Ruzicka group to control syringe pumps and VICI valves. The program was completely rewritten to perform commands specific for stop-flow experiments and the valves commands were disabled at this time. It can be easily altered to suit the needs of the user, however the person in charge of the system must be consulted before this can be done.

SETTING UP

SPR

1. Turn on power strip behind the light source, as shown in Figure 1. This turns on the valves (the light source and SPR computer are also plugged into this power strip).
2. Turn on the SPR computer.
3. Turn on the white light source (there is a power switch on the front face) and allow to warm up for 30 - 60 minutes before use.
4. Once the SPR computer give a DOS prompt [:c\ ], start the SPR system software by typing: spr<Return>.
5. The program should start and prompt for the number of channels, enter 1<Return>.

SYRINGE PUMP:

1. Turn on the syringe pump by flipping the switch on the Pump Power Source
2. Turn on the Pump Computer (and monitor, which is usually shut off).
3. Once the Pump computer gives a DOS prompt, start the pump program by typing: linda<Return>. This is a batch file which starts up QuickBasic and opens the file which contains the pump program.
4. Hit F5 to start the program.
5. Type: i [for (I)nitialize] to initialize the pump.

FLUSH OUT SYSTEM (PUMP AND TUBING):

1. Put the larger ID tubing protruding from the right side of the syringe pump into a reservoir containing warm DI water.
2. Be sure you have some receptacle at the end of the "white"(called white due to the color of the fitting) sample inlet tube and at the "blue" tube (the tube which normally attaches to the flow cell, notice there is a large piece of tubing in the box to slip over this tube so that you can extend the length to more conveniently reach some container).
3. Type l [for c(L)ean] on the pump computer to start 10 cycles of filling and emptying.
4. Switch the valves occasionally from green-green (which empties to the "blue" tube) to red-green (which empties to the "white" tube) to flow water through both tubing loops.
5. After flushing with water, flush tubing with desired fluid. If buffer or solvent is required, flush out the syringe and tubing with that liquid (which from now on I will refer to as buffer). (Note: it is preferred that nothing but buffer, ethanol or DI water is to ever enter the syringe, ALSO NOTE: NEVER run the pump if the syringe is empty). If using ethanol, it would be best to flush the syringe a number of times (at

least 10), for buffer which is mainly aqueous a couple flushes per tubing loop is probably sufficient.

6. For best results allow the reservoir of buffer to thermally equilibrate to the system temperature before proceeding with this next step.
7. Fill the syringe with buffer, if there are any bubbles trapped in the syringe, hit the b key [for Remove(B)ubble], which tries to quickly force out the bubbles by forcing 1 ml of liquid out quickly out of the syringe into the reservoir.

OPTICS

1. Clean the prism preferable with Kimwipes or lens paper moistened with DI water. Then wipe with Kimwipes or lens paper moistened with methanol or ethanol until clean and dry.
2. Make sure the polarizer is set at TM and not TE.
3. Try to minimize contact with other optics, if necessary use N_2 to blow dust from optics or **lens paper** (NOT Kimwipes) slightly moistened with methanol to clean.

SAMPLE:

1. Use a small drop of index match (glycerol - fumed silica mixture) directly on the prism.
2. Apply slide to the prism.
3. Place the flow cell (with gasket) on the sample and tighten with the bolts until secure and a good seal is made.

NORMALIZING FOR SPR

1. The SPR program should now be running and a raw spectrum of the reflected light should be on the screen.
2. Adjust the light (using the lever on the light source shutter) so that the highest point on the raw spectrum is somewhere between the second and the third line (from the top) in the spectrum box .
3. Set the averaging to 50 (or other high averaging number) by hitting a (for average) and at the prompt type 50 <Return>.
4. Hit r (for reference). The program will ask if you want to save the reference, hit y.
5. Hit n (to normalize). The screen should now show a flat line at 1. Watch this line, it will become irregular when the light source is not yet warmed up or if the flow cell is "relaxing." Keep referencing and normalizing (repeat steps 4-5) until you have a steady flat line at 1.

EXCITING SPR

1. Now the buffer can now be introduced into the flow cell. Make sure the injection valves are in the "green"-"green" configuration.

2. On the pump computer type d [for (D)ispense].
3. The program will prompt for a flow rate. To minimize bubbles at this step, the slowest rate (1.25 ml/min) is suggested, type 1.25 <Return>
4. The program will then prompt for a final syringe volume. Typically 1 ml is used, so if the syringe is full (at 25 ml), enter 24 <Return>.

Once the flow cell has completely filled the normalized spectrum should show a sharp dip in the reflected spectrum due to SPR.

THE SPR EXPERIMENT

SPR PROGRAM

- Hit t on the SPR computer to go from the normalized spectrum to the trends, which is a plot the position of the wavelength of the SPR minimum over time (minutes).
- Chose an appropriate averaging value by hitting a. The computer will show you the current averaging and prompt for a value. Enter the desired number and hit <Return>. If you are interest in kinetic analysis you may want many data points per second. The following table give a rough idea of how many data points you will get per averaging value. Keep in mind the maximum size of a file is pre-determined by the software, so for low averaging values such as 5, the file will only store ~30 minutes of data before you will be required to save the file (the computer shows a rough estimate of this time in the lower right hand corner of the screen). Therefore, if kinetics is not important, it is recommended you use an averaging of 25 (~1 data point per second) or higher.

POINTS PER SECOND & S/N WITH AVERAGING TABLE

Averaging	Points per second	S/N (roughly)
1	10	0.04
5	4.05	0.012
10	2.2	0.01
20	1.2	0.005
30	0.75	0.005
40	0.509	0.0045
50	0.47	0.004
60	0.37	0.004
70	0.35	0.004
100	.025	0.002

- Other useful commands

  y - adjusts the y-axis (wavelength). The computer will prompt for maximum and minimum wavelengths

  x - adjusts the x-axis (time). The computer will prompt for maximum and minimum time.

  Note: It is best to adjust the axis ahead of time, as any adjustments (any commands) performed during experiments will cause data points skipped.

  d - saves trends file [NOTE: NOT s, which saves the normalized curve only]. The computer prompts for a filename (limit to 8 characters).

  Also note the computer determines a maximum file size, when the file is full it will prompt you to hit <Return> and enter a file name (limit to 8 characters). The data now saved will just be replaced with a straight line, do not be alarmed.

  e - erase trends

  q - quit. The computer will prompt you to be sure you want to quit, type y for yes.

Make sure the pump computer shows "LEFT" in the upper right corner of the screen before proceeding. LEFT indicates the pump will be executing the commands to the left side or to the tubing loops and not the reservoir. To switch between "RIGHT" (reservoir) and "LEFT" (sample loops and flow cell), hit y [for s(Y)ringeValve].

LOADING THE SAMPLE LOOP

1. Make sure the injection valves are in the Red-Green configuration.
2. Place the "white" tube into the sample.
3. On the pump computer type g [for (G)etSample]
4. The computer will inform the user the sample will be drawn at 1.25 ml/min (this is necessary as at higher rates air will be pulled into the syringe due to pressure) and then will prompt for the desired volume. For best results use 2-4 ml samples but for more valuable samples such as protein solutions, 1 ml to 0.3 mls are sufficient. Keep in mind the amount of sample placed in the loop should be greater than the amount of sample that will be injected into the flow cell. (Avoid using the entire "plug" of sample as the ends will tend to mix in the flow cell.)

The following is the suggested protocol for stop-flow experiments:

INJECTING A SAMPLE

1. Make sure the injection valves are in the Red-Red configuration
2. On the pump computer type d [for (D)ispense]
3. The computer will prompt for the desired rate. The suggested rates is 10 ml/min which gives nearly step changes (~1s time constant). Type 10<Return> (for 10 ml/min). If this time constant is not important for the experiment, slower rates are acceptable. The rate selected should be 10 ml/min - 1.25 ml/min.
4. The computer will then prompt for the desired volume, or rather for the final volume of the syringe after injection (Note: the current volume of the syringe is shown in the upper left hand corner of the screen). 1 ml samples are preferred for best results but typically volumes as small as 0.20 ml are injected. Note, concentrations maybe be slightly lower with smaller samples due to some mixing in the flow system. Always inject less sample than

RINSING WITH BUFFER

1. Make sure the injection valves are in the Green-Green configuration
2. On the pump computer type d [for (D)ispense]
3. The computer will prompt for the desired rate. The suggested rates is 10 ml/min. If desired other rates, between 10 ml/min - 1.25 ml/min, are acceptable
4. The computer will then prompt for the desired volume, or rather for the final volume of the syringe after injection. 1 ml samples are typically used.
5. Multiple rinses are suggested as a good test for completeness of rinses.

If injecting more of the sample type of sample (and even the same sample but with different concentrations) just repeat loading, injecting and rinsing as needed. Be sure to keep track of how much "extra" sample may still present in the tubing, to ensure that none of the sample every reaches back to the syringe. The loops are slightly larger than 11 ml.

If injecting a different sample, the sample loop should be cleaned (see directions below) before loading into the sample loop.

CLEANING THE SAMPLE LOOP

1. Make sure the injection valves are in the Red-Green configuration
2. On the pump computer type d [for (D)ispense]
3. The computer will prompt for the desired rate. Type 10<Return> (for 10 ml/min).
4. The computer will then prompt for the desired volume, or rather for the final volume of the syringe after injection. Always dispense more then the volume of sample left in the tubing (0.5+ ml more).

When the syringe is empty, type u [for Fill(U)p] on the pump computer, which instructs the pump to fill up from the reservoir.

Note: Flowing experiments can also be performed. If short times (small volumes) are sufficient the syringe pump can be used, otherwise (and more typically) a continuous pump such as a Miligat pump or a peristaltic pump (though not preferred).

SHUTTING DOWN AND CLEANING UP

- Turn off the light source.
- Quit the SPR software program by hitting q.

CLEANING THE FLOW SYSTEM

1. Put beaker under the white sample inlet tube and another under the tube which enters the flow cell (use extra wide tube to "extend" to tube to reach beaker as described previously in the set up protocol).
2. Using warm DI water as the reservoir type l [for c(L)ean] on the pump computer (which initiates 10 rinsing cycles). Be sure to rinse all tubing by alternating between Red-Green and Green-Green configurations. It is preferred that both configurations get 10 rinses each (for a total of 20 rinse cycles) but at least 5 each is acceptable.

QUIT THE PUMP SOFTWARE

1. Quit the pump software
2. Hit q [for (Q)uit]. The program will prompt to hit any key, hit <Return>. This will return to QuickBasic.
3. Hit <Alt> key which will highlight "File" on the menu bar
4. Use the area to make the File menu appear and scroll down to highlight "Exit"
5. Hit <Return> to Exit QuickBasic.

- Turn off pump computer
- Turn off pump switch

CLEANING THE FLOW CELL

1. Unscrew Flow Cell from system.
2. Carefully clean both the flow cell and gasket with warm soapy water. Care must be taken with the gasket which is easily torn especially if faucet is flowing water rapidly.
3. Rinse well with water and then with DI water.
4. Pat dry, or dry with ethanol and N_2.
5. Wrap in either a Kimwipe or plastic wrap and store in the thermal blue box.

- Turn off SPR Computer
- Turn off power strip behind the light source.
- If applicable double check all gases are shut off

- Be sure to clean up the area of anything liquids and other clutter. There is a glass box located near the gas cylinders by the hood. Only glass should be disposed of there. All waste should properly handled. Organics should be stored in each individual's waste bottle which should be labeled with name and content. Acid/bases should be neutralized (Sodium bicarbonate is available in the hood) before disposal down the sink with copious amount of water. PLEASE do not leave chemicals unlabeled on the bench or hood for an extended period of time, especially if the substance can possibly be hazardous to an unsuspecting worker.

ONCE A MONTH FLOW SYSTEM CLEANING

A cleaning protocol is described in the Cavro manual, this is it slightly altered to include cleaning of the tubing also. The process takes a good amount of time (~3-4 hours). It should be done at least once a month, more often if particularly "sticky" or "exotic" molecules are run through the system.

1) Fill syringe with 0.1 N HCl. Let sit 30 minutes
2) Empty syringe with the valve in the Red-Green position.
3) Fill syringe will HCl again and empty immediately in the Green-Green configuration
4) Rinse Syringe with DI water (still Green-Green)
5) Rinse DI water (Red-Green)
6) Rinse 8 more times alternating between both Green-Green and Red-Green positions (Note the best way to do steps 4-6 is after emptying the acid, place the water reservoir by the tubing and type l [for c(L)ean which initiates 10 cleaning cycles) and change the valve configuration as necessary in between rinses while the syringe is self-filling)
7) Repeat steps 1-6 again this time with 0.1 M NaOHQuantitative

ANALYSIS OF SPR RESULTS

The information presented here has been described in (and mainly taken directly out of) Quantitative Interpretation of the Response of Surface Plasmon Resonance Sensors to Adsorbed Films, Linda S. Jung, Charles T. Campbell, Timothy Chinowsky, Mimi N. Mar Sinclair S. Yee, Langmuir,14,5636-5648, 1998, which should be referenced instead of this manual.

Over a narrow range, the response to changes in bulk index of refraction, in the absence of adsorption from the solution, can be approximated as linear:

$$R = m \cdot \Delta\eta = m \cdot (\eta_{final} - \eta_{initial}). \quad (1a)$$

The magnitude of the local slope, m, in any small range of indices can be thought of as a sensitivity factor for the measurement. With an effective index of refraction for the bilayer, η_{eff}, which would be the properly weighted average of η_a plus η_s, then one could simply estimate the response due to adsorption (i.e., the change in position of the SPR minimum upon adsorption):

$$R = m \cdot (\eta_{eff} - \eta_s), \quad (2a)$$

Since light is being used to probe this index of refraction, it is natural to assume that the proper weighting factor at each point in the bilayer structure should be proportional to the intensity of light that point. The evanescent electromagnetic field decays away into this medium with a characteristic decay length, l_d, of ~25-50% of the wavelength of the light. (The light is typically ~500-900 nm at the SPR minimum.) The intensity of light is the field strength squared, so it decays with height z above the metal surface as $[\exp(-z / l_d)]^2$. Thus, the proper weighting factor in calculating this average refractive index should just be $[\exp(-z / l_d)]^2 = \exp(-2z / l_d)$. This and a linear calibration plot of response versus bulk RI, gives the following equation:

$$R = m \cdot (\eta_{eff} - \eta_s) = m \cdot (\eta_a - \eta_s) \cdot [1 - \exp(-2d / l_d)]. \quad (5a)$$

Note: whenever there is substantial nonlinearity in the SPR response to changes in the bulk refractive index in the range between η_s and η_{eff}, which generally occurs when their difference is large, another method is required for proper quantification. Please refer to the Langmuir paper for more details.

PICKING η_A

The appropriate value for η_a is just the index of refraction of the molecule in pure, condensed form, which can also usually be found or measured. Most proteins, for example, have an index of refraction near 1.6, and this is not expected to be perturbed much upon adsorption since the molecule is so big relative to the fraction of it that would actually form bonds to the surface. If it is a molecule or fragment of a molecule whose refractive index cannot be measured, it can be estimated based on values for molecules with similar structure.

A more complex method of estimating refractive indices are based on the Clausius-Mossotti Equation [42], which for a sample of pure compound j is:

$$(\eta_j^2 - 1) / (\eta_j^2 + 2) = N_{j,0} A_j / (3\varepsilon_0), \quad (9)$$

Here η_j is its RI, ε_0 is the permittivity of vacuum, $N_{j,0}$ is the number density (number of molecules j per unit volume in pure j), and A_j is the frequency-dependent polarizability of the molecule. For a liquid mixture of compound j with k, each with number densities N_j and N_k, one can estimate the index of refraction of the solution, η_{soln}, using the same formula, but replacing η_j with η_{soln}, and $N_{j,0} A_j$ with the sum $N_j A_j + N_k A_k$. If the solution is ideal, then $N_j = f_j N_{j,0}$, where f_j is the volume fraction of j (i.e., the fraction of its volume occupied by j), and likewise for k. Using Eq. (9) to express A_j and A_k in terms of η_j and η_k, respectively, gives the Lorentz-Lorentz equation [43, 44]:

$$(\eta_{soln}^2 - 1) / (\eta_{soln}^2 + 2) = f_j [(\eta_j^2 - 1) / (\eta_j^2 + 2)] + f_k [(\eta_k^2 - 1) / (\eta_k^2 + 2)] \quad (10a)$$

This turns out to give a nearly linear variation in the RI of the solution with the fraction of its volume occupied by k, so that the following is true to within a few percent when ηj and ηk are both within the usual range of SPR applications (from 1.33 to 1.6):

$$\eta_{soln} = f_j \eta_j + f_k \eta_k = f_j (\eta_j - \eta_k) + \eta_k. \quad (10b)$$

Equation (10a) can be applied to determine the RI of molecules by measuring the contribution they make to the RI of a solution. For example, the addition of most proteins to aqueous buffer solution (0.3 M NaCl) causes the RI to increase by 1.8×10^{-4} RIU for every g/L of added protein [41]. (Glycoproteins and lipoproteins have somewhat lower increases [41].) The specific volume of proteins in aqueous buffer is ~0.77 mL / g [45, 46]. Using this in Eq. (10b) gives that $\eta_{protein}$ -

η_{buffer} = 0.234 RIU. Since in those solutions η_{buffer} = 1.336 RIU, we get that $\eta_{protein}$ = 1.57 RIU for the water-free protein, which is greater than the refractive index estimated for "adsorbed protein films" using ellipsometric approaches which assume a single optical thickness, since that the film volume includes a great deal of water. Here, we are instead just referring to that part of such films which are made of protein material itself, not water.

DETERMINING M

The following table is the calibration as measured on this planar SPR system.

SPR Calibration

RI	nm	RI	nm	RI	nm
1.3330	684.53	1.3470	731.13	1.3610	803.25
1.3340	687.68	1.3480	735.06	1.3620	810.22
1.3350	690.82	1.3490	739.14	1.3630	817.48
1.3360	693.95	1.3500	743.36	1.3640	825.06
1.3370	697.08	1.3510	747.75	1.3650	832.97
1.3380	700.23	1.3520	752.32	1.3660	841.21
1.3390	703.40	1.3530	757.08	1.3670	849.80
1.3400	706.62	1.3540	762.03	1.3680	858.75
1.3410	709.88	1.3550	767.19	1.3690	868.08
1.3420	713.20	1.3560	772.58	1.3700	877.78
1.3430	716.60	1.3570	778.20	1.3710	887.89
1.3440	720.07	1.3580	784.07	1.3720	898.40
1.3450	723.65	1.3590	790.19	1.3730	909.32
1.3460	727.33	1.3600	796.58		

The most important information to get from this data is the curve are **not** the absolute values for refractive index, but how the refractive index changes with wavelength. Therefore to determine the proper m value, determine the shift in SPR wavelength. Then plot the points from the starting wavelength and ~half-way to the observed full shift. The slope of these point is the m value.

DETERMINING L_D

The decay length, l_d, is a key parameter in these calculations. The probe depth of the technique is one-half this decay length. A rough but reasonable estimate is that l_d equals 37±13% of the light wavelength at the SPR minimum, λ , which for a typical minimum of 680 nm in aqueous solutions gives a probe depth $l_d / 2 = \sim 120$ nm.

A more accurate estimate of l_d comes from the Maxwell's equations:

$$l_d = (\lambda/2\pi) / \mathrm{Re}\{[\eta\mathrm{eff}^2 \varepsilon_{metal} / (\eta\mathrm{eff}^2 + \varepsilon_{metal})] - \eta\mathrm{eff}^2\}^{1/2}$$
$$= (\lambda/2\pi) / \mathrm{Re}\{-\eta\mathrm{eff}^4 / (\eta\mathrm{eff}^2 + \varepsilon_{metal})\}^{1/2}, \quad (11)$$

where ηeff is the effective index of refraction of the probed medium and ε_{metal} is the complex dielectric constant of the metal at that wavelength (see below).

DETERMINING THE DIELECTRIC FOR AU

The follow table contains literature values for Au dielectric as a function of wavelength and was taken from J Phys F: Met Phys, 1987, 17, 277-287.

Au Dielectric as a Function of Wavelength

Wavelength(nm)	Real component	Imaginary Component
687.00	-15.970	1.3750
700.00	-16.780	1.3170
712.00	-17.670	1.4350
725.00	-18.440	1.3370
737.00	-19.700	1.5100
750.00	-20.600	1.5700
762.00	-21.300	1.5000
775.00	-22.600	1.7000
787.00	-23.500	1.6000
800.00	-24.800	2.0000

The real component appears to be linear with wavelength and therefore it's slope is used to extract the dielectric constant for the desired wavelength.

y (real component) = 37.703 - (0.077792*(wavelength))

The imaginary part is not linear and is determined from plotting the imaginary component with respect to wavelength and following the trace to the desired wavelength

DETERMINING THICKNESS

When d is very small compared to l_d (the linear response regime) the following equation applies:

$$d = (l_d / 2) (R / R_{max}) = (l_d / 2) \{R / [(m \cdot (\eta_a - \eta_s)]\} \quad (7)$$

so that the response is directly proportional to the thickness of the adlayer. For cases not in the linear response regime please refer to the Langmuir paper for more details. Remember this is typically not an actual thickness, but a thickness if there was no water/buffer included in that layer (see the Langmuir paper for more details), therefore it makes more sense to convert this thickness to a coverage (see below).

The absolute accuracy will only be as good as this estimate of l_d, which is only ±35% in the absence of some calibration or better means of estimating it. Inaccuracy can also arise if the value of m or η_a used is inaccurate. At small thicknesses, the relative error in m or $(\eta_a - \eta_s)$ directly appears as the same relative error in the thickness. The relative error due to these grows rapidly as the true thickness increases to values near $l_d / 2$. For example, the error in thickness is 35% for a 20% error at a thickness of 0.35 l_d.

DETERMINING COVERAGE

Once d, is estimated by the above procedure, it is a trivial matter to convert this to surface concentration, θ, in molecules per cm^2. The conversion factor is just the bulk number density of the adsorbate, N, in units of molecules per cm^3:

$$\theta \text{ (in molecules per cm}^2\text{)} = d \text{ (in cm)} \cdot N \text{ (in molecules per cm}^3\text{)}. \quad (8)$$

The bulk number density of the adsorbate, N, can be estimated from the bulk density of the adsorbate, ρ, in units of grams per cm^3 just by dividing by the molecular weight and multiplying by Avogadro's number. The proper value of ρ to use is the value for pure, condensed bulk adsorbate, which is the same material whose index of refraction η_a was used to get d. If the adsorbate is a molecular fragment, its density can be estimated from that of similar molecules. Typical ρ values for proteins are 1.3 g / mL, for DNA 1.6-1.7 g / mL, for lipids1.09 g / mL, and for thiols 0.84g / mL.

MAKING GASKETS FOR FLOW CELL

- Using the Dow-Corning Silastic E RTV Silicone Rubber kit, mix the Silicone rubber and the curing agent as directed (1:10 ratio (by weight) respectively)
- Spread into Plexiglas gasket mold.
- Degas in vacuum until bubbles no longer form (this takes some time).
- Place Plexiglas sheet over and place weights on top to firmly sandwich the silicone and to prevent the top piece from sliding.
- Place in the glassware oven on very low heat (typically overnight).
- Carefully remove gasket from mold.
- Trim extra material around the edges. (Suggestion: Do not trim away all of the thin border around the gasket, it will help hold the gasket in place (prevent shifting due to pressure of flow) while in the flow cell.)
- Cut a channel slightly larger than the size of the light. There are several "cookie-cutter" available for cutting, none of which work very well and none of which are the correct size or optimal shape. Until an optimal cutter is made, these are used to make an initial cut and then a razor is used reshape and enlarge the channel as desired.

IMPORTANT NOTE REGARDING FLOW CELL DIMENSIONS:

- The flow cell dimensions have be altered to better suit the system as well as improve flow of samples to the surface. The most current flow cell should be used as a guide for dimensions.
- The most important change is the location of the inlet holes. The light spot is not exactly centered as expected, it is actually lower and more toward the right. A Kimwipe moistened with index match on the prism will show the location of the light, but remember once the slide is put on the prism, the light spot will move a bit more to the right.
- The height of the flow cell: the key is to make the cross sectional area of the flow channel to be as close to the cross sectional area of the tubing as possible.
- The diameter of the centered hole (inlet and outlet) should match the tubing.

APPENDIX B: SOFTWARE FOR SPR SYRINGE PUMP

This program was written in Basic using QuickBasic (Microsoft). An original version was supplied by Prof. J. Ruzicka. This version was completely rewritten to make the program easier to use (restructure, modularize, additional functionality, error checking, simplier user interface, etc.) and to customize it for SPR stop flow experiments.

```
' ********************************************************************
' ** QBASIC PROGRAM Originally written by David Holman,
' ** Ruzicka/Christian group, University of Washington version 4/11/97
' ** PURPOSE:  Provides control for pumps, valves and switches.
' **
' ** Completely Rewritten 8/97 by Linda S. Jung:  Campbell group
' ** Removing data handling and other specific aspects of the program
' ** to create a program to be used mainly for the operation of
' ** one syringe pump SPR experiments with the goal of automating typical functions.
' ********************** CONSTANTS ********************************
CONST LEFT = 1              ' ** Used to indicate left syringe valve position
CONST RIGHT = 0             ' ** Used to indicate right syring valve position
CONST BAD = 0               ' ** Boolean, indicates error
CONST OK = 1                ' ** Boolean, indicates everything is ok
CONST GETSAMPLE = 0         ' ** Constant to indicate "mode" of filling
CONST NORMAL = 1            ' ** Constant to indicate "mode" of filling
CONST NOPUMP = -1           ' ** Flag to indicate there is no comm with pump
CONST PT = 46               ' ** ASCII value of .
CONST ZERO = 48             ' ** ASCII value for zero
CONST NINE = 57             ' ** ASCII value for nine
CONST RET = 13              ' ** ASCII value for return character
CONST ESC = 27              ' ** ASCII value for esc key
CONST BACKSPC = 8           ' ** ASCII value for backspace
CONST SPACE = 32            ' ** ASCII value for space
CONST NULL = 3              ' ** ASCII value for NULL character
' ****************** CONSTANTS FOR MENUS AND PROMPTS ************
CONST MROW = 22             ' ** Typical output line (row) location
CONST MROW2 = 23            ' ** Another typical output line (row) location
CONST MCOL = 3              ' ** Typical output column location
CONST INFOROW = 1           ' ** Row where valve & volume info are printed
CONST SVALVEINFO = 1        ' ** Column where syringe valve info is
CONST SVOLINFO = 20         ' ** Column where syringe volume is
CONST CVOLINFO = 45         ' ** Column where current syring volume is
```

```
CONST EMPTYING$ = "Final Syringe Volume? " ' ** Prompt for Emptying syringe
CONST FILLING$ = "How Many mLs?"    ' ** Prompt for Filling syringe
CONST FLOWRTE$ = "Flow Rate (mL/min)? "    ' ** Prompt for Getting Flow Rate
CONST BADPUMP$ = "There is Something Wrong with the Pump"
' ************** CONSTANTS FOR MESSAGES (sub Notice) *************
CONST NOTICEROW = 12                    ' ** "Notice" message row
' ************************** CONSTANTS FOR FillUp ******************
CONST FILLSPD = 25      ' ** Rate used in FillUp, do not exceed max/min (see
below)
CONST FILLSAMP = 1.25   ' ** Rate used for filling sample loop
' ************************* CONSTANTS FOR Clean ******************
CONST CLEANSPD = 10     ' ** Rate used to flush tubing (Clean)
CONST CYCLENUMS = 10                    ' ** Number of flushes to do when cleaning
CONST BUBSPD = 25       ' ** Rate to eliminated bubble from syringe
CONST MINVOL = 1        ' ** Volume to use to get rid of bubble in syringe
' ************************* CONSTANTS FOR VARIOUS PUMP COMMANDS
*******
CONST POSITION = 2      ' ** Used to indicate interested in position
CONST READY = 3         ' ** Used to indicate interested in readiness
CONST CEASE = 4         ' ** Used to indicate the pump should stop
CONST PUMPIT = 5        ' ** Used to inidicate the should pump
CONST THISRATE = 6      ' ** To set pump rate
CONST INIT = 7          ' ** Use to indicate initialize pump
' ************************* CONSTANTS dealing with devices ************
' ** Cavro pumps (serial communication devices).
' ** Set the following addresses to "-1" if the device is not in use:
' ** The PUMPADDR$ are the SW1 setting + 1.
' ** There is a switch, SW1, on each Cavro device.
' ********************************************************************
CONST CLOCKCORRECT = 86400      ' ** Some clock correction factors
CONST PUMPADDR$ = "1"           ' ** Physical pump address
CONST PUMPPORT% = 1
CONST SYRVOL! = 25                      ' ** Volume of syringe
CONST CMD$ = "Z"        ' ** Command string for syringe > 1 ml (PumpInit)
' ** CONST CMD$ = "Z2"   ' ** Command string for syringe < 1 ml but > 0.25 ml
' ** CONST CMD$ = "Z1"   ' ** Command string for syring < 0.25
CONST MAXFLOW! = 3 * SYRVOL!  ' ** Keep the maximum reasonably slow.
CONST MINFLOW! = .05 * SYRVOL!' ** 0.05 ml/min is minimum for a 1 ml syringe.
CONST DIVISIONS = 3000                  ' ** Syringe steps
'******************** FUNCTIONS ***********************************
DECLARE FUNCTION PumpInit% (NumStr$)            ' ** Initalizes pump
DECLARE FUNCTION AnyResp% ()    ' ** Checks if the pump is communcating
DECLARE FUNCTION GetResp$ ()    ' ** Grabs pump response
```

```
DECLARE FUNCTION GetPosResp$ (instr$, InLen%, catch%)
                                        ' ** Extracts Pump Position from response
DECLARE FUNCTION FindStat$ (instr$, InLen%, catch%)
                                ' ** Returns indicator if pump gave proper response
DECLARE FUNCTION GetFlowRate! ()    ' ** Returns desired flow rate from user
DECLARE FUNCTION FlowOk! (Fl!)      ' ** Returns flow rate is it's OK
DECLARE FUNCTION HowMuch! (         ' ** Returns current syring volume
DECLARE FUNCTION GetVolume! (question$, vol!)
                                        ' ** Returns how much vol the user wants
DECLARE FUNCTION Empty% ()          ' ** Empties the syringe
DECLARE FUNCTION Fill% (which%)     ' ** Fills syringe
DECLARE FUNCTION FillUp% (jcount1%)' ** Fills syringe completely
DECLARE FUNCTION SetVol% (vol!)     ' ** Set volume
DECLARE FUNCTION PumpAmt% (vol!) ' ** Pump desired volume at set rate
DECLARE FUNCTION UserInput% (Messag$, instr$)
                                        ' ** Returns user response to prompts
DECLARE FUNCTION Switch% (jcount%)
            ' ** Switches right or left position of syringe, returns position
DECLARE FUNCTION Query% (which%)
                                ' ** Returns indicator of readiness or pump position
' ************** SUBROUTINES ************************************
DECLARE SUB Menu (ch$, indic%)          ' ** Takes user choice of menu
DECLARE SUB PrintMenu (jcount%, indic%)
                                        ' ** Prints menu and other vital info to
                                    screen
DECLARE SUB notice (mess$)              ' ** Prints errors or other notices
DECLARE SUB SendPumpCmd (which%, instr$) ' ** Send commands to pump
DECLARE SUB StopPump (StartPos%)    ' ** Stops pump and prints delivered vol
DECLARE SUB Clean (jcount1%)            ' ** Flushes tubing with water 5x
DECLARE SUB Pause (x)                   ' ** pauses program
DECLARE SUB SetRate (Fl!)               ' ** Set flow rate
'******************** Global Variables ***************************
  COMMON SHARED spd1!
 '**********EXPERIMENTAL
PARAMETERS*******************************
  SCREEN 9
  WIDTH 80, 25
  '** WIDTH 80, 43                     ' ** Another Option
  CLOSE                     'Close any files not opened by this program.
  OPEN "COM1:9600,N,8,1,CS,DS" FOR RANDOM AS #1
  ' *** "CS" disables the clear to send signal or monitoring of it?
  ' *** "DS" disables the data set ready signal or monitoring of it?
  ' *** "RS" suppresses RTS (request to send)
```

```
' ********************** End initialization ************************
 CLS
 indic% = Query%(READY)
 IF indic% > NOPUMP THEN
   SetRate SYRVOL!                        ' ** Set default speed
 END IF
 DO
   VIEW
   Menu ch$, indic%
 LOOP UNTIL ch$ = "Q"
END
'**********************


FUNCTION AnyResp%
' ***************************************************************************
' ** PURPOSE:  Times no response time from pump
' ** RETURNS:  OK if contact is made, BAD if not
' ***************************************************************************
   tim! = TIMER
   DO
    Dlay! = TIMER - tim!
    IF Dlay! > .1 THEN
     LOCATE 10, 10: PRINT ("No message from pump ");
     PRINT USING "#.##"; Dlay!; : PRINT " seconds"
    END IF
   LOOP UNTIL (LOC(1) <> 0) OR (Dlay! > 1)
   IF (LOC(1) = BAD) THEN
     AnyResp = BAD
   ELSE
     AnyResp = OK
   END IF
   LOCATE 10, 10: PRINT ("                          ")
END FUNCTION

SUB Clean (jcount1%)
' ***********************************************************
' ** Purpose:  Flush full volume of syringe through tubing
' **         returning to original syringe valve side
' ***********************************************************
 I% = 0
 flag% = OK
 DO
```

```
  I% = I% + 1
  LOCATE INFOROW, CVOLINFO:  PRINT "Current Cycle:  "; I%; "        "
  notice ("Filling Syringe From Right Valve")
  flag% = FillUp(LEFT)                    ' ** Fill syringe with water **
  IF (flag% = OK) THEN
    SetRate CLEANSPD                      ' ** Set Rate of Flow **
    notice ("Emptying Syringe To the Left Valve")
    flag% = SetVol%(0)                    ' ** Empty Syringe **
    IF flag% = NOPUMP THEN
      flag% = BAD
      notice (BADPUMP$)
    END IF
  END IF
 LOOP UNTIL ((I% = CYCLENUMS) OR (flag% = BAD))
 IF (jcount1% = RIGHT) THEN               ' ** If entered sub in right position
  dum% = Switch%(LEFT)                    ' **  switch back to the right
 END IF
END SUB ' ** Clean **

FUNCTION Empty%
' ********************************************
' ** PURPOSE: Called by Menu intiates the Empyting of syringe
' ** Parameters: None
' ** Returns:  Returns BAD if the pump malfunctions otherwise OK
' ********************************************
flag% = OK
curvol! = HowMuch!                        ' ** Get current volume **
IF (curvol! = NOPUMP) THEN                ' ** If the pump malfunctions **
     flag% = curvol!                      ' ** Then Function returns this **
ELSE
     IF (curvol! = BAD) THEN  ' ** If the syringe is empty report this **
       notice ("The Syringe is already empty!")
     ELSE
       Fl! = GetFlowRate                  ' ** Prompt user for flow rate **
       Fl! = FlowOk(Fl!)                  ' ** Make sure it's reasonable **
       IF (Fl! > BAD) THEN                ' ** If it's ok
          status = GetVolume(EMPTYING$, vol!)  ' ** prompt for vol to dispense **
          IF (vol! >= curvol!) THEN  ' ** if it does not make sense print error message
**
            make$ = ("There are " + STR$(curvol!) + " mLs in the Syringe")
            notice (make$)
          ELSEIF (status = OK) THEN       ' ** If the vol makes sense **
            CALL SetRate(Fl!)             ' ** Set the desired flow rate **
```

```
            dum% = SetVol%(vol!)          ' ** dispense to desired vol mark **
          END IF
        END IF
      END IF
END IF
Empty = flag%                             ' ** Return OK unless pump malfunctioned
END FUNCTION

FUNCTION Fill% (which%)
' ********************************************************
' ** Purpose:  Initiates filling syringe
' ** Parameters:  which% indicates if just filling or
' **              filling from sample loop (Get Sample chosen)
' ** Returns:
' ********************************************************
curvol! = HowMuch!
IF (which% = NORMAL) THEN
     Fl! = GetFlowRate!
     Fl! = FlowOk(Fl!)
ELSE
     Fl! = FILLSAMP
END IF
IF ((Fl! > BAD) OR (which% = GETSAMPLE)) THEN
     status% = GetVolume(FILLING$, vol!)
     IF ((vol! + curvol!) > SYRVOL!) THEN
        make$ = "They Syringe only holds " + STR$(SYRVOL!) + " mLs."
        notice (make$)
     ELSEIF (vol! > BAD) THEN
        CALL SetRate(Fl!)
        dum% = PumpAmt(vol!)
     END IF
END IF
Fill = OK
END FUNCTION

FUNCTION FillUp% (jcount1%)
'
*************************************************************************
*
' ** Purpose:  Fill up the syringe with water (right) and then switch back
' **           to the original syringe valve position.
```

```
'
*************************************************************************
*
 flag% = OK
 vflag% = 0
 IF (jcount1% = LEFT) THEN
   dum% = Switch(LEFT)                    ' ** Syringe valve to the right, water source'
   vflag% = OK
 END IF
 curvol! = HowMuch!
 IF (curvol! > NOPUMP) THEN
  IF (curvol! < SYRVOL!) THEN
    SetRate FILLSPD                       ' ** Set Speed
    tofill! = SYRVOL! - curvol!           ' ** Determine volume left in syringe
    flag% = PumpAmt(tofill!)              ' ** Fill Syringe with water
  ELSE
    notice ("Syringe Already Filled")
  END IF
 ELSE
  notice ("There is a problem with the pump")
  flag% = BAD
 END IF
 IF (vflag% = OK) THEN
   dum% = Switch(RIGHT)        ' ** Switch Syringe valve back if necessary **
 END IF
 FillUp% = flag%
END FUNCTION' ** FillUp **

FUNCTION FindStat$ (instr$, InLen%, catch%)
' ************************************************
' ** PURPOSE:  Find the information of interest
' **         Essentially two characters away from /
' ** RETURNS:  The start of the proper information
' **         (index in array)
' ************************************************
 DO
  I% = I% + 1
  read$ = MID$(instr$, I%, 1)              ' ** Read string character at a time
 LOOP UNTIL ((I% = InLen%) OR (read$ = "/")) ' ** Loop until / or end of string
 IF I% = InLen% THEN
   catch% = BAD                            ' ** If not found, bad
 ELSE
   catch% = I% + 2
```

```
   FindStat$ = MID$(instr$, catch%, 1)      ' ** If found, skip two characaters
  END IF
END FUNCTION

FUNCTION FlowOk! (Fl!)
' ************************************************
' ** Purpose:  Check if flow rate is ok
' ** Parameters:  Fl! = flow rate entered by user
' ** Returns:  Returns flow rate is ok, otherwise
' **          returns BAD
' ************************************************
 flag% = BAD                  ' ** flag is preset to BAD
 IF (Fl! > BAD) THEN          ' ** if the flow sent was not already bad then
 ' ************************* CASE *********************
 SELECT CASE Fl!              ' ** determine is an acceptable speed if not notify user
 ' ************************* Flow is too slow ********
   CASE IS < MINFLOW!
     make$ = "Too Slow, minimum flow rate is " + STR$(MINFLOW!)
 ' ************************* Flow is too fast ********
   CASE IS > MAXFLOW!
     make$ = "Too Fast, maximum flow rate is " + STR$(MAXFLOW!)
 ' ************************* Flow is acceptable ********
   CASE ELSE                   ' ** if the flow is ok, reset flag to OK
     flag% = OK
 END SELECT
 ' ************************* END CASE *********************
 IF flag% = BAD THEN          ' ** If flow was BAD send back BAD
   notice (make$)
   FlowOk! = BAD
 ELSE                         ' ** Else send flow rate back
   FlowOk! = Fl!
 END IF
 END IF
END FUNCTION

FUNCTION GetFlowRate!
'
' ************************************************************************
' ** Purpose:  prompt user for flow rate desired and return entry
' ** Return:   A number reflecting the flow rate or BAD if none is given
' ************************************************************************
 ' ******************* Call to UserInput for input ****************
   status% = UserInput%(FLOWRTE$, instr$)
```

```
  IF (status% = BAD) THEN                ' ** NO input return BAD
     GetFlowRate! = BAD
  ELSE
     GetFlowRate! = VAL(instr$)          ' ** Rate entered return this value
  END IF
END FUNCTION ' ** FlowRate **

FUNCTION GetPosResp$ (instr$, InLen%, catch%)
' ************************************************
' ** Purpose:  Takes string (pump response)
' **         and stores in into a single string
' ** Returns:  Returns that string
' ** Parameters:  instr$, a string
' ************************************************
  DO
    catch% = catch% + 1
    x$ = MID$(instr$, catch%, 1)         ' ** Read a character from string
    IF ASC(x$) <> NULL THEN              ' ** If not NULL store in messg$
     messg$ = messg$ + x$
    END IF
  LOOP UNTIL (I% = InLen%) OR (ASC(x$) = NULL)
  GetPosResp = messg$
END FUNCTION

FUNCTION GetResp$
' ****************************************
' ** Purpose:  Get pump response and store in string
' ** Returns:  the string
' ****************************************
  WHILE LOC(1) > 0
    Pause .01
    x$ = INPUT$(LOC(1), #1)              ' ** Read character
    instr$ = instr$ + x$                 ' ** store in string
    Pause .1
  WEND
  GetResp$ = instr$                      ' ** send resulting string
END FUNCTION

FUNCTION GetVolume (question$, vol!)
'
*************************************************************************
' ** Purpose:  prompt user for volume desired and return entry
' ** Return:   a flag stating if the entry was successful
```

```
'
'*************************************************************************
 status% = UserInput%(question$, instr$) ' ** GET desired postion
 IF (status% = OK) THEN                  ' ** If input was ok **
   vol! = VAL(instr$)                    ' ** Assign volume
 ELSE
   vol! = BAD                                    ' ** Else return vol! as BAD
 END IF
 GetVolume = status%                     ' ** return status of input
END FUNCTION

FUNCTION HowMuch!
'*************************************************************************
' ** Purpose:  Determines the actual volume contained in the pump syringe
' ** Returns:  The volume
' **     steps (vol/DIVISIONS) ss position*vol/DIVISIONS gives the volume
'*************************************************************************
 Pump1Pos% = Query%(POSITION)        ' ** Get pump position
 IF (Pump1Pos% > NOPUMP) THEN
   HowMuch! = Pump1Pos% * SYRVOL! / DIVISIONS  ' ** Determine volume and
return
 ELSE
   HowMuch! = NOPUMP
 END IF
END FUNCTION ' ** HowMuch **

SUB Menu (ch$, indic%)
'*************************************************************************
' ** Purpose:  Response to User input of menu, calls function to
' **         print menu too
' ** Parameters:  ch$ user input
' **           indic% indicates if pump is communicating or not
'*************************************************************************
   STATIC jcount1%
   PrintMenu jcount1%, indic%
   ch$ = UCASE$(INPUT$(1)) ' ** Get user input/menu choice **
   '*************************CASE STATEMENT**********************
   SELECT CASE ch$
    '************************** Remove Bubble From Syringe
   CASE "B"
     flag% = RIGHT
     IF (indic% = OK) THEN
       IF (jcount1% = LEFT) THEN
```

```
    dum% = Switch(LEFT)       ' ** Syringe valve to the right, Solvent source'
    flag% = LEFT
  END IF
  curvol! = HowMuch!
  IF (curvol! < MINVOL) THEN
    notice ("Making sure the syringe holds at least 1 ml")
    CALL SetRate(FILLSPD)
    dum% = PumpAmt(MINVOL)
    curvol! = MINVOL
  END IF
  notice ("Quickly forcing out 1 mL")
  vol! = curvol! - MINVOL
  CALL SetRate(BUBSPD)
  dum% = SetVol(vol!)
  IF (flag% = LEFT) THEN
    dum% = Switch(RIGHT)
  END IF
END IF
' ************************** CLEAR SCREEN
CASE "C"
 CLS
'******************** EMPTY TO DESIRED VOLUME POSITION *****
CASE "D"
  IF (indic% = OK) THEN
    indic% = Empty
  END IF
'******************** Put Desired Volume into Syringe ******
CASE "F"
  IF (indic% = OK) THEN
    indic% = Fill(NORMAL)
  END IF
'************************ Get Sample into Sample Loop *****
CASE "G"
  flag% = LEFT
  IF (indic% = OK) THEN
    notice ("Sample will be obtained 1.25 ml/min from the left")
    IF (jount1% = RIGHT) THEN
      dum% = Switch%(RIGHT)
      flag = RIGHT
    END IF
    indic% = Fill(GETSAMPLE)
    IF flag% = RIGHT THEN
       dum% = Switch%(LEFT)
```

```
      END IF
    END IF
  '**************** Initialize Syringe Pump *************
  CASE "I"
   NumStr$ = "1"
   notice ("Initializing Syringe Pump")
   temp% = PumpInit%(NumStr$)
   IF (temp% > NOPUMP) THEN
     indic% = OK
   ELSE
    indic% = NOPUMP
   END IF
   IF (jcount% = RIGHT) THEN
      jcount% = jcount1%
      jcount1% = (Switch%(jcount%))
   END IF

  '********************** FLUSH OUT TUBING ********************
   CASE "L"
   IF (indic% = OK) THEN
      notice ("Flushing Tubing TEN Times")
      CALL Clean(jcount1%)
      CLS
   END IF
  '************************** STOP CAVRO PUMP ***************
   CASE "S"
   ' ** Not really a menu choice taken care of elsewhere
  '****************** Completely Fill Syringe with water *********
   CASE "U"
   IF (indic% = OK) THEN
     notice ("Filling Syringe From the Right")
     dum% = FillUp(jcount1%)
     CLS
   END IF
  '******************** SWITCH SYRINGE VALVE ****************
   CASE "Y"
   jcount% = jcount1%
   jcount1% = (Switch%(jcount1%))          ' ** Calls function which switches
                                           ' ** and then the new position
  '************
  END SELECT
END SUB ' ** Menu **
```

```
SUB notice (mess$)
' ************************************************************
' * Purpose:  Prints out notices needed in the same location
' * Parameter:  String to be issued
' ************************************************************
 LOCATE NOTICEROW, MCOL: PRINT SPACE$(70)  ' ** Clear Line
 LOCATE NOTICEROW, MCOL
 PRINT mess$                        ' ** Print message
END SUB

REM $DYNAMIC
SUB Pause (x)
' ********************************************************************
' ** This procedure simply causes the program to wait the specified
' ** length of time.
' ********************************************************************
 st = TIMER
 DO
  IF st > TIMER THEN st = st - CLOCKCORRECT ' ** correct for midnight rollover
 LOOP UNTIL TIMER >= (st + x)
END SUB ' ** Pause **

REM $STATIC
SUB PrintMenu (jcount1%, indic%)
' ************************************************************
' ** Purpose:  Print out Menu
' ** Parameters:  jcount% which indicates syringe valve position
' **            indic% indicates whether the pump is communicating
' **                  or not
' ************************************************************
 LOCATE MROW, MCOL: PRINT SPACE$(77)
 LOCATE MROW2, MCOL: PRINT SPACE$(77)
 LOCATE INFOROW, SVOLINFO: PRINT SPACE$(79)
 '***************** PRINT OUT INFO AND MENU ****************
 IF (indic% = NOPUMP) THEN
   LOCATE INFOROW, SVOLINFO: PRINT ("There is a Problem with the Pump    ")
 ELSE
   curvol! = HowMuch!
   IF jcount1% > RIGHT THEN            ' ** Print syringe valve position
    LOCATE INFOROW, SVALVEINFO: PRINT " LEFT"
   ELSE
    LOCATE INFOROW, SVALVEINFO: PRINT "RIGHT"
   END IF
```

```
   LOCATE INFOROW, SVOLINFO:  PRINT "Syringe Vol:  "; SYRVOL!
   LOCATE INFOROW, CVOLINFO:  PRINT "                 "
   LOCATE INFOROW, CVOLINFO:  PRINT "Current Vol:  "; curvol!
  END IF
  ' ******************** Print Menu *********************************
  LOCATE MROW, MCOL: PRINT "(I)nitialize, Remove(B)ubble, s(Y)ringeValve,
Fill(U)pSyringe, (C)learScreen"
  LOCATE MROW2, MCOL: PRINT "(G)etSample, (F)illSyringe, (D)ispense,
(S)topPump, c(L)ean, (Q)uit"
END SUB

FUNCTION PumpAmt% (vol!)
' ***********************************************************************
' ** PURPOSE:  Pump volume sent
' ** PARAMETER: vol! - the volume to be "moved"
' ** Returns:  Indicator if it was stopped by user.
' ***********************************************************************
   IF spd1! = 0 THEN
    SendPumpCmd CEASE, dum$
    EXIT FUNCTION
   END IF
   tim! = TIMER
   DO
     chek! = TIMER - tim!
   LOOP UNTIL (Query%(READY) = OK)
   Pump1Pos% = Query%(POSITION)        ' ** Get pump position
   incr% = CINT((vol! / SYRVOL!) * DIVISIONS)  ' ** Determine how far it will move
   StartPos% = Pump1Pos%               ' ** Keep track of starting position
   EndPos% = Pump1Pos% + incr%         ' ** Keep track of ending position

   IF (EndPos% > DIVISIONS) OR (EndPos% < 0) THEN
    BEEP: BEEP: BEEP                   ' ** This should already be
    notice ("Too far")                 ' ** taken care of but it is
    INPUT junk$                        ' ** kept just in case
   END IF
  ' ** Estimate How long it takes at the currently set speed, spd1! ml/min?
  ' ** The estimated time neglects backlash compensation.
   tim! = ABS((vol! / spd1!) * 60)
   place$ = LTRIM$(RTRIM$(STR$(EndPos%)))          ' ** Determine stopping pt
   SendPumpCmd PUMPIT, place$                      ' ** Send command to pump
   StartTim! = TIMER
   flag% = OK
   ' ************ LOOP WHILE PUMPING to see if Stop is requested ********
```

```
  DO
   IF UCASE$(INKEY$) = "S" THEN
      StopPump (StartPos%)
      flag% = BAD
   ELSE
      timesofar! = (TIMER - StartTim!) / tim!
      guessvol! = timesofar! * vol!
      notice ("About " + STR$(guessvol!) + " mls pumped so far.")
   END IF
  LOOP UNTIL (((TIMER - StartTim!) >= tim!) OR (flag% = BAD))
  ' ******* END LOOP if finished or if Stop is hit ********************
  PumpAmt% = flag%
  DO
  LOOP UNTIL Query%(READY) = OK
END FUNCTION' ** PumpAmt **

FUNCTION PumpInit% (NumStr$)
' ************************************************
' ** Assumes Only one pump for now
' ** Purpose:  To initialize the syring pump
' ** Returns:  Integer which indicates if successful or not
' ************************************************
 Pump1Pos% = 0                          ' ** Move pump to empty/zero volume
 SendPumpCmd INIT, dum$
 DO                                     ' ** Check until pump is ready to go **
   rd1% = Query%(READY)
   Pause .1
 LOOP UNTIL ((rd1% = OK) OR (rd1% = NOPUMP))
 PumpInit% = rd1%
END FUNCTION' ** PumpInit **

FUNCTION Query% (which%)
' ************************************************************************
' ** Purpose:  Indicates syringe position or readiness for accepting commands.
' ** Parameters:  which% indicate operation, position or readiness
' ** Returns: if which% = POSITION  returns syringe position (1 to DIVISIONS steps).
' **         if which% = READY returns BAD, not ready, or OK, ready.
' **           Only the 0 or 1 of the 5th bit of the response is used
' **           here.  The response contains more specific error
' **           information if needed.
' **         if there is no pump response return NOPUMP
' ************************************************************************
  flag% = OK
```

```
DO
  IF LOC(1) > 0 THEN
    junk$ = INPUT$(LOC(1), #1)   ' ** Clean buffer
  END IF
  flag2% = OK
  SendPumpCmd which%, dum$
  flag2% = AnyResp%
  IF (flag2% > BAD) THEN
    ' ************************************************************
    instr$ = ""                                  ' ** Clear InStr$
    instr$ = GetResp$
    InLen% = LEN(instr$)                         ' ** determine length of InStr$
    status$ = ""              ' ** clear x$
    status$ = FindStat$(instr$, InLen%, catch%)  ' ** Returns status of request
    IF (catch% = BAD) THEN
      flag% = BAD
    ELSE
     compare% = (ASC(status$) AND SPACE)
     ' ********************** IF LOOKING FOR READINESS ***********
     IF (which% = READY) THEN
       IF (compare% = BAD) THEN
         Query% = BAD
        ELSE
         Query% = OK
        END IF
      ' ********************** IF LOOKING FOR POSITION *********
      ELSEIF (compare% = BAD) THEN ' ** AND which% = POSITION
        flag% = BAD                              ' ** Not ready to give position yet
      ELSE '** (which% =POSITION) and (compare% = SPACE)
        messg$ = ""                              ' ** clear messg$
        messg$ = GetPosResp$(instr$, InLen%, catch%)
        Pump1Pos% = VAL(messg$)                  ' ** Put value of position
        Query% = Pump1Pos%                       ' ** Return position
      END IF
      ' ******************* END IF READY OR POSITION **********
    END IF
    ' ********************** END IF ABLE TO GET INFO FROM PUMP ***
  END IF
  ' ************************* END IF ABLE TO EST. Comm with PUMP ***
LOOP UNTIL ((flag% = OK) OR (flag2% = BAD))
' ***************************** END DO LOOP ***********************
junk$ = INPUT$(LOC(1), 1)    ' ** Clean out input buffer
IF (flag2% = BAD) THEN
```

```
    Query% = NOPUMP
  END IF
END FUNCTION ' ** Query **

SUB SendPumpCmd (which%, instr$)
SELECT CASE which%
 CASE RIGHT
    PRINT #1, "/" + PUMPADDR$ + "OR" + CHR$(RET)
 CASE LEFT
    PRINT #1, "/" + PUMPADDR$ + "IR" + CHR$(RET)
 CASE POSITION
    PRINT #1, "/" + PUMPADDR$ + "?" + CHR$(RET)
 CASE READY
    PRINT #1, "/" + PUMPADDR$ + "Q" + CHR$(RET)
 CASE CEASE
    PRINT #1, "/" + PUMPADDR$ + "T" + CHR$(RET)
 CASE PUMPIT
    PRINT #1, "/" + PUMPADDR$ + "A" + instr$ + "R" + CHR$(RET)
 CASE THISRATE
    PRINT #1, "/" + PUMPADDR$ + "V" + instr$ + "R" + CHR$(RET)
 CASE INIT
    PRINT #1, "/" + PUMPADDR$ + CMD$ + "R" + CHR$(RET)
 END SELECT
 Pause .5
END SUB

SUB SetRate (Fl!)
' ************************************************************************
' ** PURPOSE:  Set flow rate on pump, make sure the rates are valid first
' ** Parameters: Fl! is the flow rate in ml/min.
' **          Fl! = (# of halfsteps/s)(60 s/min)(Syringe1Vol!/6000 halfsteps)
' **          i.e.  desired # of halfsteps/s, hs = Fl! * 100 /SYRVOL!
'
 ************************************************************************
    hs% = CINT(Fl! * 100 / SYRVOL!)
     ' ** spd1! is the same as Fl! within a resolution of SYRVOL!/6000.
    spd1! = (hs% / 100) * SYRVOL!
    DO                                      ** Loop until pump is ready
     temp% = Query%(READY)
    LOOP UNTIL (temp% = OK) OR (temp% = NOPUMP)
    IF temp% = OK THEN
      HsSTr$ = LTRIM$(RTRIM$(STR$(hs%)))
      SendPumpCmd THISRATE, HsSTr$
```

```
    END IF
END SUB ' **  SetRate **

FUNCTION SetVol% (vol!)
' ************************************************************
' ** Moves the syringe to an absolute position in terms of
' ** an absolute volume, vol!.
' ** This is done by calculating the volume change based on
' ** the syringe's current position and then using subroutine
' ** PumpAmt(vol!) where vol! is the volume change.
' ************************************************************
  curvol! = HowMuch!                        ' ** determine current volume
  IF (curvol! > NOPUMP) THEN
    DeltaVol! = vol! - curvol!              ' ** determine change in volume
    SetVol% = PumpAmt(DeltaVol!)            ' ** dispense that amount
  ELSE
    SetVol% = NOPUMP
  END IF
END FUNCTION' **  SetVol **

SUB StopPump (StartPos%)
' ************************************************************
' ** PURPOSE:  Stop pump and print out vol delivered
' ** PARAMETERS:  StartPos% - where the pump started pumping
' ************************************************************
    SendPumpCmd CEASE, dum$
    fin% = Query%(POSITION)
    dif! = ABS(fin% - StartPos%) * SYRVOL! / DIVISIONS
    diff$ = "Delivered:  " + STR$(dif!) + " ml"
    notice (diff$)
END SUB

FUNCTION Switch% (jcount%)
' ** Purpose:  Switches syringe valve
' ** Returns:  New syringe valve position (right or left)
' ** Parameter: jcount% indicate if valve is currently right or left
' ************************************************************************
  IF (jcount% = LEFT) THEN                          ' ** if valve is left
    SendPumpCmd RIGHT, dum$                 ' ** switch right
    Switch% = RIGHT                         ' ** return right
  ELSE                                      ' ** otherwise it's right
    SendPumpCmd LEFT, dum$                  ' ** switch left
    Switch% = LEFT                          ' ** return left
```

```
 END IF
END FUNCTION ' ** Switch **

FUNCTION UserInput% (Messag$, instr$)
' **  UserInput (formally sub MenuInput)
' **  Purpose:  Handle User Input for prompts and menus
' **  Returns:  the status of the input (OK for good input, BAD for <ESC>)
' **  Parameters:  Messag$ - The prompt for the user to respond
' **               InStr$ - The user's response
' ********************************************************************
   status% = OK
   LOCATE MROW, MCOL: PRINT SPACE$(77): LOCATE MROW2, MCOL: PRINT
SPACE$(77)
  ' ********************** PRINT OUT PROMPT ******************
   LOCATE MROW, MCOL: PRINT Messag$;
   PrintLoc% = 3 + LEN(Messag$)
   instr$ = ""
   InLen% = 0
  ' **********************DO LOOP **********************
   DO
    char$ = INPUT$(1)
    CharNum% = ASC(char$)
    notice (" ")              ' ** Clears Old warnings
   ' ************************ CASE ********************
    SELECT CASE CharNum%
    ' ****************** NUMBER TYPED *************
     CASE PT, ZERO TO NINE
      instr$ = instr$ + char$
      InLen% = InLen% + 1
      LOCATE MROW, PrintLoc% + InLen%: PRINT char$;
    ' ******************* ESCAPE KEY HIT *************
     CASE ESC
      status% = BAD
    ' ******************* BACKSPACE KEY HIT **********
     CASE BACKSPC
      IF InLen% > 0 THEN
        LOCATE MROW, PrintLoc% + InLen%: PRINT " " ' ** backspace in array
        InLen% = InLen% - 1
        instr$ = MID$(instr$, 1, InLen%)
      END IF
    ' ******** Something Other than a number, Return, ESC or backspace *****
     CASE ELSE
      IF (CharNum% <> RET) THEN
```

```
      LOCATE NOTICEROW, MCOL: PRINT "BAD INPUT, Ignoring character"
     END IF
  ' ********
   END SELECT
  ' ********************* END CASE *********************
  LOOP UNTIL ((CharNum% = RET) OR (status% = BAD))
                  ' ** The return or ESC key was pressed.
' ***********************END DO LOOP ***********************
  UserInput% = status%    ' ** Return status
END FUNCTION ' ** UserInput **
```

APPENDIX C: SOFTWARE FOR DETERMINING STICKING PROBABILITIES AND RATE CONSTANTS FROM SPR ADSORPTION CURVES

This program was written in C using CodeWarrior (Metrowerks) for Macintosh. The program really contains two versions of itself. One called SmoothAlgo which performs the Fick's Law / rate / sticking probability calculations on the "raw" SPR data and performs the Savitzy - Golay derivative smoothing. The other version, FitAlgo, assumes the SPR data was first smoothed as described in Equation 11 in Chapter 4. The code was not fully optimized.

```
/*********************************************************/
/*File: Include.h: contains necessary include files for the entire project */
/*********************************************************/
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <types.h>
#include <MacHeaders.pch>  /* CodeWarrior Specific */
#include <stdlib.h>

/*************************************************/
/* File:     Param.h                             */
/* Contains:  Constants often need to be adjusted */
/*              Or constants needed by all files  */
/*************************************************/
/* Molecular Weight */
#define MASS              52800.0       /* Mwt for SA */

/* Solution Concentration */
#define CONC                            5.2e14        /* Conc. in molec/cm3 0.05
mg/ml SA*/

/*Diffusion Constant and Depletion Layer */
#define DIFF              7.4e-7        /* SA */
#define DIFFLAYER         0.0019        /* Diffusion layer 0.0019 for SA */

/* System specific constants
#define TOTH              5.0e-2        /* Total "Height" 5x10^-2 */
#define SEGS              2000          /* Divide height up in SEGS+1 segments*/
```

```
#define DEL_STEP        1e-7        /* Every DEL_STEP second is calculated */
#define STOPTIME        4.8         /* time flow stops SA (NewEEB)*/
/* #define STOPTIME     1000        /* simulate constant flow*/

/* Necessary for S-Golay 1st derivative smooth */
#define NORM            10          /* 28 for 7, 10 for 5, 60 for 9 */
#define SMNUM           5           /* Smoothing value see above */
/* change SMNUM you have to change conv[] (CalcFunctions.c Function
SmoothAlgo)*/

/* Generic Constants */
#define NA              6.022e23    /* Avogadros number, molec/mol */
#define ATSURFACE       0           /* First step at the surface */

/* Constants for determining Sticking */
#define TwoPi           6.28        /* 2 pi */
#define K               1.38e-23    /* Boltzmann K in J/K */
#define T               298.0       /* Room temperature 298 K*/

/*************************************************************/
/*  File:  main.c                                            */
/*  Program written to do kinetic analysis of adsorption on surfaces  */
/*  Linda S. Jung 2/20/98                                    */
/*************************************************************/
#include "include.h"
#include "Param.h"

/* Functions in Funct.c */
int StartPrompt();          /* Asks user if using raw (to be smoothed) data or fit data
*/
int EndPrompt();            /* Asks user they want do another */
void WhichFile (char *file, char *file2);    /* Asks user for data file */
void PauseFn();             /* Pauses program during execution */

/* Functions in CalcFunctions.c */
void SmoothAlgo(FILE *datafile, FILE *resultfile ,double h, double vt);
            /* Algorithim to use when "cleaned" raw data is used */
void FitAlgo(FILE *datafile, FILE *resultfile ,double h, double vt);
            /* Algorithim to use for fit data files */
void RecordHeader(FILE* resultfile, char* result);          /* Create datafiles headers */
void RecordData(FILE* resultfile, double realtime, double cov, double rate, double
stickprob, double conc[]);    /* Enter data into file */
void PrintData(double realtime, double cov, double stickprob, double conc[]);
```

```
                /* Print Data to screen */
double CalcThermVel();      /*  Returns Thermal velocity */
void InitialConc(double conc[]);              /* Initialize array */
double CalcFlux(double surf_conc);
                /* Returns Flux based on Thermal velocity and surface conc */
double CalcSticking(double rate, double surf_conc);
                /* Returns Sticking Probability */

/************************/
/*  BEGIN MAIN PROGRAM */
/************************/
void main( void )
{
 /* Initialize Variables */
 FILE *datafile;              /* Pointer to file containing data */
 FILE *resultfile;            /* Pointer to file for results */
 char filename[50], result[50];
 int reply;
 double h, vt;

 do {
    reply = StartPrompt();    /* Prompt for raw or fit type file */
     /*****************************************************************/
     /* NOTE:  raw file does not mean raw file take from the SPR directly, this      */
     /*        file has been "cleaned" with the program CleanSPR which makes sure */
     /*        there is a data point every 0.5 s by extrapolation, etc.             */
     /*        Results will probably vary if the actual raw data is used since the      */
     /*        present SPR program often skips data points
*/
     /*                                                                              */
     /*        Fit data are typically data calculated base on a triple exponential      */
     /*         fit to SPR data                                                      */
     /*                                                                              */
     /* ALSO note the datafiles have to only be in the format of                     */
     /*                                                                              */
     /*         number number                                                        */
     /*                                                                              */
     /*    I did not include error checks to deal with other (such as a file         */
     /*    which may still include headers) so the program will probably just        */
     /*    freeze up or crash, sorry, but I make my files that way so I don't        */
     /*    feel the need to check for files which aren't made that way
*/
    /*****************************************************************/
```

```
    WhichFile(filename, result);              /* Calls whichfile to get datafilename */
    datafile = fopen(filename, "r");          /* Opens that file for reading */
    if (datafile != NULL) {                   /* If file exists */
       resultfile = fopen(result, "w");       /* Create result datafile (file.done) */
       printf ("Creating file %s\n", result);

       /* Create Headers for result file */
       RecordHeader(resultfile, result); /* Create header for datafile */
       h = TOTH/SEGS;                         /* Calculate height of each layer */
       vt = CalcThermVel();                   /* Calculate vt, thermal velocity */
       if (!reply) SmoothAlgo(datafile, resultfile, h, vt);    /* if r call SmoothAlgo */
       else FitAlgo(datafile, resultfile, h, vt);                              /* if f call
FitAlgo */

        /* Close All Files */
       fclose(datafile);
       fclose(resultfile);

    } /* END IF data exists */
    else printf("Can't find that file.\n");

    reply = EndPrompt();
  } while (reply==0);
  printf ("All done.\n");
  exit(0);
}/*  END MAIN PROGRAM */

/*************************************************/
/* File:     Prompts.c                           */
/* Contains:  Functions ask user for input       */
/*************************************************/
#include "include.h"
#include "Param.h"

#define TAIL ".done"          /* Defines what is put on the end of the calculation file */

/* Functions in Prompts.c */
int StartPrompt();            /* Asks user if using raw (to be smoothed) data or fit data
*/
int EndPrompt();              /* Asks user they want do another */
void WhichFile (char *file, char *file2);      /* Asks user for data file */

int StartPrompt ()
```

```
/*************************************************************/
/* Function:  Start Prompt                                   */
/*  Returns: int indicating whether user is using a raw or fit data file  */
/*************************************************************/
{
 char ans[1];                  /* charcter string to catch user entry */
 int  reply, check;            /* reply indicates answer and */
                               /* check is a flag for bad entry */
  do {
        check=0;                              /* initialize check for each prompt */
        reply=0;                              /* initialize reply */
        printf ("\nRaw or Fit data (r or f)? "); /* Prompt for file type */
        scanf("%s", ans);                     /* Read in user input  */

        if (!(strncmp(ans, "r", 1))) reply=0;       /* if r is entered reply stays 0 */
        else if (!(strncmp(ans, "f", 1))) reply++; /* if f is entered reply = 1 */
        else                                  /* entry is invalid ask again */
            {
                check++;                      /* set flag */
                printf("Sorry you did not enter a valid response.\n");
             }
  } while (check);                            /* repeat if flag, check, has been set */
  return(reply);
} /********************* End StartPrompt *********************/

int EndPrompt ()
/*************************************************/
/* Function:  End Prompt                         */
/*  Returns: int indicating whether user has another  */
/*              data file to do                  */
/*************************************************/
{
   char ans[1];                 /* Character string to catch user entry */

    printf ("\nAnother File (y/n)? ");          /* Prompt */
    scanf("%s", ans);                          /* Read in user input */
    return(strncmp(ans, "y", 1));              /* return 0 is yes, otherwise some value */
                                               /* anything but y is consider no */
} /********************* End EndPrompt *********************/

void WhichFile (char *file, char *result)
/*************************************************************/
/* Function:  asks user for the datafile name sends back   */
```

```
/*          the result file as filename appended with    */
/*          TAIL at the end to indicate the finished file */
/*************************************************************/
{
 int length;
 char label[10];

  sprintf (label, TAIL);                    /* Proper tail to result file */
  printf ("Enter datafilename: ");          /* Prompt user for filename */
  scanf("%s", file);                        /* Read in user input */
  length = strlen (file);                   /* determine length of input */
  memcpy(result, file, length);             /* copy into resultname */
  result[length] = '\0';                    /* Clean up the end of the name */
                                            /*  gets rid of extraneous junk */
  strcat(result, label);                    /* Create resulting file as    */
                                            /*  orignalfilename.TAIL       */
} /******************** End WhichFile ********************/

/*************************************************************/
/* File:    CalcFunctions.c                                  */
/* Contains:  Functions to do the calculations and report results */
/*************************************************************/
#include "include.h"          /* file containing all necessary include files */
#include "Param.h"            /* file with all the necessary parameters for calculations */

/* Functions in CalcFunctions.c */
void SmoothAlgo(FILE *datafile, FILE *resultfile ,double h, double vt);
                /* Algorithim to use when "cleaned" raw data is used */
void FitAlgo(FILE *datafile, FILE *resultfile ,double h, double vt);
                /* Algorithim to use for fit data files */
void RecordHeader(FILE* resultfile, char* result);
                /* Create Header for datafiles */
void RecordData(FILE* resultfile, double realtime, double cov, double rate, double
stickprob, double conc[]);
                /* Enter data into file */
void PrintData(double realtime, double cov, double stickprob, double conc[]);
                /* Print Data to screen */
double CalcThermVel();                  /*  Returns Thermal velocity */
void InitialConc(double conc[]);        /* Initialize array */
double CalcFlux(double surf_conc, double vt);
                /* Returns Flux based on Thermal velocity and surface conc */
double CalcSticking(double rate, double surf_conc, double vt);
                /* Returns Sticking Probability */
```

```
void PauseFn();                          /*Pauses program during execution */

double CalcThermVel()
/*************************************************************/
/* Function CalcThermVel returns the calculated thermal      */
/*  velocity.  Based on vt = sqrt(1/2pi kT/m) resulting       */
/*  units are in cm/s (converted with (kg m2)/s2 J)           */
/*   kg/1000g, NA, and 100cm/m                                */
/*************************************************************/
{
        return(100*sqrt((1/TwoPi)*(K*T*NA*1000/MASS)));
} /* END CalcThermVel */

void InitialConc(double conc[])
/*************************************************************/
/* Function:  Initializes array to initial concentration */
/*************************************************************/
{
        int i;                          /* counter */
        for (i=0; i<=SEGS; i++) conc[i] = CONC;   /* fills array with CONC */
} /* End InitialConc*/

void SmoothAlgo(FILE* datafile, FILE* resultfile, double h, double vt)
/*************************************************************/
/* Function SmoothAlgo   */
/*************************************************************/
{ /* Begin Smooth */
  double conc[SEGS];          /* Array which holds concentrations of each slot */
  double dconc, time, dt, ads, dads;
  double tracktime, realtime=0;
  double rate, cov=0;
  double gain, loss, totflux, stickprob;

 /* structure hold time and coverages for each set of data */
  struct dataline {
       double amt;
       double sec;
  } window[SMNUM]={0,0, 0,0, 0,0, 0,0, 0,0};

  /* counters, tracking variables and flags */
  double count, j=0;
  double temp, timejump;
  int k, read = 0;
```

```
int i = 0;
int jump=1;                 /* tracks how steps at a time to take as time steps increase
*/
int fract=2;                /* Fractions of seconds recorded for data */
int stop, flag = 0;         /* flags which tell loops when to quit early */
int mid = ((SMNUM+1)/2)-1;
            /* Middle number of data set, dictated by number used for smooth */
int next = (SMNUM+1)/2;                 /* Number next to Middle of data set */
int last = SMNUM-1;                     /* Very last number of set */
int conv[]={ -2, -1, 0, 1, 2}; /* Factors used to smooth data, here for 5 smooth */

InitialConc(conc);          /* Initialize Array */
count = 0.0;                /* Initialize counter */

do { /* BEGIN outer DO LOOP */
    for (k=0; k<SMNUM; k++) window[k]=window[k+1];
                            /* Shift all values array to read in another */
    fscanf(datafile, "%Lf", &time);         /* Read first column, time */
    fscanf(datafile, "%Lf", &ads);          /* Read second column, coverage */
    window[last].sec = time;
            /* Each newly read data set go at the end of the structure array */
    window[last].amt = ads;
    read++;                 /* Keep track of how much data is read */
    if (read > mid) {
                /* When enough data is read to do smooth calc start calculating */
      timejump = window[next].sec - window[mid].sec;
      temp = 0.0;
      for (k=0; k<SMNUM; k++) temp = window[k].amt*conv[k] + temp;
      dads = temp/NORM;
      if (timejump > 0) {
        rate = dads/(timejump);  /* Calculate Rate */
        dt = DEL_STEP * jump;
        j=0;
        do {  /* Middle do loop */
            count++;                    /* Increment for loop counter */
            stop = 0;                   /* initalize do loop flag */
            i = ATSURFACE;              /* initalize do loop counter */
            loss = rate;                /* Set Loss to rate ads */
            cov = cov + (rate*dt);      /* Track surface coverage */
            j = j + jump;               /* Advances J an appropriate step */
            tracktime = realtime + dt;  /* Determine time */
            if (realtime > time) {      /* Check to make sure it in range */
              dt = time - realtime;     /* It the step was too big resize */
```

```
                realtime = time;          /* reset the actual time, used for rounding */
              }                           /* HOPEFULLY never used */
              else realtime = tracktime;

              do {/* BEGIN inner DO LOOP */
              /*************************************************/
              /*  Calculate concentrations in layers/slots     */
              /*   Yes functions would have this part easier to */
              /*   to read but will significantly slow the program */
              /*   on some of the larger files, so stick with the mess */
              /*************************************************/
                   if (i > ATSURFACE) loss = gain;
                   if (conc[i] == conc[i+1]) {/* If layer & next are the same */
                      gain = 0;           /* concentration then no diffusion*/
                      stop++;             can stop  */
                   }
                   else if (i == SEGS-1) gain = 0; /* Last Layer only*/
                   else if
((realtime<STOPTIME)&&(i>(DIFFLAYER*SEGS/TOTH))){
                      gain = ((CONC - conc[i])*DIFF/h);
                      stop++;/* In this time regime calc stops at this layer */
                   }
                   else { /* "Normal" case */
                      dconc = (conc[i+1] - conc[i]);
                      gain = (DIFF*dconc/h);
                   }
                   totflux = loss-gain;
                   conc[i] = conc[i] - ( (totflux*dt)/h );
                   if (conc[i] < 0) {          /* If things go wrong quit! */
                      printf("ABORT! at %d: %Le with %Le jumping %d, dt %Le,
totflux %Le\n", i, realtime, conc[i], jump, dt, totflux);
                      stop++;
                      flag++;
                   }
                   i++;                        /* On to the next layer */
                   if( Button() ) PauseFn();

              } while ((i < SEGS) && (!stop));
              /* END inner DO WHILE for each segment */

              if (count >= (1/(fract*DEL_STEP*jump) ) ) {
              /* Store set number of datapoints */
                 /* Calc a sticking probability */
```

```
                stickprob = CalcSticking(rate, conc[ATSURFACE], vt);
                /* Print results to file */
                RecordData(resultfile, realtime, cov, rate, stickprob, conc);
                /* Print results to screen */
                PrintData(realtime, cov, stickprob, conc);
                if (jump < 200) {          /*  More like 128 prevents steps with x.x */
                   jump=jump*2;            /*  increase step size */
                   dt = dt*2;              /*  which means dt increases also */
                }
                count = 0;  /* Reset Counter */
              } /* END IF for storing datapoints*/
            } while ((realtime < time) && (!flag)); /* END middle do while */
            } /* END IF timejump > 0 */
        } /* END if array has enough data*/
    } while (!feof(datafile) && !flag); /*  END outer DO-WHILE: end of file or a flag  */
} /* END SMOOTH */

void FitAlgo(FILE* datafile, FILE* resultfile, double h, double vt)
/*********************************************************/
/* Function SmoothAlgo  */
/*********************************************************/
{
 double conc[SEGS];          /* Array which holds concentrations of each slot */
 double dconc, time, ads, dads, dt;
 double prevtime=0;
 double prevads=0;
 double realtime,time_step, rate, cov;
 double gain, loss, totflux, tracktime;
 double stickprob;
 /* counters, tracking variables and flags */
 int stop, flag = 0;             /* flags which tell loops when to quit early */
 int jump=1;                     /* tracks how steps at a time to take as time steps increase
*/
 int fract=2;                     /* What fractions of seconds time is recorded in */
 int i = 0;
 double count, j=0;
 /*********/
 /* START */
 /*********/
 InitialConc(conc);                          /* Initialize Array */
 count = 0.0;                                /* Initialize counter */
 do { /* BEGIN outer DO LOOP */
      fscanf(datafile, "%Lf", &time);        /* Read first column, time */
```

```
fscanf(datafile, "%Lf", &ads);          /* Read second column, coverage */
if ((time-prevtime) > 0) {              /* Make sure time progresses */
  dads = ads-prevads;                   /* Calc change in coverage */
  cov = prevads;
  rate = dads/(time-prevtime);          /* Calculate Rate */
  time_step = (time-prevtime)/DEL_STEP; /* How many DEL_STEPS */
  realtime = prevtime;
  dt = ((time-prevtime)/time_step) * jump;
  j=0;
  do {
       count++;                         /* Increment for loop counter */
       stop = 0;                        /* initalize do loop flag */
       i = ATSURFACE;                   /* initalize do loop counter */
       loss = rate;                     /* Set Loss to rate ads */
       cov = cov + (rate*dt);           /* Track surface coverage */
       j = j + jump;                    /* Advances J an appropriate step */
       tracktime = realtime + dt;       /* Determine time */
       if (tracktime > time) {          /* Check to make sure it in range */
          dt = time - realtime;         /* It the step was too big resize */
          realtime = time;              /* reset the actual time, used for rounding */
       }
       else realtime = tracktime;

       do { /* BEGIN inner DO LOOP */
       /*******************************************************/
       /*  Calculate concentrations in layers/slots           */
       /*   Yes functions would have this part easier to
         */
       /*   to read but will significantly slow the program   */
       /*   on some of the larger files, so stick with the mess */
       /*   Yes I realize it's repetitive with SmoothAlgo     */
       /*******************************************************/
            if (i > ATSURFACE) loss = gain;
            /* if not at the surface what the prior layer */
            /*   had just gained from diffusion will be lost here */
            dconc = (conc[i+1] - conc[i]);
                        /* Calc difference in concentration between layers */
            gain = (DIFF*dconc/h);     /* Calc gain from diffusion */

            if (conc[i] == conc[i+1]) { /* If layer & next are the same */
               gain = 0;                /* concentration then no diffusion*/
               stop++;                  /* can stop  */
            }
```

```
                else if (i == SEGS-1) gain = 0;  /* Last Layer only*/
                else if ((realtime<STOPTIME)&& (i>(DIFFLAYER*SEGS/TOTH)) )
                {
                   gain = ((CONC - conc[i])*DIFF/h);
                   stop++;  /* In this time regime calc stops at this layer */
                }
                totflux = loss-gain;
                conc[i] = conc[i] - ( (totflux*dt)/h );
                if (conc[i] < 0) {              /* If things go wrong quit! */
                   printf("ABORT! at %d: %Le with %Le jumping %d\n", i, realtime,
conc[i], jump);
                   stop++;                      /* Set flags */
                   flag++;
                }
                i++;                            /* On to the next layer */
                if( Button() ) PauseFn();    /* For my benefit, to pause the program */
             } while ((i < SEGS) && (!stop));
             /* stop when all layers are done or when told to stop */
             /*****************************************/
             /* END inner DO WHILE for each segment */
             /*****************************************/
             if (count >= (1/(fract*DEL_STEP*jump) ) ) {
                          /* Store set number of datapoints */
                /* Calc a sticking probability */
                stickprob = CalcSticking(rate, conc[ATSURFACE], vt);
                /* Print results to file */
                RecordData(resultfile, realtime, cov, rate, stickprob, conc);
                /* Print results to screen */
                PrintData(realtime, cov, stickprob, conc);
                if (jump < 200) {
                   jump=jump*2;              /*   increase step size */
                   dt = dt*2;                /*   which means dt increases also */
                }
                count = 0;                   /* Reset Counter */
             } /* END IF */
          } while ((realtime < time) && (!flag));          /* END while */
       } /* END IF time has passed */
       /*  make current values previous */
       prevads = ads;
       prevtime = time;
    } while (!feof(datafile) && !flag); /*  END outer DO-WHILE: end of file or a flag  */
  } /* End FitAlgo */
```

```
void RecordHeader(FILE* resultfile, char* result)
/*********************************************************************/
/* Function RecordHeader records header for the selected data recorded in the resultsfile
*/
/*              as indicated in RecordData. Send pointer to the result file as
*/
/*              well as the file name                                          */
/*********************************************************************/
{
  /* Create Headers for result file */
     fprintf(resultfile, "Datafile %s\n", result) ;    /* Records datafile name in file */
  /* Record parameters: */
     fprintf(resultfile, "D=%Le\tH=%Le\tSlots=%d\tConc=%Le\tDiffDist=%Le\tStart-
%Le\n", DIFF, TOTH, SEGS, CONC, DIFFLAYER, STOPTIME) ;

  /* Record parameters: Timesteps, Molecular Weight, Temp (usually room temp) */
     fprintf(resultfile, "InitialTimeStep=%Le\tMwt=%Le\tTemp=%Le\n",
DEL_STEP,MASS, T) ;

  /* Data column titles */
    fprintf(resultfile,
"Time\tEstCov\tEstMRate\tRate\tStickingProbability\tConcentration(at
surface)\t5\t10\t20\t30\t40\t50\t60\t80\t100\t150\t200\t250\t300\t400\t500\t600\t700\t800
\t900\t1000\t1500\t1999\n") ;

  /* Initial line, just to have initial conc at time 0, probably not necessary */
     fprintf(resultfile, "0\t\t\t\t\t%Le\n",CONC) ;
}/* END RecordHeader */

void RecordData(FILE* resultfile, double realtime, double cov, double rate, double
stickprob, double conc[])
/*********************************************************/
/* Function RecordData records selected data to the resultsfile.          */
/*              Send pointer to file, time, coverage, rate, S and conc array  */
/*********************************************************/
{
fprintf(resultfile,"%Lf\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le
\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%
Le\t%Le\n", realtime, cov, rate, rate/conc[0], stickprob, conc[ATSURFACE], conc[4],
conc[9], conc[19], conc[29], conc[39], conc[49], conc[59], conc[79], conc[99],
conc[149], conc[199], conc[249], conc[299], conc[399], conc[499],conc[599], conc[699],
conc[799], conc[899], conc[999], conc[1499], conc[1999]);
}/* RecordData*/
```

```
void PrintData(double realtime, double cov, double stickprob, double conc[])
/*************************************************************/
/* Function PrintData prints some selected data to the screen so I can monitor */
/*              the progress, send time, coverage, S and conc array          */
/*************************************************************/
{
              printf("%Lf\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\t%Le\n",
              realtime, cov, stickprob, conc[ATSURFACE], conc[4],
              conc[9],conc[49],conc[99],conc[999],conc[1999]);
} /* END PrintData */

double CalcFlux(double surf_conc, double vt)
/*************************************************************/
/* Function CalcSticking returns the calculated flux to the surface          */
/*              send the concentration and vt, the thermal  velocity          */
/*************************************************************/
{
              return(vt*surf_conc);
}/* END CalcFlux */

double CalcSticking(double rate, double surf_conc, double vt)
/*************************************************************/
/* Function CalcSticking returns calculated sticking probability.            */
/*              A rate of adsorption (double), concentration,
              */
/*              and vt,the thermal velocity, is sent.                        */
/*              Calls to CalcFlux is made to get flux                        */
/*************************************************************/
{
              return(rate/CalcFlux(surf_conc, vt));
}/* END CalcSticking */

void PauseFn()
/*************************************************************/
/* Function PauseFn() Pauses execution of program until                       */
/*              the return character is hit      NOT NECESSARY              */
/*************************************************************/
{
  char ans;
              printf("Press Return to resume\n");
              ans = getchar();
   return;
```

```
}/* End PauseFN */
```

APPENDIX D: SOFTWARE FOR DATA ACQUISITION ON THE LEYBOLD SYSTEM

The software is written in C in National Instruments Lab Windows CVI. An original version was supplied by M. A. Leskovar from the Olmstead group. It was written for angle resolved XPS. The program was cleaned up, modularized and completely rewritten for our purposes. The program is designed to interact with the Data Acquisition Unit (DAU) on the Leybold system to do XPS, AES and ISS experiments from the Gateway PC (however not of these options were implemented). Another feature, Thicken, was written for a specific type of ISS experiments performed by S. C. Parker. Missing from this appendix is the user interface, which is contructed graphically within Lab Window CVI.

```
/************************************************************/
/* File: protttype.h - contains prototypes for functions.   */
/*                                                          */
/* Changes:  6-21-95:  LG & LSJ removed angle related code  */
/*           8-16-95:  Declaration of SaveXPSData changed,  */
/*                     added int subnumber                  */
/*           8-6-96:  Modularize program - wrote DatetoStr, DrawGraph, */
/*                     ReGraph and CheckOrder               */
/*           8-7-96:  SetStep and Round added               */
/*           8-13-96:  Added RoundStep and OkDwell          */
/*           12-4-96:  Added NoAESYet, ChkThStps, SaveThickSweep */
/*                     and SaveThickData                    */
/************************************************************/
int GetCount(unsigned short Dwell,unsigned short *Count);
int SendXPSEnergy(double Energy);
void ErrorRoutine(char* function,int error);
double SetStep(double range);
double Round(double number, double fraction);
double RoundStep(double energyinc, double incstep);
int NoAESYet(int Status);
int ChkThStps(int Status);
void MakeFName(char* filename, int runnum, int region) ;
void InttoStr(int num, char* string);
int DatetoStr(char* strmon, char* strday);
int CheckOrder(int KorB, int status);
int OkDwell(double* dwelltimes, int counter, int status);
```

```
int DrawGraph(unsigned int sweeps, int graph);
void ReGraph(int counter, unsigned int* steps);
int IsBitSet(unsigned int Input, unsigned int Bit);
int SetBit(unsigned int Input, unsigned int Bit);
int SaveXPSSweep(int EnergyRegionCounter, unsigned int Index);
int SaveThickSweep(int EnergyRegionCounter, unsigned int Index);
int NewXPSPoint (int EnergyRegionCounter,unsigned
                                int Index,double CurrentEnergy,unsigned short Count);
int DoXPSSweep(int EnergyRegionCounter);
int SaveXPSData(int EnergyRegionCounter, int subnumber);
int SaveThickData(void);
int SaveStatus(void);
int WaitAWhile(short Delay);
int Initialize(void);
int SetInfoWin(void);

/*************************************************************/
/* global.h contains Global declarations                     */
/*                                                           */
/* 8-16-96  Changed XPSData to a one dimensional array       */
/*          Took out extraneous declarations                 */
/* 8-22-95  Changed DataFileLocation to new directory        */
/*          DataFileLocation info is moved to takedata.h
            */
/*          status file information is now here              */
/*          StatusFile now holds just the file name the path */
/*          is stored in STATUS_PATH in global.h             */
/*          this is done so that you can use other "status"  */
/*          files (must be in cvi/xps/proginfo) and also     */
/*          assumes all data files in the new directory only */
/*************************************************************/
TypicalRegion Region;
double CurrentSweep[MAX_XPS_POINTS];
double XPSData[MAX_REGIONS][MAX_XPS_POINTS];
double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
int Handle,GraphHandle,CurrentRegion, KorB;
                                    /* KorB distinguished spectroscopy type */
unsigned int CurrentDataFileNumber;
unsigned int CurrentNumberOfSweeps;
int ControlCode,SweepStatus;
double XMin,XMax;
double YMin,YMax;
```

```
/* Both of these file location rely also on
  FILE_PATH in takedata.h which defines the
  path  c:/cvi/xps */

char StatusFile[25] = "status.txt";
/*************************************************************/
/*  File:  struct.h                                          */
/*  Purpose:  Data structure for the data collection regions */
/*                                                           */
/*  11/13/94 MAL                                             */
/*  06/21/95 LG & LSJ removed angle related portions of software */
/*  06/21/95 LSJ change MAX_REGIONS to define maximum number (5)*/
/*              and substitute into definition of structure. */
/*  07/25/95 LSJ add 2 constants ARRAYS_XPSSTR to define number of */
/*              arrays in XPS structure necesary unfortunately by way the */
/*              software takes data, can be gotten around but more of a */
/*              nuisance.                                    */
/*              OFFSET is also necessary for adjusting to how the data is taken.*/
/*              Both of these constants are used in functions for setting  loop */
/*              the proper loop restraints                   */
/*  10/13/95 LSJ added Another member to XPS structure, HVRange */
/*              This indicates with Range is the max         */
/*************************************************************/

/*************************************************************/
/*  Constants for structure sizes                            */
/*************************************************************/
#define MAX_REGIONS 5
#define MAX_XPS_POINTS 1500
#define MAXSTRING 25
#define ARRAYS_XPSSTR 4
#define OFFSET 1
/*************************************************************/
/*  Structure definitions                                    */
/*************************************************************/
struct InfoStruct
{
        char RegionName[21];
        unsigned int RegionType;
};

struct XPSStruct
{
```

```
                unsigned int NumRegions;
                double PhotonEnergy;
                double HVRange;
                double EnergyBeg[MAX_REGIONS];
                double EnergyInc[MAX_REGIONS];
                double EnergyEnd[MAX_REGIONS];
                double EnergyDwell[MAX_REGIONS];
                unsigned int EnergyNumSteps[MAX_REGIONS];
                unsigned int NumberOfSweeps;
};

typedef struct TypicalRegion
{
                struct InfoStruct Info;
                struct XPSStruct XPS;
}TypicalRegion;

/*************************************************************/
/* File: takedata.h                                          */
/* Purpose: contains defines for region types and these files:  */
/*              xpsout.c                                     */
/*              getcount.c                                   */
/*              loadreg.c                                    */
/*              graphcontrol.c                               */
/*                                                           */
/*   6-21-95  LG & LSJ removed angle related stuff           */
/* To be completed: Constants for getcount.c need to be verified */
/*   8-21-95  Moved more general defines to the top, defined */
/*            the FILE_LABEL ".d"                            */
/*   8-22-95  Changed DataFileLocation to a define called    */
/*            DATA_PATH and changed StatusFile Location part */
/*            define STATUS_PATH and part array containing just */
/*            the filename declared in global.h so that you can */
/*            specify a different filename, but not a different path! */
/*   10-13-95  Added some constants so that the HVRange can be changed
                */
/*   2-20-96  Removed FILE_LABEL                             */
/*   12-4-96  Added Constant TH for thicken data             */
/*            Added Two strings, BAD_THNUM and BAD_CHOICE    */
/*************************************************************/
/* A couple general defines */

#define BUFFER_SIZE 121
```

```
#define MaxThSteps 11

#define TH 3   /* = Thicken Data */
#define AE 2  /* = AES */
#define IS 1   /* = ISS */
#define XP 0  /* = XPS */

/*************************************************************/
/*  DEFINES for Savexps.c Start.c                            */
/*************************************************************/
#define DATA_PATH "c:\\cvi\\xps\\data\\"
#define STATUS_PATH "c:\\cvi\\xps\\proginfo\\"
#define DATA_LABEL "dat"
#define LAST_SPEC 6  /*  This is one beyond the max number of regions */
                     /*   This graph is used after all spectra so the */
                     /*    one that contains the first regions says  */
                     /*    so and not just spectrum                  */
#define WIN_TITLE "File Exists"
#define WIN_ASKS "Do you want to change the run number?"
#define BAD_ENERGY "Bad Energy (start/end), fix it or crash!"
#define BAD_DWELL "Bad Dwell Time!"
#define BAD_THNUM "Too Many Points Per Sweep! (11 max)"
#define BAD_CHOICE "Sorry no AES yet"

/* These two are needed for being able to properly adjust     */
/*   the expression in a SetCtrlVal command in Start.c. They  */
/*   allow the counter to be used to properly set the contant  */
/*   which refers to which place in the interface window the   */
/*   new value is to be written */

#define EI_JMP 4  /*  As the constants for setting the EI is every 4 */
#define EI_OFF 3/* Needed to shift the number set with EI_JMP */

/*************************************************************/
/*    GetParamfile                                           */
/*************************************************************/
#define PWIN_TITLE "New parameter file?"
#define PWIN_NO_FILE "No such paramter file exists"

/*************************************************************/
/*  DEFINES for XPSOUT.C                                     */
/*************************************************************/
/*  Maximum voltage from the High Voltage Amplifier      */
```

```
#define HVA_MID_RANGE 1638.97
#define HVA_HIGH_RANGE 3200
#define HVA_LOW_RANGE 160
/* Minimum step                */
#define MINSTEP 0.025
/* MINSTEP should multiply by proper multiplier depending on Range */
#define LOWMULT 0.1        /* For 1600 eV Range */
#define MIDMULT 1          /*  For .16 KV Range */
#define HIGHMULT 2         /*  For 3.2KV Range */

/*  Maximum voltage output by D/A convertor                */
#define MAX_AO_VOLTS 10.
/*  Device number asigned to board by configuration utility */
#define NI_DEVICE_NUMBER 1
/*  D/A channel hooked up to analyzer        */
#define XPS_AO_CHAN 1

/*  Error codes                */
/*  Input value out of range   */
#define ERR_OUT_OF_RANGE -2
/*  Problems configuring device                */
#define ERR_BAD_CONFIGURE -3

/*************************************************************/
/*  DEFINES for GETCOUNT.C                                   */
/*************************************************************/
#define TRUE 1
#define FALSE 0

/*  Which counter to use to take data */
#define DATA_COUNTER 2
/*  Which counter to use to provide gate pulses */
#define PULSE_COUNTER 1
/*  Where are the data signals coming in from? 6 means SOURCE1 */
#define DATA_SOURCE 6
/*  What do we use for a timebase for the gating pulse? */
/*  2 means use internal 100 kHz clock for 10 micro sec resolution  */
#define PULSE_TIMEBASE 2
#define PULSE_CLOCK_PERIOD 0.00001
#define DELAYTIME 3
#define MINTIME 0.00004

/*************************************************************/
```

```
/* DEFINES for GRAPHCONTROL.C                                          */
/*********************************************************************/
#define NORMAL 0
#define CLOSEBUTTON 1
#define HALTBUTTON 3

/*********************************************************************/
/* DEFINES for bit fields for Region Types                             */
/*********************************************************************/
#define PHOTOREG  0
#define PHOTOMULT  1
#define PHOTOPOINT 2
#define AUGERREG  3
#define AUGERMULT  4
#define AUGERPOINT 5
#define TIMEREG  6
#define TIMEMULT  7
#define TIMEPOINT  8
#define OTHERREG  15
#define OTHERMULT  16
#define OTHERPOINT 17

/*********************************************************************/
/* LabWindows/CVI User Interface Resource (UIR) Include File           */
/* Copyright (c) National Instruments 1996. All Rights Reserved.       */
/*                                                                     */
/* WARNING: Do not add to, delete from, or otherwise modify the contents */
/*          of this include file.                                      */
/* FILE WRITTEN BY LabWindow from the constructed UIR                  */
/*********************************************************************/
   /* Panels and Controls: */

#define GRAPHPANEL              1
#define GRAPHPANEL_GRAPH        2
#define GRAPHPANEL_CLOSEBUTTON      3  /* callback function: GraphControl
*/
#define GRAPHPANEL_HALTBUTTON       4  /* callback function: GraphControl
*/
#define GRAPHPANEL_SWEEPNUM         5
#define GRAPHPANEL_REGIONNUM        6
#define GRAPHPANEL_BUTTONDEC        7

#define GRAPHPAN_2              2
```

```
#define GRAPHPAN_2_GRAPH          2
#define GRAPHPAN_2_CLOSEBUTTON    3  /* callback function: GraphControl
*/
#define GRAPHPAN_2_HALTBUTTON     4  /* callback function: GraphControl
*/
#define GRAPHPAN_2_SWEEPNUM       5
#define GRAPHPAN_2_REGIONNUM      6
#define GRAPHPAN_2_BUTTONDEC      7

#define GRAPHPAN_3                3
#define GRAPHPAN_3_GRAPH          2
#define GRAPHPAN_3_CLOSEBUTTON    3  /* callback function: GraphControl
*/
#define GRAPHPAN_3_HALTBUTTON     4  /* callback function: GraphControl
*/
#define GRAPHPAN_3_SWEEPNUM       5
#define GRAPHPAN_3_REGIONNUM      6
#define GRAPHPAN_3_BUTTONDEC      7

#define GRAPHPAN_4                4
#define GRAPHPAN_4_GRAPH          2
#define GRAPHPAN_4_CLOSEBUTTON    3  /* callback function: GraphControl
*/
#define GRAPHPAN_4_HALTBUTTON     4  /* callback function: GraphControl
*/
#define GRAPHPAN_4_SWEEPNUM       5
#define GRAPHPAN_4_REGIONNUM      6
#define GRAPHPAN_4_BUTTONDEC      7

#define GRAPHPAN_5                5
#define GRAPHPAN_5_GRAPH          2
#define GRAPHPAN_5_CLOSEBUTTON    3  /* callback function: GraphControl
*/
#define GRAPHPAN_5_HALTBUTTON     4  /* callback function: GraphControl
*/
#define GRAPHPAN_5_SWEEPNUM       5
#define GRAPHPAN_5_REGIONNUM      6
#define GRAPHPAN_5_BUTTONDEC      7

#define GRAPHPAN_6                6
#define GRAPHPAN_6_GRAPH          2
#define GRAPHPAN_6_CLOSEBUTTON    3  /* callback function: GraphControl
*/
```

```
#define GRAPHPAN_6_HALTBUTTON       4  /* callback function: GraphControl
*/
#define GRAPHPAN_6_SWEEPNUM         5
#define GRAPHPAN_6_REGIONNUM        6
#define GRAPHPAN_6_BUTTONDEC        7

#define MAINPANEL                   7
#define MAINPANEL_ENERGYBEG_1       2    /* callback function: InfoWin */
#define MAINPANEL_ENERGYINC_1       3    /* callback function: InfoWin */
#define MAINPANEL_ENERGYEND_1       4    /* callback function: InfoWin */
#define MAINPANEL_ENERGYDWELL_1     5    /* callback function: InfoWin */
#define MAINPANEL_ENERGYBEG_2       6    /* callback function: InfoWin */
#define MAINPANEL_ENERGYINC_2       7    /* callback function: InfoWin */
#define MAINPANEL_ENERGYEND_2       8    /* callback function: InfoWin */
#define MAINPANEL_ENERGYDWELL_2     9    /* callback function: InfoWin */
#define MAINPANEL_ENERGYBEG_3       10   /* callback function: InfoWin */
#define MAINPANEL_ENERGYINC_3       11   /* callback function: InfoWin */
#define MAINPANEL_ENERGYEND_3       12   /* callback function: InfoWin */
#define MAINPANEL_ENERGYDWELL_3     13   /* callback function: InfoWin
*/
#define MAINPANEL_ENERGYBEG_4       14   /* callback function: InfoWin */
#define MAINPANEL_ENERGYINC_4       15   /* callback function: InfoWin */
#define MAINPANEL_ENERGYEND_4       16   /* callback function: InfoWin */
#define MAINPANEL_ENERGYDWELL_4     17   /* callback function: InfoWin
*/
#define MAINPANEL_ENERGYBEG_5       18   /* callback function: InfoWin */
#define MAINPANEL_ENERGYINC_5       19   /* callback function: InfoWin */
#define MAINPANEL_ENERGYEND_5       20   /* callback function: InfoWin */
#define MAINPANEL_ENERGYDWELL_5     21   /* callback function: InfoWin
*/
#define MAINPANEL_NSWEEPS           22   /* callback function: InfoWin */
#define MAINPANEL_RUNNUMBER         23   /* callback function: InfoWin */
#define MAINPANEL_GOBUTTON          24   /* callback function: Start */
#define MAINPANEL_QUITBUTTON        25   /* callback function: QuitFunction
*/
#define MAINPANEL_PHOTONENERGY      26   /* callback function: InfoWin */
#define MAINPANEL_INITGRAFSCALE     27
#define MAINPANEL_NUMBER_REGIONS    28   /* callback function: InfoWin
*/
#define MAINPANEL_PARAMETER_FILE    29 /* callback function: GetParamFile
*/
#define MAINPANEL_HVRANGE           30   /* callback function: InfoWin */
#define MAINPANEL_SPEC_SWITCH       31
```

```
#define  MAINPANEL_REGION1          32
#define  MAINPANEL_REGION2          33
#define  MAINPANEL_REGION4          34
#define  MAINPANEL_REGION5          35
#define  MAINPANEL_Stripe        36
#define  MAINPANEL_REGION3          37

   /* Menu Bars, Menus, and Menu Items: */
      /* (no menu bars in the resource file) */

   /* Callback Prototypes: */

int  GetParamFile(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int  GraphControl(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int  InfoWin(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int  QuitFunction(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int  Start(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);


/*************************************************************/
/*  File:  main.c - contains main driver                     */
/*                  and several functions:  ErrorRoutine, InttoStr,  */
/*                  DatetoStr and MakeFName   (8-13-96)      */
/*************************************************************/
#include <utility.h>
#include <dataacq.h>
#include <formatio.h>
#include <userint.h>
#include <ansi_c.h>
#include "takedata.h"
#include "prottype.h"
#include "takedat.h"
#include "struct.h"
#include "global.h"
#include <time.h>

/*************************************/
/*  Function:  Main                  */
```

```
/* Purpose: Main driver for the                    */
/*                program                          */
/*************************************/
main()
{
     int Status;
     Handle=LoadPanel(0,"takedat.uir",MAINPANEL);
     Status=Initialize();
     if(Status==TRUE)
     {
        DisplayPanel(Handle);
        RunUserInterface();
     }
     else
     {
        MessagePopup("Fatal Runtime Error",
        "Unable to load region\ninformation in MAIN.\n\nAborting program
execution...");
     }
}

/************************************************/
/*  Function: ErrorRoutine                      */
/*  Purpose: Pops up an error window            */
/************************************************/
void ErrorRoutine(function,error)
char *function;
int error;
{
  char temp[80];
  Fmt(temp,"%s<%s error = %i",function,error);
  MessagePopup("ERROR",temp);
}

/************************************************/
/*  Function: InttoStr                          */
/*  Purpose: 2 digit numbers to string (puts in 0 */
/*                in single digit numbers)      */
/************************************************/
void InttoStr(num, string)
int num;
char *string;
{
```

```
  if (num < 10)
                Fmt(string, "%s<0%d", num);
  else
                Fmt(string, "%s<%d", num);
}

/*************************************************/
/*  Function:  DatetoStr                         */
/*  Purpose:  Determine dates send back month and */
/*          and day as string and return year as  */
/*          an integer                            */
/*************************************************/
int DatetoStr(strmonth, strday)
char *strmonth, *strday;
{
                time_t thetime;
                struct tm date;

                thetime = time(0);
                date = *localtime(&thetime);  /* Gets date */

                /* Changes month and day to strings */
                InttoStr((date.tm_mon+1), strmonth);
                InttoStr((date.tm_mday), strday);
                return(date.tm_year);

} /*  End DatetoStr */

/*************************************************/
/*  Function: MakeFName                          */
/*  Purpose:  Forms data filenames               */
/*   FileNames have the following format now     */
/*              xx(month)xx(day)xx(runnumber)x(region).dat */
/*************************************************/
void MakeFName(filename, runnum, region)
char *filename;
int runnum, region ;
{
                char themonth[3], theday[3], run[3];
                DatetoStr(themonth, theday);
                InttoStr(runnum, run);
                Fmt(filename, "%s<%s%s%s%s%d.%s",DATA_PATH, themonth,
theday, run, region, DATA_LABEL);
```

```
} /* End MakeFName */

/*************************************************************/
/* File: infowin.c - contains two functions                  */
/* Functions: Infowin - Gets values from input window        */
/*            SetInfoWin - Sets values given from input window */
/*                                                           */
/* Changes: 6-21-95 LG & LSJ removed angle related code      */
/*          7-25-95 LSJ put in loop for getting info from    */
/*                  the window about regions (5 regions)     */
/*          8-11-95 fixed loop dep. to number of regions     */
/*          8-16-95 took out unnecessary variables           */
/*          8-16-95 added the setting and getting of the number */
/*                  or regions to the data structure         */
/*          8-22-95 added comments                           */
/*          10-13-95 Added lines necessary to add new feature */
/*                  on window which is selection of HVRange  */
/*************************************************************/
#include <ansi_c.h>
#include <userint.h>
#include <formatio.h>
#include <math.h>
#include "struct.h"
#include "takedata.h"
#include "takedat.h"
#include "prottype.h"
/*************************************************************/
/* Function InfoWin                                          */
/* Purpose Gets the information from the UserInterface and puts */
/*         it into the Structure                             */
/*************************************************************/
int InfoWin(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2)
{
   extern int Handle, KorB;
   extern TypicalRegion Region;
   extern unsigned int CurrentDataFileNumber;
   int number, count=0;

  GetCtrlVal(Handle,MAINPANEL_NUMBER_REGIONS,&Region.XPS.NumRegions);
   GetCtrlVal(Handle,MAINPANEL_NSWEEPS,&Region.XPS.NumberOfSweeps);
   GetCtrlVal(Handle,MAINPANEL_RUNNUMBER,&CurrentDataFileNumber);
```

```
  GetCtrlVal(Handle,MAINPANEL_SPEC_SWITCH,&KorB);
                /* ISS (KE)  or XPS (BE) */
  number = 1+OFFSET;
  do {
       GetCtrlVal(Handle,number++, &Region.XPS.EnergyBeg[count]);
       GetCtrlVal(Handle,number++, &Region.XPS.EnergyInc[count]);
       GetCtrlVal(Handle,number++, &Region.XPS.EnergyEnd[count]);
       GetCtrlVal(Handle,number++, &Region.XPS.EnergyDwell[count]);
       count++;
  } while (count < Region.XPS.NumRegions);
  GetCtrlVal(Handle,MAINPANEL_PHOTONENERGY,&Region.XPS.PhotonEnergy)
;
  GetCtrlVal(Handle,MAINPANEL_HVRANGE,&Region.XPS.HVRange);
  return(0);
} /* END InfoWin */

/*************************************************************/
/* 8-22-95  NEW FUNCTION ADDED allowing specification of parameter */
/*          file, calls initalize to set info                */
/* Function GetParamFile                                     */
/* Purpose  Gets the parameter file from the user interface  and */
/*        changed the information on the window accordingly  */
/*************************************************************/
int GetParamFile(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2)
{
  extern int Handle;
  extern char StatusFile[25];

  GetCtrlVal(Handle,MAINPANEL_PARAMETER_FILE,StatusFile);
  Initialize();
  return(0);
} /* END InfoWin */

/*************************************************************/
/* Function SetInfoWin                                       */
/* Purpose  Gets the information from the Structure and puts it */
/*        in into the user interface                         */
/*              changes 8-12-96:  Change to change all regions (zeros unused */
/*              regions)                                     */
/*************************************************************/
int SetInfoWin()
{
```

```
 extern TypicalRegion Region;
 extern int Handle, KorB;
 extern unsigned int CurrentDataFileNumber;
 int number, count=0;
 char string[25];

 SetCtrlVal(Handle,MAINPANEL_NSWEEPS,Region.XPS.NumberOfSweeps);
 SetCtrlVal(Handle,MAINPANEL_RUNNUMBER,CurrentDataFileNumber);
 SetCtrlVal(Handle,MAINPANEL_NUMBER_REGIONS.Region.XPS.NumRegions);
 SetCtrlVal(Handle,MAINPANEL_SPEC_SWITCH,KorB);
                          /* ISS (KE)  or XPS (BE) */
 number = 1+OFFSET;
 do {
      SetCtrlVal(Handle, number++, Region.XPS.EnergyBeg[count]);
      SetCtrlVal(Handle, number++, Region.XPS.EnergyInc[count]);
      SetCtrlVal(Handle, number++, Region.XPS.EnergyEnd[count]);
      SetCtrlVal(Handle, number++, Region.XPS.EnergyDwell[count]);
      count++;
 } while (count < MAX_REGIONS);
 /* NOTE this goes to MAX_REGIONS so the unused regions which */
 /* have been all set to zero after the parameter file is    */
  /*  read in will be zeroed out on the screen */

  SetCtrlVal(Handle,MAINPANEL_PHOTONENERGY,Region.XPS.PhotonEnergy);
            SetCtrlVal(Handle,MAINPANEL_HVRANGE,Region.XPS.HVRange);

  return(0);
} /*  End SetInfoWin */


/*************************************************************/
/*  File:  initial.c - contains 1 function                   */
/*  Function:  Initialize value to structure which holds all the info.  */
/*             It gathers all the info from the status.dat file  */
/*                                                          */
/*  Changes:  6-21-95  LG & LSJ removed angle related code  */
/*             6-22-95 LSJ adds loop for number of max regions  */
/*                        defined by constant MAX_REGIONS   */
/*             8-11-95 fixed loop to NumRegions instead of  */
/*                        MAX_REGIONS                       */
/*              8-16-95 removed unnecessary variables       */
/*             8-16-95 Now what it reads is as  follows,    */
/*                        Current data run number, number of regions, */
```

```
/*                              photon energy , region info                    */
/*                  10-13-95 Added HVRange in status files                     */
/*                  8-12-96 Added loop to zero out  unused regions             */
/*************************************************************/
#include <utility.h>
#include <userint.h>
#include <ansi_c.h>
#include <formatio.h>
#include "takedata.h"
#include "takedat.h"
#include "struct.h"
#include "prottype.h"

int Initialize(void)
{
   extern TypicalRegion Region;
   extern unsigned int CurrentDataFileNumber;
   extern int KorB;
   extern char StatusFile[25];
   int Status,FileHandle;
   long Size;
   int theyear, year, count=0;
   char *Buffer, filename[BUFFER_SIZE],themon[3], mon[3], theday[3], day[3];
   Buffer=(char *)calloc(BUFFER_SIZE,sizeof(char));
   Fmt(filename,"%s<%s%s", STATUS_PATH, StatusFile);
   Status=GetFileInfo(filename,&Size);
   if(Status==TRUE)
   {
      /*  figure out the date */
      theyear = DatetoStr(themon, theday);
      FileHandle=OpenFile(filename,1,0,1);
      if(FileHandle!=-1)
      {
         Status=ReadLine(FileHandle,Buffer,80);
         Status=ReadLine(FileHandle,Buffer,80);
         /* RUN NUMBER */
         Scan(Buffer,"%i %i%s%s", &CurrentDataFileNumber, &year, mon, day);
         if ((year != theyear) || (strcmp(mon,themon)) || (strcmp(day,theday)))
               CurrentDataFileNumber = 1;
         Status=ReadLine(FileHandle,Buffer,80);
         Status=ReadLine(FileHandle,Buffer,80);
         Scan(Buffer,"%s>%i", &Region.XPS.NumRegions); /* NUMBER REGIONS */
         Status=ReadLine(FileHandle,Buffer,80);
```

```
        Status=ReadLine(FileHandle,Buffer,80);
        Scan(Buffer,"%s>%f", &Region.XPS.PhotonEnergy);
                                /* PHOTON ENERGY */
        Status=ReadLine(FileHandle,Buffer,80);
        Status=ReadLine(FileHandle,Buffer,80);
        Scan(Buffer,"%s>%f", &Region.XPS.HVRange);  /* Range */
        Status=ReadLine(FileHandle,Buffer,80);
        Status=ReadLine(FileHandle,Buffer,80);
        Scan(Buffer,"%s>%i", &KorB);  /* ISS or XPS */

        do { /* PARAMETER FOR EACH NUMBER OF REGIONS */
             Status=ReadLine(FileHandle,Buffer,80);
             Status=ReadLine(FileHandle,Buffer,80);
             Scan(Buffer,"%s>%f%f%f%f%d",&Region.XPS.EnergyBeg[count],
                  &Region.XPS.EnergyInc[count], &Region.XPS.EnergyEnd[count],
                  &Region.XPS.EnergyDwell[count],&Region.XPS.NumberOfSweeps);
             count++;
        } while (count < Region.XPS.NumRegions);

        /* CLEAR out other unused Regions */
        while (count < MAX_REGIONS) {
              Region.XPS.EnergyBeg[count]=0;
              Region.XPS.EnergyInc[count]=0;
              Region.XPS.EnergyEnd[count]=0;
              Region.XPS.EnergyDwell[count]=0;
              count++;
        }
        CloseFile (FileHandle);
        SetInfoWin();
       } /* END IF FILE OPENS */
     } /* IF END FILE EXISTS */
  return(TRUE);
}

/*****************************************************************/
/* File getcount.c                                               */
/* Function GetCount                                             */
/* Input:  unsigned short  :  number of timebase periods to collect */
/*                    counts                                     */
/* Output: int : FALSE if everything went OK                     */
/*           ERR_BAD_CONFIGURE if problems configuring device    */
/*             NI-DAQ Error if problem occured during counting   */
/*                                                               */
```

```
/*  11/10/94 MAL Get number of counts in a counter for a certain          */
/*               time, specified by the number of periods of a            */
/*               gating counter running with a certain resolution         */
/*                                                                        */
/*  To be done:  Clarify line if ((Time2-Time1)>0.3) GateSt....           */
/**************************************************************/

#include <utility.h>
#include <dataacq.h>
#include <userint.h>
#include "struct.h"
#include "prottype.h"
#include "takedata.h"

int GetCount(unsigned short Dwell,unsigned short *Count)
{
    extern int ControlCode;
    extern int SweepStatus;
    int Status,RetVal;
    int Error=FALSE;
    short GateStillGoing,OverFlow;
    double Time1,Time2;
    /**************************************************************/
    /*  Reset counters DATA_COUNTER & PULSE_COUNTER                */
    /**************************************************************/
    Status=CTR_Reset(NI_DEVICE_NUMBER,DATA_COUNTER,1);
    if(Status!=FALSE) {
       ErrorRoutine("Resetting DATA_COUNTER",Status);
       Error=TRUE;
    }
    Status=CTR_Reset(NI_DEVICE_NUMBER,PULSE_COUNTER,1);
    if(Status!=FALSE) {
       ErrorRoutine("Resetting PULSE_COUNTER",Status);
       Error=TRUE;
    }
    /**************************************************************/
    /*  Configure the counter used to count pulses.                */
    /*  Gated by PULSE_COUNTER                                     */
    /**************************************************************/
    Status=CTR_Config(NI_DEVICE_NUMBER,DATA_COUNTER,0,1,0,0);
    if(Status!=FALSE) {
       ErrorRoutine("Configuring DATA_COUNTER",Status);
       Error=TRUE;
```

```
}
/*****************************************************/
/* Configure the counter used to gate the data-taking counter.     */
/*****************************************************/
Status=CTR_Config(NI_DEVICE_NUMBER,PULSE_COUNTER,0,0,0,0);
if(Status!=FALSE) {
   ErrorRoutine("Configuring PULSE_COUNTER",Status);
   Error=TRUE;
}
/*********************************************/
/* Did the counters configure OK?                          */
/*********************************************/
if(Error) {
   RetVal=ERR_BAD_CONFIGURE;
}else {
    /***********************************************************/
    /* Configure data-taking counter to count from DATA_SOURCE       */
    /***********************************************************/
    Status=CTR_EvCount(NI_DEVICE_NUMBER,
                DATA_COUNTER,DATA_SOURCE,0);
    if(Status!=FALSE) {
    ErrorRoutine("Setting up DATA_COUNTER",Status);
    Error=TRUE;
    }
    /***********************************************************/
    /* Configure gating counter with resolution PULSE_TIMEBASE       */
    /***********************************************************/
    Status=CTR_Pulse(NI_DEVICE_NUMBER,
          PULSE_COUNTER,PULSE_TIMEBASE, DELAYTIME, Dwell);
    if(Status!=FALSE) {
    ErrorRoutine("Setting up PULSE_COUNTER",Status);
    Error=TRUE;
    }
    /***********************************************************/
    /* Wait for data taking pulse to begin by waiting                  */
    /*                                                                 */
    /***********************************************************/
    Time1=Timer();
    do {
    /* Status=CTR_State(NI_DEVICE_NUMBER,
                PULSE_COUNTER, &GateStillGoing);                */
     Time2=Timer();
    /*}while((!GateStillGoing)&&((Time2-Time1)<MINTIME)); */
```

```
        }while((Time2-Time1)<MINTIME);
        /*********************************************************/
        /*  Wait for data taking to end, signalled by a return status (ERROR)
                */
        /*  or the TC signal from the gate going low                    */
        /*********************************************************/
        if(Error) {
                RetVal=ERR_BAD_CONFIGURE;
        } else {
        do{
           Status=CTR_State(NI_DEVICE_NUMBER,PULSE_COUNTER,
                                &GateStillGoing);
           ProcessSystemEvents();
           if(ControlCode==HALTBUTTON) {
              SweepStatus=HALTBUTTON;
           }
        }while((GateStillGoing)&&(SweepStatus!=HALTBUTTON));
        RetVal=Status;
        CTR_EvRead(NI_DEVICE_NUMBER,DATA_COUNTER,&OverFlow,Count);
      }
    }
    return(RetVal);
}
/*************************************************************/
/*  File:  doxpssw.c - contains 1 function                   */
/*  Function:  DoXPSSweep                                    */
/*  Changes 12-4-96:  Add slight modifcation for Thicken Expt */
/*                it calls SaveThickSweep for thicken experiments */
/*************************************************************/

#include "struct.h"
#include "prottype.h"
#include "takedata.h"
#include "takedat.h"
#include <math.h>
#include <userint.h>
#include <stdlib.h>

int DoXPSSweep(int EnergyRegionCounter)
{
    extern TypicalRegion Region;
    extern int CurrentRegion, KorB;
    extern double XMin,XMax;
```

```
extern int SweepStatus;
double CurrentEnergy, tempE;
unsigned int Index;
unsigned short Dwell,Count;
CurrentEnergy=Region.XPS.EnergyBeg[EnergyRegionCounter];
XMin=CurrentEnergy;
XMax=Region.XPS.EnergyEnd[EnergyRegionCounter];
if (KorB==XP) {
            XMin=-XMin;
            XMax=-XMax;
}
Dwell=(unsigned short) (Region.XPS.EnergyDwell[EnergyRegionCounter] /
                                        PULSE_CLOCK_PERIOD+.5);
Region.XPS.EnergyNumSteps[EnergyRegionCounter] =
            (int)fabs((Region.XPS.EnergyBeg[EnergyRegionCounter] -
            Region.XPS.EnergyEnd[EnergyRegionCounter]) /
            Region.XPS.EnergyInc[EnergyRegionCounter])+1;
SweepStatus= NORMAL;  /* CHANGED THIS TEMPORARILY */
for(Index=0;Index<Region.XPS.EnergyNumSteps[EnergyRegionCounter];Index++)
{
    SendXPSEnergy(CurrentEnergy);
    GetCount(Dwell,&Count);
    if(SweepStatus==HALTBUTTON) {
            break;
    }
    /* tempE is used because if taking XPS (BE) must send the */
    /*  negative value of the current energy */
    if (KorB==XP) tempE = -CurrentEnergy;
    else tempE = CurrentEnergy;
    NewXPSPoint(EnergyRegionCounter,Index,tempE,Count);
    if (KorB==XP)
            CurrentEnergy -= Region.XPS.EnergyInc[EnergyRegionCounter];
    else
            CurrentEnergy += Region.XPS.EnergyInc[EnergyRegionCounter];
}
if (!SweepStatus) {
   if (KorB < TH)
      SaveXPSSweep(EnergyRegionCounter,
            Region.XPS.EnergyNumSteps[EnergyRegionCounter]);
   else
      SaveThickSweep(EnergyRegionCounter,
            Region.XPS.EnergyNumSteps[EnergyRegionCounter]);
}
```

```
    return(0);
}
/*************************************************************/
/* File: grafctrl.c - contains one function                  */
/* Function: GraphControl                                    */
/*************************************************************/
#include "takedat.h"
#include "takedata.h"
#include <userint.h>
#include <utility.h>

int  GraphControl(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2)
{
    extern int ControlCode;

    if (event == EVENT_COMMIT)
    {
       switch (control)
       {
          case GRAPHPANEL_CLOSEBUTTON :
                ControlCode=CLOSEBUTTON;
                Beep();
                RemovePopup(0);
                break;
          case GRAPHPANEL_HALTBUTTON :
                ControlCode=HALTBUTTON;
                Beep();
                break;
          default :
                ControlCode=NORMAL;
                break;
       }
    }
    return(0);
 } /* End GraphControl */

/*************************************************************/
/* File: xpsout.c - contains 1 function                      */
/* Function SendXPSEnergy                                    */
/* Input: double : kinetic energy to be examined             */
/* Output: double : voltage to be output                     */
/* Error: double : ERR_OUT_OF_RANGE energy out of range      */
```

```
/*                                                                    */
/*  Send an appropriate voltage to the D/A converter for the energy requested */
/*  11/9/94 MAL                                                       */
/*  10/13/95 Changed constant for HV Range with value stored in       */
/*           structure which the interface dictates                   */
/*  8/5/96  Energy will vary with type of spectroscopy                */
/*********************************************************************/

#include "takedata.h"
#include "struct.h"
#include <dataacq.h>
extern TypicalRegion Region;
extern int CurrentRegion, KorB;
int SendXPSEnergy(double Energy)
{
    double OutputVoltage,KineticEnergy;
    int Status;

    if (KorB == XP)
            KineticEnergy=Region.XPS.PhotonEnergy-Energy;
    else
            KineticEnergy = Energy;
    OutputVoltage=(KineticEnergy/Region.XPS.HVRange)*MAX_AO_VOLTS;
    Status=AO_VWrite(NI_DEVICE_NUMBER,XPS_AO_CHAN,OutputVoltage);
    return(Status);
}

/*********************************************************************/
/*  File:  SaveXPS.c - contains five functions
            */
/*  Functions:  SaveXPSSweep                                          */
/*              NewXPSPoint                                           */
/*              SaveXPSData                                           */
/*              SaveXPSInfo                                           */
/*              SaveStatus                                            */
/*                                                                    */
/*  Changes:  6-21-95  LG & LSJ removed angle related code.           */
/*              8-11-95  Loops Fixed                                  */
/*              Removed Extraneous declarations and other things we   */
/*              did not understand or see relevence of.  Now it works.*/
/*              Combined SaveXPSData and SaveXPSInfo                  */
/*              into SaveXPSData.                                     */
/*              8-16-95  Changes to elimate extra datastructures      *
```

```
/*              to help elimate stack over flow                          */
/*              Change the way data files are numbered now regions are   */
/*              saved with the same file number but with different _#    */
/*              numbers                                                  */
/*              Took out number defining buffer size use                 */
/*               constants defined in takedata.h instead                 */
/*              Save status now also saves the number of regions         */
/*              8-17-95  Changed save to save with month-day-            */
/*               filenumber_subnumber format                             */
/*              Changed SaveXPSSweep so it won't redraw graph if it      */
/*              doesn't need to do it                                    */
/*              8-21-95  Changed .dat to a defined string, cleaned up some string */
/*              commands                                                 */
/*              8-22-95  Add function comment headers and comments       */
/*              9-95     changed so status files can have different  names */
/*              0-13-95  HV Range was added to the status files          */
/*              Header file changed also to say energy step              */
/*              instead of energy dwell                                  */
/*              2-15-96 Changes made to data structure require that functions be */
/*              sent the region number as the data become more           */
/*              "dimensional"                                            */
/*              2-20-96 Changed how the data file names are formated     */
/*              8-5-96  Took out all date handling, now in function DatetoStr in */
/*              main.c                                                   */
/*              A Changed titles in saved files from XPS to generic regions*/
/*              Added save spectra type in files                         */
/*              Took out graph redrawing out of SaveXPSSweep             */
/*              and put it in start.c instead                            */
/*              also included spectroscopy type in data files            */
/*              8-13-96 Datafilenumber determined in function called     */
/*              MakeFName located in main.c                              */
/************************************************************/
#include "struct.h"
#include "takedat.h"
#include "takedata.h"
#include "prottype.h"
#include <userint.h>
#include <formatio.h>
#include <ansi_c.h>
#include <analysis.h>
#include <string.h>
/************************************************************/
/* Function: SaveXPSSweep                                   */
```

```
/* Purpose:  Sums up the total cps from a sweep into XPSData          */
/*****************************************************************/
int SaveXPSSweep(int EnergyRegionCounter, unsigned int NumSteps)
{
    extern double CurrentSweep[MAX_XPS_POINTS];
    extern double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
   extern double XPSData[MAX_REGIONS][MAX_XPS_POINTS];
   extern unsigned int CurrentNumberOfSweeps;
   unsigned int Counter;

   for (Counter=0;Counter<=NumSteps;Counter++)
   {
      XPSData[EnergyRegionCounter][Counter]+=CurrentSweep[Counter];
      CurrentSweep[Counter] = XPSData[EnergyRegionCounter][Counter] /
            (double)CurrentNumberOfSweeps;
   }
   return(0);
} /* End SaveSweep */

/*****************************************************************/
/* Function: NewXPSPoint                                         */
/* Purpose:  Get the next data point                             */
/*****************************************************************/
int NewXPSPoint(int EnergyRegionCounter,unsigned int Index,double
CurrentEnergy,unsigned short Count)
{
   extern TypicalRegion Region;
   extern double CurrentSweep[MAX_XPS_POINTS];
   extern double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
   extern int GraphHandle;
   extern double XMin,XMax,YMin,YMax;
   double NormalizedCount;
   int AxisChanged;

   NormalizedCount =
            (double)Count/(Region.XPS.EnergyDwell[EnergyRegionCounter]);
   AxisChanged=FALSE;
   if(NormalizedCount<YMin)
   {
      YMin=NormalizedCount*0.95;
      AxisChanged=TRUE;
   }
   if(NormalizedCount>YMax)
```

```
  {
    YMax=NormalizedCount*1.05;
    AxisChanged=TRUE;
  }
  if(AxisChanged)
    SetAxisRange(GraphHandle,GRAPHPANEL_GRAPH,VAL_MANUAL,
                 XMin,XMax,VAL_MANUAL,YMin,YMax);
  PlotPoint(GraphHandle,GRAPHPANEL_GRAPH,CurrentEnergy,
            NormalizedCount,VAL_SMALL_EMPTY_SQUARE,VAL_RED);
            CurrentE[EnergyRegionCounter][Index]=CurrentEnergy;
  CurrentSweep[Index]=NormalizedCount;
  return(0);
} /* End NewPoint */

/*************************************************************/
/* Function: SaveXPSData                                     */
/* Purpose:  Saves the XPS data into a file that is specified by */
/*           the datafile location, date and run/subregion number */
/*           A header contining that info is also stored in the file */
/*************************************************************/
int SaveXPSData(int EnergyRegionCounter, int subnumber)
{
    extern TypicalRegion Region;
    extern double XPSData[MAX_REGIONS][MAX_XPS_POINTS];
    extern double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
    extern unsigned int CurrentDataFileNumber;
    extern char DataFileLocation[BUFFER_SIZE];
    extern int KorB;
    unsigned int Counter;
    char FileName[BUFFER_SIZE],Buffer[BUFFER_SIZE], themonth[3], theday[3];
    int DataFileHandle, theyear;

    MakeFName(FileName, CurrentDataFileNumber, subnumber);
    theyear = DatetoStr(themonth, theday);
    DataFileHandle=OpenFile(FileName,2,0,1);
    /**********************/
    /* Saving the file header  */
    /**********************/
    if (KorB) Fmt(Buffer,"ISS Region  %dof%d", subnumber,
                         Region.XPS.NumRegions);
    if (!KorB) Fmt(Buffer,"XPS Region  %dof%d", subnumber,
                         Region.XPS.NumRegions);
    WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
```

```
   Fmt(Buffer,"%s<%s-%s-%d:  Run Number:  %d -- %d%s",(themonth),
        theday, theyear, CurrentDataFileNumber,
        Region.XPS.NumberOfSweeps," sweeps");
   WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s<%s%f eV    %s%f","Photon Energy: ",
        Region.XPS.PhotonEnergy,"Step Energy: ",
        Region.XPS.EnergyDwell[EnergyRegionCounter]);
   WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s<Beginning Energy: %f Ending Energy: %f Energy Inc:  %f",
        Region.XPS.EnergyBeg[EnergyRegionCounter],
        Region.XPS.EnergyEnd[EnergyRegionCounter],
        Region.XPS.EnergyInc[EnergyRegionCounter]);
   WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"eV\tcps");
   WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
   /*********************************/
   /* Saving the data                */
   /* File format month_day_run#_region   */
   /*********************************/
   for(Counter=0;Counter <
            Region.XPS.EnergyNumSteps[EnergyRegionCounter];Counter++)
   {
      if (KorB==0)
            Fmt(Buffer,"%s<%f\t%f", (Region.XPS.PhotonEnergy +
                 CurrentE[EnergyRegionCounter][Counter]),
                 XPSData[EnergyRegionCounter][Counter] /
                 Region.XPS.NumberOfSweeps);
      else
            Fmt(Buffer,"%s<%f\t%f", (CurrentE[EnergyRegionCounter][Counter]),
                    XPSData[EnergyRegionCounter][Counter] /
                    Region.XPS.NumberOfSweeps);
      WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
   }
   CloseFile(DataFileHandle);
   return(0);
} /*  End SaveXPSData */

/*************************************************************/
/*  Function:  SaveStatus                                    */
/*  Purpose:   Saves the last set of parameters entered for  */
/*              reference.                                   */
/*************************************************************/
int SaveStatus()
```

```
{
   extern TypicalRegion Region;
   extern unsigned int CurrentDataFileNumber;
   extern int KorB;
   extern char StatusFile[25];
   char Buffer[BUFFER_SIZE], string[50], FileName[BUFFER_SIZE],
   char themonth[3], theday[3];
   int StatusFileHandle, theyear, count=0;
   /*  figure out the date */
   theyear = DatetoStr(themonth, theday);
   Fmt(FileName,"%s<%s%s", STATUS_PATH, StatusFile);
   StatusFileHandle=OpenFile(FileName,2,0,1);
   Fmt(Buffer,"%s","Current Data File Number year mon day");
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);  /* RUN NUMBER */
   Fmt(Buffer,"%s<%i %i %s %s",CurrentDataFileNumber, theyear, themonth, theday);
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s","Number of Regions");   /* NUMBER OF REGIONS */
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s<%i",Region.XPS.NumRegions);
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s","Photon Energy");         /* PHOTON ENERGY */
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s<%f",Region.XPS.PhotonEnergy);
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s","Multiplier");
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1); /* Multiplier */
   Fmt(Buffer,"%s<%f",Region.XPS.HVRange);
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s","Spectroscopy (0=XPS, 1=ISS, 2=AES)");
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
   Fmt(Buffer,"%s<%i", KorB);
   WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);

   do { /* SAVES THE PARAMETERS FOR EACH NUMBER OF REGIONS */
        sprintf(string, "Region #%d Information", count+1);
        Fmt(Buffer,"%s",string);
        WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
        Fmt(Buffer,"%s<%f %f %f %f %i",Region.XPS.EnergyBeg[count],
                Region.XPS.EnergyInc[count],Region.XPS.EnergyEnd[count],
                Region.XPS.EnergyDwell[count],Region.XPS.NumberOfSweeps);
        WriteLine(StatusFileHandle,Buffer,strlen(Buffer)+1);
        count++;
    } while (count < Region.XPS.NumRegions);
```

```
  CloseFile(StatusFileHandle);
  return(0);
} /* END SaveStatus */

/*************************************************************/
/* File Thick.c                                              */
/* Purpose: Contains special Thicken functions               */
/* Created 12/5/96          LSJ                              */
/* Contains: ChkThStps                                       */
/*           SaveThickSweep                                  */
/*           SaveThickData                                   */
/*           */
/* Description: Thicken is basically ISS but only a small number of  */
/*           of data points which "focus" on a peak a followed for a  */
/*           a number of sweeps. There points are averaged and then   */
/*           tracked over time to study the degradation of that ISS   */
/*           peak. The data for all regions are saved in one file     */
/*           Where the Sweep number is the first datacolumn and each  */
/*           subsequent column is the average counts per second for   */
/*           each region identified by its begining energy (<cps>(BE))  */
/*************************************************************/
#include <formatio.h>
#include <ansi_c.h>
#include "prottype.h"
#include "struct.h"
#include "takedat.h"
#include "takedata.h"
/*************************************************************************/
/* Function: ChkThStps                                                   */
/* Purpose: Make sure Thicken user doesn't exceed max steps to do expt in good time
*/
/*************************************************************************/
int ChkThStps(Status)
int Status;
{
    extern TypicalRegion Region;
    char Buffer[50];
    int counter=0, check=1;

    do {
        if (((int)fabs((Region.XPS.EnergyBeg[counter] -
            Region.XPS.EnergyEnd[counter]) /
            Region.XPS.EnergyInc[counter])+1) > MaxThSteps)
```

```
        check--;
        counter++;
    } while ((counter < Region.XPS.NumRegions) && (check));
    if (!check) {
       Fmt(Buffer,"[Reg%d]%s\n", counter, BAD_THNUM);
       ErrorRoutine(Buffer,Status);
    }
    return(check);
} /* End ChkThStps*/

/*************************************************************/
/* Function: SaveThickSweep                                  */
/* Purpose:  Averages up the total cps from a sweep into XPSData */
/*************************************************************/
int SaveThickSweep(int EnergyRegionCounter, unsigned int NumSteps)
{
    extern double CurrentSweep[MAX_XPS_POINTS];
    extern double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
    extern double XPSData[MAX_REGIONS][MAX_XPS_POINTS];
    extern unsigned int CurrentNumberOfSweeps;
    unsigned int Counter;
    double temp = 0;

    for(Counter=0;Counter<=NumSteps;Counter++)
    {
        temp += CurrentSweep[Counter];
    }
    XPSData[EnergyRegionCounter][(CurrentNumberOfSweeps-1)] = temp/NumSteps;

    return(0);
} /* End SaveSweep */

/*************************************************************/
/* Function: SaveThickData                                   */
/* Purpose:  Saves the Thicken data into a file that is specified by */
/*           the datafile location, date and run/subregion number */
/*           A header contining the info for each region info */
/*           is also stored.  Each region data saved as a column */
/*************************************************************/
int SaveThickData()
{
    extern TypicalRegion Region;
```

```
extern double XPSData[MAX_REGIONS][MAX_XPS_POINTS];
extern double CurrentE[MAX_REGIONS][MAX_XPS_POINTS];
extern unsigned int CurrentDataFileNumber;
extern char DataFileLocation[BUFFER_SIZE];
unsigned int Counter, Columns;
char FileName[BUFFER_SIZE],Buffer[BUFFER_SIZE], themonth[3], theday[3];
char tempstr[BUFFER_SIZE];
int DataFileHandle, theyear, big;

MakeFName(FileName, CurrentDataFileNumber, 1);
theyear = DatetoStr(themonth, theday);
DataFileHandle=OpenFile(FileName,2,0,1);
/*************************/
/* Saving the file header */
/*************************/
Fmt(Buffer,"Thickness Data");
WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
Fmt(Buffer,"%s<%s-%s-%d:  Run Number:  %d with %d%s", (themonth),
      theday, theyear, CurrentDataFileNumber, Region.XPS.NumberOfSweeps,
      "sweeps");
WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
Fmt(Buffer, "Photon Energy - %f ev", Region.XPS.PhotonEnergy);
WriteLine(DataFileHandle, Buffer, strlen(Buffer)+1);
for (Counter=0; Counter<Region.XPS.NumRegions; Counter++){
     Fmt(Buffer,"%s<Beginning Energy: %f Ending Energy: %f",
           Region.XPS.EnergyBeg[Counter], Region.XPS.EnergyEnd[Counter]);
     WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
     Fmt(Buffer,"%s<Step Energy:  %f, Energy Inc:  %f",
           Region.XPS.EnergyDwell[Counter], Region.XPS.EnergyInc[Counter]);
     WriteLine(DataFileHandle,Buffer,strlen(Buffer)+1);
     Fmt(Buffer, "-----------------------------------");
     WriteLine(DataFileHandle, Buffer, strlen(Buffer)+1);
}
/*******************/
/* Saving the data */
/*******************/
Fmt(Buffer, "Sweep\t");
for(Counter=0;Counter<Region.XPS.NumRegions;Counter++){
    /* type each column header */
    sprintf(tempstr, "<CPS>(%d)\t", (int)Region.XPS.EnergyBeg[Counter]);
    strcat(Buffer, tempstr);
}
WriteLine(DataFileHandle, Buffer, strlen(Buffer)+1);
```

```
    for(Counter=0;Counter<Region.XPS.NumberOfSweeps;Counter++) {
        Fmt(Buffer, "%d\t", (Counter+1));
        for (Columns=0; Columns<Region.XPS.NumRegions; Columns++) {
            sprintf(tempstr, "%f\t", XPSData[Columns][Counter]);
            strcat(Buffer, tempstr);
        }
        WriteLine(DataFileHandle, Buffer, strlen(Buffer)+1);
    }
    CloseFile(DataFileHandle);
    return(0);
} /*  End SaveThickData */

/************************************************/
/*  File:  quit.c - contains QuitFunction                    */
/************************************************/
#include <userint.h>
#include <utility.h>
#include <dataacq.h>
#include "struct.h"
#include "prottype.h"
#include "takedata.h"

int QuitFunction(int panel, int control, int event, void *callbackData,
                                 int eventData1, int eventData2)
{
    if (event==EVENT_COMMIT)
    {
        Beep();
        if(ConfirmPopup("","Do you really want to quit?"))
        {
            SaveStatus();
            /* Set AO output voltage to zero */
            AO_VWrite(NI_DEVICE_NUMBER,XPS_AO_CHAN,0.0);
            QuitUserInterface(0);
        }
    }
    return(0);
 }
```

VITA

LINDA SUNA. JUNG

University of Washington

1999

Linda Jung was born on December 26, 1970 in Korea. She received a BS in Biochemistry / Chemistry with a minor in Computer Science and Engineering - Software from the University of California, San Diego in 1993. She completed the requirements for the a Ph.D. in Physical Chemistry from the University of Washington, Seattle in 1999.