Robotic Target Following Under Dynamic and Collaborative Vision Feedback

Hui Xiao

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

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Xu Chen, Chair

Santosh Devasia

Sawyer Fuller

Program Authorized to Offer Degree:

Mechanical Engineering

University of Washington

Abstract

Robotic Target Following Under Dynamic and
Collaborative Vision Feedback

Hui Xiao

Chair of the Supervisory Committee:
Xu Chen
Department of Mechanical Engineering

Following rapidly and precisely a moving target has become the core functionality in robotic systems for transportation, manufacturing, and surgery. Among existing target following methods, vision-based tracking continues to thrive as one of the most popular and is the closest method to human perception. However, the slow sampling speed of vision sensors, the time delays of visual outputs, and the irregular processing time of image data fundamentally hinder real-time applications. Such limitations from the visual sensing dynamics need special attention when following fast-moving targets. Ignoring the visual sensing dynamics has caused unstable and even unsafe robot behavior.

This dissertation provides fast robotic target following considering the sensing dynamics. Starting from the fundamental problem in disturbance compensation where the sampling rate of the feedback sensor is insufficient to capture the high-frequency dynamics of the disturbance signal, we propose an information recovering algorithm to reconstruct fast-sampled signals from slowly-sampled ones. The reconstruction is made possible by re-parameterizing the dense autoregressive models of signals to a sparse structure. To reduce the recovering error at lower signal-to-noise ratios, we improve from the new autoregressive model to a framework using infinite impulse response (IIR) filter structures. Furthermore, the idea of signal recovering is extended to a collaborative sensing system, where two sensors—each sam-

pling at a different rates—collaborate to provide fast signal recovering significantly beyond the Nyquist frequency of individual sensors.

The information recovering algorithm has been engineered into an extended visual servoing control system for a high-speed target following. The extended visual servoing compensates for the following error caused by target motions by estimating the target velocities online, while the recovering algorithm compensates for the slow-sampling and time delays. Under a more challenging scenario where there are multiple feedback signal flows with irregular sampling intervals and variable delays, we propose a memory-enabled auto-restart Kalman filter (M-ARKF) to compensate for the full sensing dynamics, and additionally handle any jumping velocities of the tracked object. Simulation and experimentation on a dual-arm robotic manipulator and a robotic air-hockey player validate the proposed algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Prof. Xu Chen, for the countless hours in class and office to teach me knowledge and inspire my research ideas, for his trust, support, encouragement, and guidance to all my projects, as well as for his sincere care of my life and family. I would like to thank my committee members Prof. Santosh Devasia, Prof. Sawyer Fuller, Prof. Jeffrey Lipton and Prof. Byron Boots for all their help and advice with my Ph.D. study. I also wish to thank Prof. Yaakov Bar-Shalom for his valuable inputs to my research papers.

I also had great pleasure of working with the talented students in MACS lab: Tianyu Jiang, Dan Wang, Mingyu Wang, Mengying Leng and Guowei Hua. My appreciation also goes out to my family Weixiang Zeng, Jianxue Xiao and Qianxia Jiang for their support through my Ph.D years.

NOMENCLATURE

$\{A\}$ A 3D coordinate system with origin at porint A.

AR(n) An autoregressive model with an order of n.

${}^{A}\xi_{B}$ The 3D pose of frame $\{B\}$ with respect to frame $\{A\}$.

${}^{A}\boldsymbol{v}_{B}$ The 3D velocity of frame $\{B\}$ with respect to frame $\{A\}$.

$\oplus$ The composition operator of relative poses, e.g., ${}^{A}\xi_{C} = {}^{A}\xi_{B} \oplus {}^{B}\xi_{C}$.

${}^{A}\boldsymbol{R}_{B}$ The rotation matrix corresponding to the relative pose ${}^{A}\xi_{B}$.

${}^{A}\boldsymbol{t}_{B}$ The translation vector corresponding to the relative pose ${}^{A}\xi_{B}$.

$\boldsymbol{R}_{x}(\theta)$ The rotation matrix that corresponds to the 3D rotating of θ degrees about x axis.

$\boldsymbol{A}'$ The transpose of matrix $\boldsymbol{A}$.

LCM(M, N) The least common multiple of integers M and N.

$t\{x[n]\}$ The timestamp of signal $x[n]$. If $x[n]$ is uniformly sampled with period T, then $t\{x[n]\} = nT$.

$x_1[n] \leftrightarrow x_2[n]$ Two sequences $x_1[n]$ and $x_2[n]$ are equal and aligned in time (i.e.,$x_1[n] = x_2[n]$, $x_1[n]$ and $x_2[n]$ have the same timestamps for any n).

$\lceil x \rceil$ The ceiling function that maps a real number x to the smallest following integer.

$\mathrm{mod}(a, b)$ he remainder after division of a by b.

$\boldsymbol{A}^{\dagger}$ The Moore–Penrose inverse of matrix $\boldsymbol{A}$

Chapter 1

INTRODUCTION

1.1 Problem introduction

This dissertation considers robotic target following of a dynamic target based on visual feedback. Figure 1.1 shows two basic configurations of vision-based target following problem. Here, the goal is to control the relative 3D pose between the dynamic target and robot (noted as ${}^{E}\xi_{T}(t)$) such that it can converge to a desired profile ${}^{E}\xi_{T}{}^{*}(t)$. This desired profile could be as simple as a constant pose, or some designed motion trajectories to interact with the target object. The robot could be static like a fixed manipulator, a mobile robot like drones of cars, or the combination of both. The eye-in-hand configuration fixes cameras on the robot so the camera movement is synced with the robot motion. This configuration is common in mobile robots and manipulators where a dynamic view of the target is needed. The other eye-to-hand configuration controls the robot motion based on feedback from a static camera.

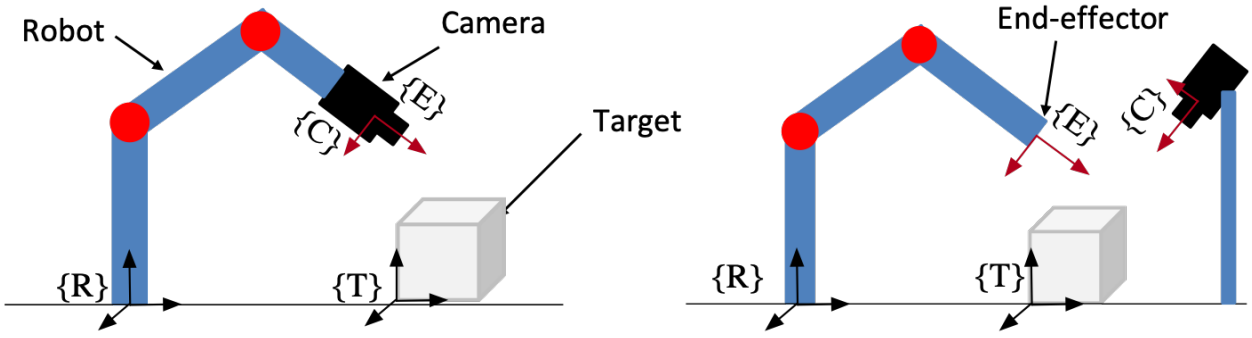


Figure 1.1: Robotic target following problem. Left: eye-in-hand configuration. Right: eye-to-hand configuration

Different from the traditional look-then-move strategy, vision-based target following uses real-time visual feedback to compute adaptive motion commands at every control steps. Such an approach consists of an inner kinematic loop for low-level actuator control and an outer visual servoing loop for calculating the goals (e.g., position, velocity and force) to be reached by robot actuators. The major barriers for implementing visual servoing control for following moving targets include the slow sampling speed, the time delays, and the irregular sampling time of vision sensors. We refer to these three fundamental limitations as visual sensing dynamics. The slow sampling speed is inherited from 1) the camera's relatively low image acquisition rate (e.g., 60 frames per second for a modern industrial camera) and 2) the complex image processing pipeline (e.g., target localization, 3D pose estimation, etc.). The time delays are mainly caused by hardware limitations of the data transmission bandwidth and the time cost for image processing. The irregular sampling intervals come from the variable image processing time (e.g, many iterative methods consume variable time depending on the convergence rates), image drops during acquisition, and temporary target loss due to image blur or objects moving out of the angle of view. These visual sensing dynamics, if not properly compensated in the control loop, will downgrade the target following accuracy or even cause control loop failures and unsafe robot behaviors when the target is moving rapidly. In this dissertation, we will focus on the robotic target following under visual sensing dynamics. The objective is to provide systematic methods of higher target following control bandwidth by compensating the effect of slow sampling, time delays, and variable sampling intervals.

1.2 Applications

The capability of following a moving target relates to high-impact robotic applications ranging from autonomous ground or aerial vehicles that follow a leading target, to surgical robot arms that track the motions of human organs, and to robotic manipulators that perform pick-and-place tasks in a highly dynamic environment (e.g., above the sea or in a turbulence airflow). A more detailed example is robotized surgery systems as shown in Figure 1.2. Here,

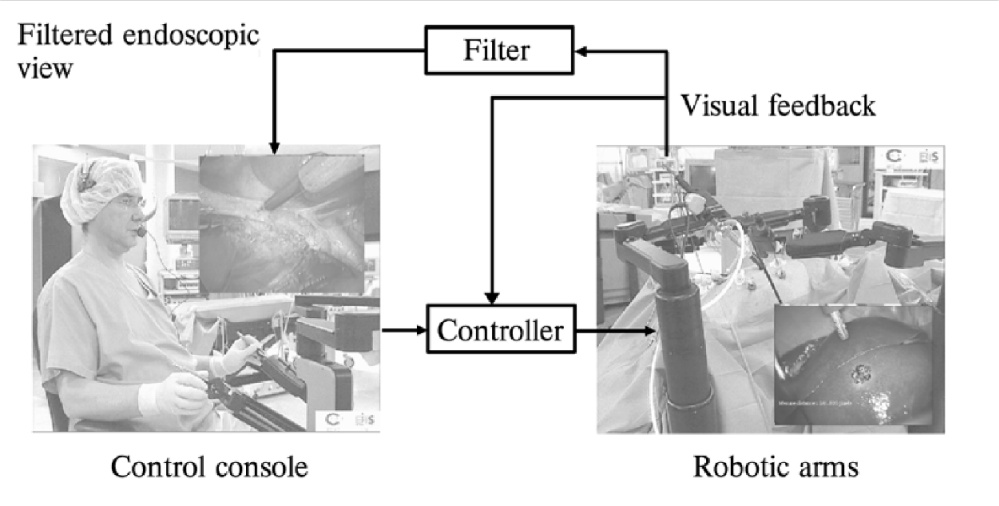


Figure 1.2: System overview of a robotized surgery system[1].

the doctor teleoperates the robot for surgery. The robot needs to actively adjust its position to compensate the physiological motion of the human's organs, such that the doctor can feel that he is operating on a scene that is almost still. The visual sensing dynamics is significant because the target (i.e., organs) movement is fast compared with the slow sampling speed of camera. As a consequence, slowly updated visual feedback and sensor delays would downgrade the target following accuracy and cause dangerous failures.

The proposed work of compensating visual sensing dynamics can be extended to border applications beyond the robot target following. For example, a fundamental problem in feedback control is when the system is subject to fast disturbances but can only get slowly updated sensor feedback. A single-rate high-gain control [2] is incapable to compensate for the error caused by such disturbance. Such difficulties occur to selective laser sintering, an additive manufacturing process that employs galvo scanners to steer high-power laser beams and relies on non-contact, slow sensing such as visual feedback to enhance the product quality. Combined with exiting disturbance compensation algorithms, the visual sensing compensation extend the algorithm's capability to disturbances beyond the Nyquist frequency of the sensor.

1.3 Related research

Visual servo control

In typical vision-based target following, image processing first extracts useful information from raw pixel data, then visual servo control algorithms calculate the motion commands of the robot. From a controls perspective, the goal of visual servoing is to minimize the errors between the desired and measured visual features. Based on how the visual features are defined, model-based visual servoing approaches can be classified into position-based and image-based [3, 4]. The position-based visual servoing (PBVS) defines visual features in the 3D space while the image-based visual servoing (IBVS) defines visual features in the 2D image space [3]. There are also extended approaches that use more sophisticated visual features [5, 6], or deploy a hybrid system [7, 8, 9] that combines advantages of IBVS and PBVS while trying to avoid their shortcomings. There are also recent learning-based approaches directly train neural network models to predict the control commands [10, 11].

The above mentioned approaches are based on the assumption of a motionless target. A moving target, however, demands additional and careful considerations in tracking. One design is to add an integral term in the velocity control law to compensate the object-following error [12]—effective only if the target moves at a constant velocity. A more commonly used approach is to estimate the target velocity by Kalman filters [13, 14, 15, 16], extended Kalman filters [17] or other filtering techniques [18]. Then one can compute the robot motion commands, assuming the target is moving at the estimated velocity.

Hardware approaches to compensate the sensing dynamics

Many works for tackling the barriers of visual sensing dynamics have been focused on developing advanced imaging hardware or communication protocols [19, 20, 21] that can provide fast and timely image data. There are expensive high-speed cameras in the market that can acquire images at 1000 frames per seconds. Huang [22] developed a high-performance robotic manipulation system based on high-speed visual feedback updated at 1KHz. The

image from the camera is gray-scale with a low pixel resolution of 512×512. The developed image processing can obtain one-dimensional target positions at the same sampling rate. The most recent work [23] has shown improved image resolution (1024×768) at the same sampling speed. Note that both works only applied simple image processing such that the existing hardware can process one image frame under a millisecond. Most state-of-the-art computer vision algorithms (e.g., convolution neural networks), however, require much more computation time. There are proposed frameworks [24] to skip part of the 1kHz updated images prior to the convolution neural networks, so the sampling rate of the final data flows is reduced. The main issue to consider when adopting a high-speed camera is the significantly increased computation cost and budget cost. Another issue for the high-speed camera is the requirement for a good lighting condition for the target object, due to the reduced exposure time. As a consequence, a high-speed camera most likely need a larger aperture lens and additional lighting equipment.

There are other fundamentally different cameras used in vision-based robot control. One is called position sensitive detector (PSD camera) [25] that can convert an incident light spot directly to analog signals. Reference [25] reported a target tracking system at 500 Hz based on PSD camera. Another special type of camera is called dynamic vision sensors, or event-based cameras [26]. This type of sensor is fundamentally different in the way that, instead of giving the intensities for each pixel, it only measures the brightness change for each pixel. The brightness change (either increase or decrease) for a pixel is called an event. The event-based camera is getting increasingly popular in the robotics field, in view of its feature of high temporal resolution, high dynamic range, and low latency. On the other hand, an event-based camera is still very expensive and has limited resolutions. Besides, due to the fundamental data format, the event-based camera is incapable of detecting a static or nearly static scene.

Software approaches to compensate the sensing dynamics

From an algorithm point of view, reconstruction of intersampling information is the key to enable fast dynamic controls under slow sensing. Certainly, recovering an arbitrary random signal is impossible. The pioneering work of Shannon and its extensions [27, 28] have shown that in order for the original analog information to be fully recovered from its samples, (i) the analog signal must be perfectly band-limited below the Nyquist frequency, and (ii) an ideal lowpass filter (acausal and not interpretable using a transfer function) is availale in the reconstruction process. Many reconstruction methods exist to approximate the original information-rich signal based on different assumptions [28]. For example, [29] formulates the problem to a maximum error minimization problem and utilizes sampled-data H_∞ control theory to find the best approximation under the H_∞ norm. Assuming a fast system model is given and the integrative sensor dynamics are available, [30] estimates intersample states of motion from slow and blurred images. Most literature focus on signal reconstruction under a feedforward design scheme. In a feedback closed-loop scenario, however, a greater challenge arises, as approximation errors may be amplified after going through the closed-loop dynamics, and major feedforward techniques are not applicable here.

In the track of vision-based robotics control research, visual sensing dynamics compensation (VSDC) [31, 32] was developed and tested to be successful in target tracking experiments [33] under slow-sampled sensing with delays. It formulated a customized dual-rate Kalman filter where the prediction runs at a faster rate and the correction step runs at a slower rate.

1.4 Contribution

In this dissertation, we propose a software and hardware solution to compensate sensing dynamics and the integration to robotic target following and target tracking applications. The contributions of the dissertation are:

Real-time recovery of fast data flows from sparse measurements

We first approach to tackle the problem of slow-sampling speed by developing an information recovery algorithm to reconstruct a fast-sampled data flow from the slow-sampled (sparse) data [34]. The recovery of intersample data is based on a re-parameterized autoregressive model of the fast signal. Furthermore, if the signal is narrow-band in the frequency domain, we developed an improved design of information recovery that has the additional capability to filter out the noise in the slow measurement [35]. In the case that there are two sensors, we designed and modeled the collaborative sensing configurations. A more complicated version of information recovery can recover the fast signal by combining data from both sensors and predicting the missing intersample data [36]. All the above information recovery algorithms have been embedded in the disturbance rejection control loop. The combined controller, called multirate forward-model disturbance observer, can rejects system disturbance at the fast sampling rate, even if the disturbances have beyond-Nyquist components.

Robotic target following under slow and delayed sensing

The conventional visual servo control [3] assumes a static target. If the target is moving, significant following error grows as the target speed increases. We have proposed a modified visual servo control algorithm [37] that can compensate for the following error caused by the target motion. The compensation relies on the target position and velocity obtained from Kalman filters. To additionally compensate for the slow and delayed sensing, We engineered an information recovery to work with Kalman filter. The Kalman filter first estimates the delayed target state at a slow sampling rate. The information recovery then reconstructs a fast-sampled target state and compensates for the delay [37].

Target state estimation considering full sensor dynamics and collaborative sensor

The above state estimation is still under the assumption that all input data are regularly sampled. To additionally handle the irregularly sampling time, the proposed memory-enabled auto-restart Kalman filter (M-ARKF) considers each measurement to have variable time delays. We developed this algorithm for the perception system of a robotic air-hockey player, where the robot is designed to play air-hockey games with a human player, under both traditional frame-based cameras and an new event-based camera. M-ARKF can recover from external disturbances quickly, by a designed triggering mechanism for filter re-initialization.

The remainder of the dissertation is organized as follows. Chapter 2 first introduces a method to re-parameterize an autoregressive model to a different structure. This model re-parameterization is the foundation for various versions of information recovery algorithms, which are detailed in Chapter 3. Chapter 3 also includes the multirate forward-model disturbance observer for beyond-Nyqusit disturbance rejection. Then we come back to the robotic target following problem in Chapter 4 and discuss how to integrate information recovering and Kalman filters, such that the robot can still follow a fast moving target under slow and delayed sensing. The perception system design for the robotic air-hockey player is included in Chapter 5. Finally, Chapter 6 concludes the dissertation.

Chapter 2

RE-PARAMETERIZATION OF AUTOREGRESSIVE MODEL

The key to compensate the controller performance loss due to sensor's slow-sampling speed is to reconstruct fast-sampled signals from the slow-sampled ones. This reconstruction is possible only if we know some prior information of the fast-sampled signals to be recovered. To see this, consider a random signal. If a reconstruction method exists without any prior information, then a random signal can be recovered as well, which contradicts the definition of random signal. In this dissertation, we design the reconstruction algorithms based on the assumption of knowing the model structure of fast-sampled signals.

There are two models used in our discussion. One is the state space model where the signal and its time-derivatives are put into a column vector which is called a state $\boldsymbol{x}$. The state space model is a linear transformation from a state $\boldsymbol{x}[k]$ to the next state $\boldsymbol{x}[k+1]$.

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] \tag{2.1}$$

In this chapter, however, we will focus on the other model, called autoregressive model.

2.1 Introduction to autoregressive model

For a discretized signal sequence $d[n]$, the autoregressive model states the following linear equation

$$d[k] = -a_1 d[k-1] - a_2 d[k-2] - \cdots - a_n d[k-n],\ a_n \neq 0 \tag{2.2}$$

where $a_1, a_2, \ldots, a_n$ are the parameters of this autoregressive model, n is the order of this autoregressive model. We use AR(n) to represent an autoregressive model of an order n. Eq. (2.2) can be written in a more compact form:

$$(1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}) d[k] = A(z^{-n}) d[k] = 0 \tag{2.3}$$

where z^{-n} is the delay operator such that $z^{-n}d[k] = d[k-n]$. The n-order polynomial $A(z^{-1})$, referred as the autoregressive model of $d[n]$, contains all parameters in the AR(n) model. $A(z^{-1})$ is also referred to as a polynomial model or internal signal model of $d[n]$.

The autoregressive model can be obtained from the discrete transfer functions of $d[n]$.

Lemma 2.1. *If a discretized signal $d[n]$ has a transfer function $D(z^{-1}) = B(z^{-1})/A(z^{-1})$. Then the autoregressive model of $d[n]$ is $A(z^{-n})$.*

Proof. By definition of a transfer function, $d[n]$ is the response of its transfer function $D(z^{-1})$ with a delta impulse signal $\delta[n]$ as input.

$$d[n] = \frac{B(z^{-1})}{A(z^{-1})}\delta[n] \tag{2.4}$$

That is,

$$A(z^{-1})d[n] = B(z^{-1})\delta[n] \tag{2.5}$$

Let n_b be the order of polynomial $B(z^{-1})$, then $B(z^{-1})\delta[n] = 0$ when $n > n_b$. Then $A(z^{-1})d[n] = 0$ at the steady state. Hence $A(z^{-1})$ is an autoregressive model of $d[n]$. □

2.2 Change the order of an autoregressive model

One signal sequence could have multiple autoregressive models. Given an autoregressive model $AR(n_1)$ of order n_1, we can re-parameterize this model to a higher order n_2.

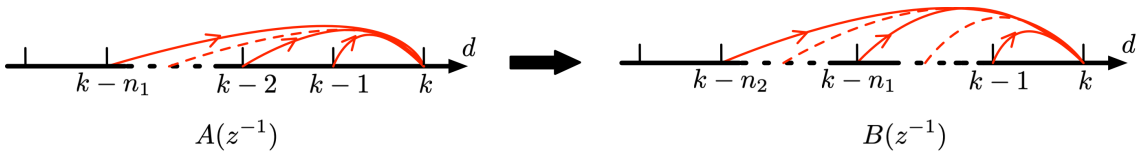


Figure 2.1: Re-parameterize an AR(n_1) model to AR(n_2) model.

Theorem 2.1. *If there exists an autoregressive model $A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_{n_1}z^{-n_1}$ for a sequence $d[n]$. Then we can obtain another autoregressive model $B(z^{-1}) =$*

$1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_2} z^{-n_2}$ *($n_2 > n_1$) for the same sequence* $d[n]$. *The parameters* $\{b_i\}_{i=1,2\ldots,n_2}$ *can be solved from the following system of linear equations*

$$\begin{bmatrix} \tilde{\boldsymbol{M}}_k & \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{n_2} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_{n_2-n_1} \\ b_1 \\ \vdots \\ b_{n_2} \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_{n_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.6}$$

where

$$\tilde{\boldsymbol{M}}_k \triangleq \begin{bmatrix} 1 & 0 & \ldots & 0 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_m & \ddots & \ddots & 1 \\ 0 & \ddots & \ddots & a_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & a_m \end{bmatrix}_{n_2 \times (n_2 - n_1),} \tag{2.7}$$

and $\boldsymbol{e}_j$ *is the elemental column vector whose entries are all zero except for the* j*-th entry, which equals* 1.

Proof. Consider the polynomial equation

$$A(z^{-1})F(z^{-1}) + B(z^{-1}) = 0, \tag{2.8}$$

where $A(z^{-1})$, $F(z^{-1})$ and $B(z^{-1})$ are three polynomials with order n_1, $(n_2 - n_1)$ and n_2, respectively. If $A(z^{-1})$ is the given autoregressive model (i.e., $A(z^{-1})d[n] = 0$), then

$$B(z^{-1})d[n] = -A(z^{-1})F(z^{-1})d[n] = 0. \tag{2.9}$$

That is, $B(z^{-1})$ is also an autoregressive model of $d[n]$ (with a higher order of n_2). Expanding the Eq. (2.8) and collecting the coefficients of z^{-i} $(1, 2, \cdots, n_2)$, one can obtain n_2 equations, which can be written in matrix format Eq. (2.6). □

For example, an AR(4) model $B(z^{-1}) = 1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}$ for a signal $d[n]$ can be obtained from its AR(2) model $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2}$ by solving the following system of linear equations

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 1 & 0 & 0 \\ a_2 & a_1 & 0 & 0 & 1 & 0 \\ 0 & a_2 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ 0 \\ 0 \end{bmatrix}. \tag{2.10}$$

2.3 Re-parameterize autoregressive model with constraints

The system of linear equations (2.6) is under-determined (there are n_2 equations, $2n_2 - n_1$ unknowns to solve). This enables a design freedom for the new autoregressive model. More specifically, we can assign some of the parameters in the new autoregressive model, then solve the remaining parameters such that the new model is still valid.

Theorem 2.2. *If there exists an autoregressive model* $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_1} z^{-n_1}$ *for a sequence* $d[n]$. *Then we can obtain a new autoregressive model* $B(z^{-1}) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_2} z^{-n_2}$ *for the same sequence* $d[n]$. *There can be* s *number of parameters in the new model come with assigned values* $\{b_{k_1}, b_{k_2}, \ldots, b_{k_s}\}$. *The remaining parameters* $\{b_{u_1}, b_{u_2}, \ldots, b_{u_t}\}$ $(s + t = n_2,\ t \geq n_1)$ *can be solved from the following system of*

linear equations

$$\begin{bmatrix} \tilde{\boldsymbol{M}}_k & \boldsymbol{e}_{u_1} & \boldsymbol{e}_{u_2} & \cdots & \boldsymbol{e}_{u_t} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_{s+t-n_1} \\ b_{u_1} \\ \vdots \\ b_{u_t} \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_{n_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \sum_{i=0}^{s} \boldsymbol{e}_{k_i} b_{k_i} \tag{2.11}$$

where

$$\tilde{\boldsymbol{M}}_k \triangleq \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_m & \ddots & \ddots & 1 \\ 0 & \ddots & \ddots & a_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_m \end{bmatrix}_{(s+t)\times(s+t-n_1),} \tag{2.12}$$

and $\boldsymbol{e}_j$ *is the elemental column vector whose entries are all zero except for the* j*-th entry, which equals* 1*.*

Proof. Similar to the proof of Theorem 2.1, we construct the polynomial equation

$$A(z^{-1})F(z^{-1}) + B(z^{-1}) = 0, \tag{2.13}$$

where $A(z^{-1})$ is the original model of $d[n]$, $F(z^{-1})$ is a polynomial with an order of $(s+t-n_1)$, and $B(z^{-1})$ is the new model with partially assigned parameters. Expanding the above equation and collecting the coefficients of z^{-i} $(1, 2, \cdots, s+t)$, one can obtain $s+t$ equations, which can be written in matrix format Eq. (2.12). □

For example, one can re-parameterize an AR(2) model $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2}$ as an AR(4) model with two assigned parameters $B(z^{-1}) = 1 + b_2 z^{-2} + b_4 z^{-4}$. Note that b_1 and

b_3 in the new model have been assigned with the value of zero. The remaining parameters (i.e., b_2 and b_4), according to Theorem 2.2, can be solved from the following system of linear equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & 1 & 0 \\ a_2 & a_1 & 0 & 0 \\ 0 & a_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ b_2 \\ b_4 \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ 0 \\ 0 \end{bmatrix}. \tag{2.14}$$

Theorem 2.2 has enabled a lot of freedom when designing the new autoregressive model. First, we can choose the order of the new model. Second, for each parameter in the new model, we could make it either fixed (i.e., with an assigned value) or adaptive (i.e., solved by system of linear equations). Assigning a fixed value of zero will cut off the connection between $d[n]$ and one of its delayed variant $d[n-k]$. We have used this design in our information recovery algorithm to skip the missing data point due to slow-sampling. Third, we have the freedom to design what value to be assigned to each fixed parameter. Special designed values can add noise filtering capability to the information recovery algorithm. We will discuss more in the next chapter.

Chapter 3

INFORMATION RECOVERY AND BEYOND-NYQUIST DISTURBANCE REJECTION

In this chapter, we discuss information recovery algorithms that can recover a fast-sampled signal from slow-sampled one. We will first introduce the basic information recovery algorithm and it's application to disturbance compensation control in section 3.1. This basic information recovery algorithm is based on a special structure design of the re-parameterized autoregressive model. If the disturbance is a narrow-band signal, Section 3.2 proposes a more complicated information recovery algorithm that has the additional noise filtering capability. Section 3.3 will discuss information recovery from two slow-sampled signals collected from two sensors with different sampling speeds.

3.1 Rejecting beyond-Nyquist disturbances with the information recovery algorithm

Our initial idea of information recovery comes from the controller design for compensating disturbances with high frequency components. A fundamental challenge arises in feedback control if the sampling of the output is not fast enough to capture the major frequency components of the disturbances—or more specifically, when significant disturbances occur beyond Nyquist frequency. Consider the system in Fig. 3.1, where the solid and dashed lines represent, respectively, continuous- and discrete-time signal flows. The main elements here include the continuous-time plant $P_c(s)$, the sampler $\mathscr{S}$ at a sampling time of T_s sec, the discrete-time controller $C(z)$, and the signal holder $\mathscr{H}$. Given the common hardware complication and the theoretical limitation [38] of generalized hold functions, we assume that the DAC implements a ZOH throughout the discussion.

The control system is subject to external disturbance d_c. An example of such disturbance

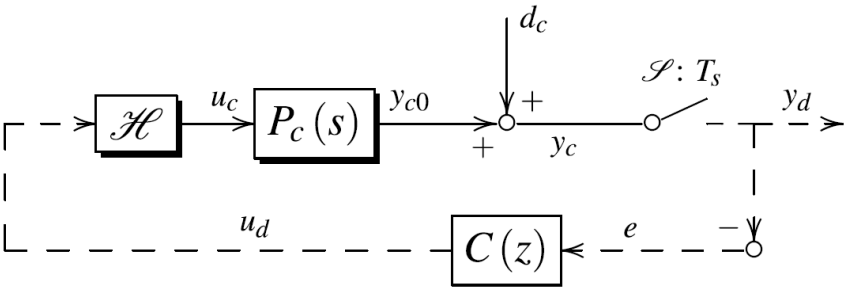


Figure 3.1: Block diagram of a sampled-data control system.

is the external forces applied to the read/write head of a hard disk drive due to mechanical vibration. Under the assumption that the sampling rate of feedback sensor is sufficiently high, state-of-the-art single-rate high-gain control [2] has been shown successful. However, when the feedback sensor's sampling rate is limited due to, for example, hardware limitations or time requirements for data processing, the performance of single-rate controls will be downgraded. Research [39] has shown that if the disturbance is near or beyond the Nyquist frequency (i.e., half of the sensor's sampling frequency), the single-rate control is actually amplifying the disturbance, instead of compensating it.

To overcome this fundamental challenge, we provide a mixed-rate feedback solution for the missing piece of exact disturbance rejection at frequencies beyond the Nyquist limitation. This is achieved by the introduction of a multirate forward-model disturbance observer (MR-FMDOB) that enables full rejection of structured disturbances at both the sampling and any uniformly spaced inter-sample instances. Built on top of a baseline controller at the regular sampling time of T_s, such an exact compensation scheme constructs an internal feedback loop implemented at a higher sampling rate, where the inter-sample signals are constructed with model-based prediction using the slow T_s-sampled data. Integrating features of all-stabilizing control parameterization [40, 41], the internal add-on loop guarantees the overall closed-loop stability and offers the convenience of being decoupled from the design of the baseline sub-Nyquist controller. Similar to [42], an internal model of the disturbance is implicitly

integrated in the inner loop. The major differences here are the decoupled design of the fast-rate MR-FMDOB from the slow-rate baseline feedback control, and the all-stabilizing based observer structure that preserves features of the baseline servo. These properties enable additional design features such as adaptive control in the baseline or the multirate compensator, and are also useful for applications where the existing baseline servo has special features that are required to be preserved.

This section provides a discrete-time regulation scheme for exact sampled-data rejection of disturbances beyond Nyquist frequency. By introducing a model-based multirate predictor and a forward-model disturbance observer, we show that the intersample disturbances can be fully attenuated despite the limitations in sampling and sensing.

3.1.1 Multirate forward-model disturbance observer

Fig. 3.2 shows the proposed servo scheme for disturbance rejection beyond Nyquist frequency. Two groups of discrete signals are present, with their different sampling rates indicated by the dashed (slower) and dotted (faster) signal flows.

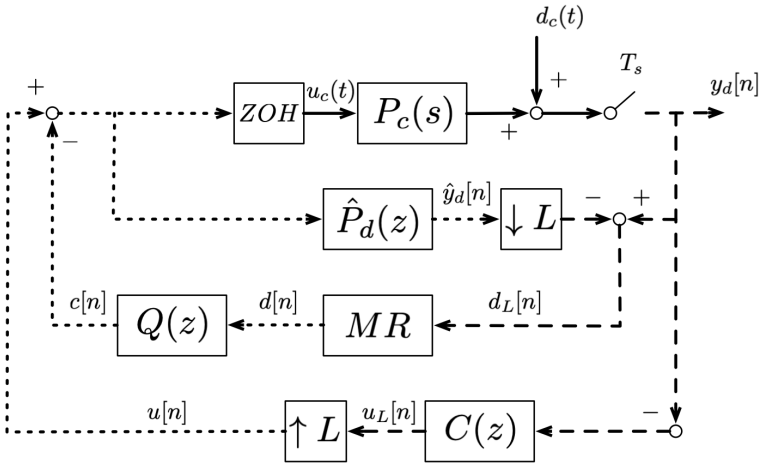


Figure 3.2: The proposed multirate disturbance rejection scheme.

In the block diagram, the upsampler (between $u_L[n]$ and $u[n]$) generates a fast signal $u[n]$

at a sampling time of T_s/L. The upsampled signal then passes through a T_s/L-based ZOH, with a transfer function $H_L(s) = (1 - \mathrm{e}^{-sT_s/L})/s$.

The beyond-Nyquist disturbance rejection consists of two fast-sampling transfer functions $Q(z)$ and $\hat{P}_d(z)$, a downsampling operator, and a model-based information recovery block (MR) inbetween the downsampler and $Q(z)$. In the subsequent derivations, we show that although $y_d[n]$ only contains information sampled at T_s, the inter-sample information in $d_c(t)$ can be fully reconstructed with MR in Fig. 3.2, if $d_c(t)$ satisfies a disturbance model; and in that case, $c[n]$—the output of $Q_(z)$—can fully remove the effect of the beyond-Nyquist sampled disturbance at a fast sampling period of T_s/L.

If the sampling time in Fig. 3.2 were T_s/L, the focused signal is the sampled output, and the downsampler and the MR block were removed, then the top part of the block diagram is equivalent to the structure in Fig. 3.3. Here, $P_d(z)$ is the ZOH equivalent of $P_c(s)$, with a fast sampling time T_s/L; $d[n](\triangleq d_c(nT_s/L))$ and $y[n]$ are the T_s/L–sampled disturbance and the plant output, respectively.

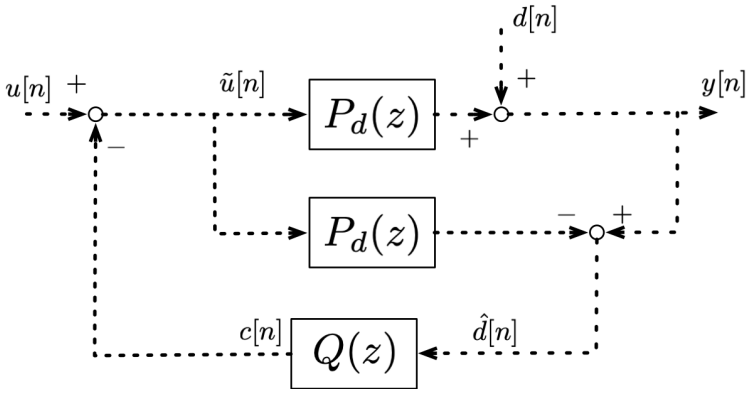


Figure 3.3: The proposed forward-model disturbance observer

The disturbance compensation structure is branched from internal model control (IMC) [43], a special case of Youla-Kucera/all-stabilizing parameterization [44]. Compared with a standard IMC, the major difference here is that $u[n]$—the command external to the local feedback loop—is injected before the plant instead of serving as a reference before $Q(z)$. The

resulting output satisfies, after some block-diagram algebra,

$$Y(z) = P_d(z)U(z) + (1 - P_d(z)Q(z))D(z), \tag{3.1}$$

where the relationship between $u[n]$ and the output remains intact as if the local feedback does not exist; and additional dynamics is introduced between $d[n]$ and $y[n]$. The first property allows the design of the local feedback to be separated from that of the baseline control command $u_L[n]$. In other words, the design of $Q_L(z)$ is decoupled from that of $C(z)$ in Fig. 3.2. The second property enables a disturbance estimation and cancellation scheme. In more details, with the forward plant model in the center of Fig. 3.3, block diagram analysis gives that the input to $Q(z)$ is

$$Y(z) - P_d(z)\tilde{U}(z) = \cancel{\big(P_d(z)\tilde{U}(z)} + D(z)\big) \cancel{- P_d(z)\tilde{U}(z)},$$

i.e. $d[n]$ in time domain. This disturbance estimation is then processed by the cancellation filter $Q(z)$ to be discussed in the remainder of this subsection. Based on the above intuitions, we will hereafter refer to the proposed scheme in Fig. 3.3 the forward-model disturbance observer (FMDOB).

The FMDOB has the same characteristic equation as IMC, since changing the allocation of input signals does not alter closed-loop stability. Hence FMDOB shares the same advantageous property of guaranteed stability as in IMC [43], if $Q(z)$ and $P_d(z)$ are stable.

With the stability property and the decoupled design principle, we now use the affine Q parameterization $1 - P_d(z)Q(z)$ to design $Q(z)$ for disturbance rejection. Observe the structure of $1 - P_d(z)Q(z)$. To achieve an exact rejection of the disturbance at a particular frequency ω_o in (3.1), it must be that

$$1 - P_d(\mathrm{e}^{j\omega_o})Q(\mathrm{e}^{j\omega_o}) = 0. \tag{3.2}$$

Let φ be the phase of $P_d(\mathrm{e}^{j\omega_o})$ at ω_o. (3.2) is equivalent to:

$$|Q(\mathrm{e}^{j\omega_o})| = \frac{1}{|P_d(\mathrm{e}^{j\omega_o})|}, \ \text{phase}(Q(\mathrm{e}^{j\omega_o})) = -\varphi. \tag{3.3}$$

In other words, $Q(\mathrm{e}^{j\omega_o}) = P_d(\mathrm{e}^{j\omega_o})^{-1}$ so that $Q(z)$ inverts the dynamics of $P_d(z)$ at ω_o.[1]

[1]We assume that the plant does not have a null gain at ω_o.

Recall that $Q(z)$ must be stable. Certainly, unless for special minimum-phase plants with a relative degree of zero, it is not feasible to always assign an exact full inversion $P_d(z)^{-1}$ to $Q(z)$ due to instability and non-properness. In addition, when $\omega \neq \omega_o$, the magnitude of $1 - P_d(\mathrm{e}^{j\omega})Q(\mathrm{e}^{j\omega})$ must be maintained small to avoid amplification of other components in $d[n]$. The following proposition achieves a point-wise stable inversion while providing the needed small gain to $|1 - P_d(\mathrm{e}^{j\omega})Q(\mathrm{e}^{j\omega})|$ for $\omega \neq \omega_o$.

Let $1/(2T_s) < \Omega_o < L/(2T_s)$ (in Hz) and $\omega_o = 2\pi\Omega_o T_s/L$ be the frequency of a major disturbance component beyond the baseline Nyquist frequency, $1/(2T_s)$ Hz, in Fig. 3.2. Let $\varphi = \text{phase}(P_d(\mathrm{e}^{j\omega_o}))$ be the phase response of the T_s/L–sampled discrete-time plant at ω_o; and assume that $P_d(\mathrm{e}^{j\omega_o}) \neq 0$ (otherwise no feedback design can achieve the disturbance rejection). We design the structure of filter $Q(z)$ to be

$$Q(z) = gQ_o(z)(b_0 + b_1 z^{-1}), \tag{3.4}$$

with $g \in [0, 1]$, and

$$b_0 = \frac{\cos\varphi - \sin\varphi \cot\omega_o}{|P_d(\mathrm{e}^{j\omega_o})|},\ b_1 = \frac{1}{|P_d(\mathrm{e}^{j\omega_o})|}\frac{\sin\varphi}{\sin\omega_o}; \tag{3.5}$$

$$Q_o(z) = \frac{1}{2}\frac{(1-k_2)(1+z^{-1})(1-z^{-1})}{1 + k_1(1+k_2)z^{-1} + k_2 z^{-2}},\ k_1 = -\cos\omega_o. \tag{3.6}$$

It can be proven [34] that such $Q(z)$ design satisfies the following three properties.

1. $1 - P_d(z)Q(z)$ in (3.1) equals $1 - g \overset{g=1}{=} 0$ at ω_o and can be controlled to have almost unity gain at other frequencies—in other words, the feedback system in Fig. 3.3 fully rejects all disturbances at ω_o at the sampling instances when $g = 1$, while maintaining the system dynamics at other frequencies.

2. Amplification at $\omega \neq \omega_o$, if any, is controlled by choosing

$$k_2 = \frac{1 - \tan(\frac{\pi B_w T_s}{L})}{1 + \tan(\frac{\pi B_w T_s}{L})} \tag{3.7}$$

and g, where B_w (in Hz) is the 3-dB disturbance-rejection bandwidth of $Q_o(z)$ centered around ω_o.

3. The overall magnitude of $1 - P_d(\mathrm{e}^{j\omega})Q(\mathrm{e}^{j\omega})$ satisfies

$$\int_0^{\pi} \ln|1 - P_d(\mathrm{e}^{j\omega})Q(\mathrm{e}^{j\omega})|\mathrm{d}\omega = \pi\left(\sum_{i=1}^{n_\gamma} \ln|\gamma_i| - \ln|\sigma + 1|\right), \tag{3.8}$$

where $\{\gamma_i\}_{i=1}^{n_\gamma}$ ($n_\gamma \geq 0$) is the set of unstable zeros of $1 - P_d(z)Q(z)$ ($\{\gamma_i\}_{i=1}^{n_\gamma} \triangleq \emptyset$ if $n_\gamma = 0$), and

$$\sigma = \lim_{z\to\infty} P_d(z)Q(z)/(1 - P_d(z)Q(z)).$$

3.1.2 Model-based information recovery

In this subsection, design of the MR block in Fig. 3.2 is provided, to establish the equivalence of the MR-FMDOB to the fast-rate FMDOB in Fig. 3.3.

Recall that the input to $Q(z)$ in Fig. 3.3 is an estimate of $d[n]$; while the input to MR is the slow T_s–sampled $d_L[n]$. The next result shows that if $d[n]$ satisfies an autoregressive model, $d[n]$ can be recovered from $d_L[n]$. Intuitively, with such a signal model, inter-sample values of the disturbance can be reconstructed by using the historical data. We first provide the general case of the information recovery, then present an example that specifically addresses prediction at a particular frequency.

Theorem 3.1. *If there exists an autoregressive model $A(z^{-1})$ for $d[n]$, we show that $d[n]$ can be fully recovered from the slowly sampled $d_L[n]$ by*

$$d[nL] = d_L[n] \tag{3.9}$$

and for $k = 1, 2, ..., L-1$,

$$d[nL + k] = W_k(z^{-}1)d_L[n] \tag{3.10}$$

$$= w_{k,0}d_L[n] + w_{k,1}d_L[n-1] + \cdots + w_{k,n_{w_k}}d_L[n - n_{w_k}] \tag{3.11}$$

where n_{w_k} is the order of $W_k(z^{-1})$. The minimum required order for $W_k(z^{-1})$ is $n^*_{w_k} = m-1$, in which case the coefficients of $W_k(z^{-1}) = w_{k,0} + w_{k,1}z^{-1} + \cdots + w_{k,m-1}z^{-m+1}$ come from

the unique solution of

$$M_k \begin{bmatrix} f_{k,1} \\ \vdots \\ f_{k,L(m-1)-m+k} \\ \hline w_{k,0} \\ \vdots \\ w_{k,m-1} \end{bmatrix} = - \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{3.12}$$

$$M_k \triangleq \left[\begin{array}{c|cccc} \tilde{M}_k & e_k & e_{k+L} & \dots & e_{k+(m-1)L} \end{array} \right] \tag{3.13}$$

$$\in \mathbb{R}^{[L(m-1)+k]\times[L(m-1)+k]}, \tag{3.14}$$

where e_j is the elemental column vector whose entries are all zero except for the j-th entry, which equals 1*; and $\tilde{M}_k$ is*

$$\tilde{M}_k \triangleq \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_m & \ddots & \ddots & 1 \\ 0 & \ddots & \ddots & a_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_m \end{bmatrix}_{[L(m-1)+k]\times[L(m-1)-m+k].} \tag{3.15}$$

Proof. See A.1. □

Connection to autoregressive model re-parameterization: Note that Theorem 3.1 can be viewed as a special form of Theorem 2.2, under the problem configuration of information recovery. Eq. 3.11 defines the structure of the new autoregressive model.

Transient response: It is recognized that $W_k(z^{-1})$ is a finite-impulse-response filter. The transient of the reconstruction process in (3.11) equals n_{w_k} discrete time steps, which is usually very fast compared to the transient of the feedback servo control.

Corollary 3.1. *$\forall\phi \in \mathbb{R}$ and $1/(2T_s) < \Omega_o < 1/T_s$, let $d_c(t) = \cos(2\pi\Omega_o t + \phi)$. Let the analog signal be sampled at T_s, i.e., Ω_o is in between the Nyquist frequency and the sampling frequency. The beyond-Nyquist information can be fully recovered by letting $\omega_o = 2\pi\Omega_o T_s/L$, $L = 2$, and reconstructing the T_s/L-sampled $d_L[n] = \cos(\omega_o n + \phi)$ with*

$$d_L[2n] = d[n] = d_c(nT_s), \tag{3.16}$$

$$d_L[2n+1] = \left(2\cos\omega_o - \frac{1}{2\cos\omega_o}\right) d[n] - \frac{1}{2\cos\omega_o} d[n-1]. \tag{3.17}$$

Proof. Ω_o is below the new Nyquist frequency corresponding to the sampling time of $T_s/2$. Based on Shannon's sampling theory, $L = 2$ is sufficient for $d_L[n]$ to recover the analog information. For $d_L[n] = \cos(\omega_o n + \phi)$, we have, from the table of Z transform,

$$d_L[n] = \frac{(1 - z^{-1}\cos\omega_o)\cos\phi - z^{-1}\sin\omega_o\sin\phi}{1 - 2z^{-1}\cos\omega_o + z^{-2}}\delta[n];$$

and hence $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} = 1 - 2\cos\omega_o z^{-1} + z^{-2}$, as $A(z^{-1})d_L[n] = (1 - z^{-1}\cos\omega_o)\cos\phi\delta[n] - \sin\omega_o\sin\phi\delta[n-1] = 0\ \forall n \geq 2$. Apply Theorem 3.1 with $m = 2$ in $A(z^{-1})$ and $L = 2$. Only one intersample point needs to be reconstructed, i.e., $k = 1$ in (3.11). The constrained Diophantine equation becomes

$$A(z^{-1})F(z^{-1}) + z^{-1}W(z^{-2}) = 1, \tag{3.18}$$

and (3.12) is

$$\left[\begin{array}{c|cc} 1 & 1 & 0 \\ a_1 & 0 & 0 \\ a_2 & 0 & 1 \end{array}\right] \left[\begin{array}{c} f_1 \\ \hline w_0 \\ w_1 \end{array}\right] + \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.19}$$

which gives $d_L[2n+1] = w_0 d[n] + w_1 d[n-1]$ in (3.17).

□

3.1.3 Numerical verification

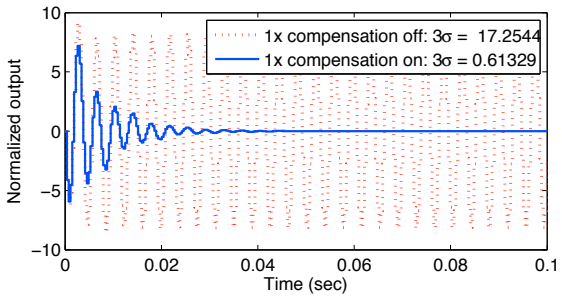
Consider a plant $P_c(s) = 3.74488 \times 10^9 \mathrm{e}^{-0.01s}/(s^2 + 565.5s + 319775.2)$, which is the model of a high-precision linear stage used in semiconductor manufacturing. Let the sampling time be

limited at $T_s = 1/2640$ second, and the baseline discrete-time controller be a PID controller $C(z) = k_p + k_i/(z-1) + k_d(z-1)/z$ with $k_p = 1/13320$, $k_i = 1/33300$, and $k_d = 1/2775$. Plotting the closed-loop discrete-time sensitivity function reveals that such a baseline design provides a common discrete-time loop shape with sufficient gain and phase margins.

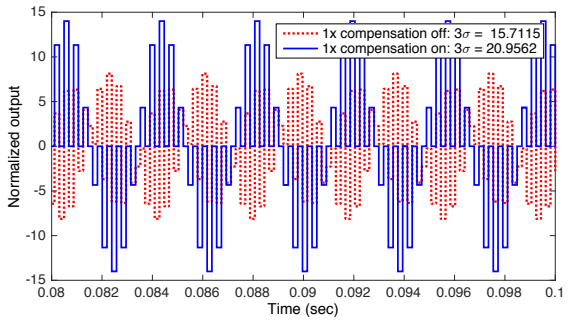
Continuous-time vibrations are applied to the plant, above Nyquist frequency. For convenience of illustration, we denote Ω_N(=1320 Hz) as the Nyquist frequency. To see the limitation (and danger) of sub-Nyquist design in this case, the narrow-band disturbance observer [45] is applied on top of the PID controller, to enable infinite-gain control at selective frequencies below Ω_N. Such a design provides perfect compensation of sinusoidal signals below Nyquist frequency, and is termed 1x compensation. Fig. 3.4 presents the corresponding plant output. When the disturbance occurs at 2376 Hz (i.e. $1.8\Omega_N$), although the sub-Nyquist servo enhancement enforces the T_s-sampled output to converge to zero in Fig. 3.4a, the actual output is significantly amplified. The 3σ (σ is the standard deviation) value of the $T_s/2$-sampled output increased from 15.71 to 20.96, yielding more than 130% of error amplification. To reveal more details in the error amplification, Fig. 3.4c shows the spectrum of the output sampled at $T_s/2$ sec. Two peaks are present in each spectral plot: the second revealing the energy of the actual disturbance; while the first—symmetric to the second peak with respect to Ω_N—comes from the aliased disturbance component below Nyquist frequency. After aliasing below Nyquist frequency, the two peaks with opposite phase values are canceled to yield the deceiving zero steady state value in Fig. 3.4a. Yet, the output energy in the top plot of Fig. 3.4c is actually increased, as the sub-Nyquist design did not target at the true disturbance spectrum beyond Ω_N.

In Table 3.1, a series of disturbances at frequencies between Ω_N and $2\Omega_N$ are applied to the plant. Based on the 3σ value of the output, the largest servo degradation under sub-Nyquist high-gain control occurred when the disturbance frequency is closest to the Nyquist frequency, where the tracking errors get amplified by 165.12% compared to the baseline value.

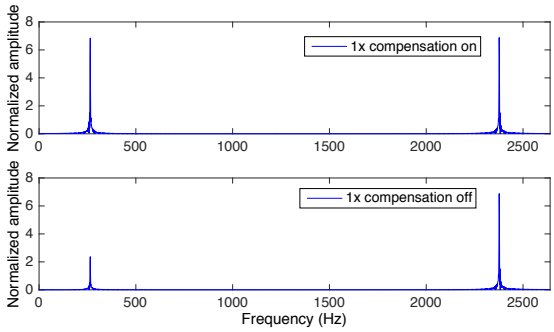
Figure 3.5 shows an example of the time-domain signal recovery by the MR algorithm.



(a) output sampled at T_s.



(b) output sampled at $T_s/2$.



(c) Fast Fourier Transform (FFT) of $y_c(t)$ sampled at $T_s/2$.

Figure 3.4: Plant output for the case with disturbance at $1.8\Omega_N$.

Here, the actual harmonic signal is plotted in $*$ at a fast sampling time of $T_s/2$; the measured signal is plotted in the solid black line marked by circles. Under the limited sensing, the MR successfully reconstructed the hidden fast data (marked in rectangular) at the intersample instances.

3.2 Information recovery for narrow-band signals

In section, we focused on a more specific problem of recovering fast-sampled narrow-band signals from slow sampled ones. The narrow-band signals are those that contains only narrow spikes in its frequency domain. Mathematically, the extreme narrow-band signals

Table 3.1: Servo degradation w.r.t. different disturbances: $y_c(t)$ sampled at $T_s/20$.

disturbance frequency	3σ value of y		amplification
	1x compensation		
	off	on	
$1.3\Omega_N$	14.76	24.37	165.12%
$1.7\Omega_N$	15.25	21.01	137.72%
$1.8\Omega_N$	15.71	20.96	133.38%
$1.9\Omega_N$	17.39	20.94	120.38%

are summations of sinusoidal signals. That is,

$$d_c(t) = \sum_{i=1}^{m} \lambda_i \sin(2\pi f_i t + \phi_i) \tag{3.20}$$

Signals in this formula are common in system where there are periodic movements or physical vibrations. For example, in the beam steering system for selective laser sintering process, a galvo scanner is placed before an optical focusing system (e.g., an F-theta Lens) so that the laser bean maintains uniform energy density ans much as possible across the entire material surface (see Figure 3.6). During the sintering process, vibration of the optical system could greatly downgrade the accuracy of the beam positioning accuracy and thus affect the quality of the sintered part [35]. To reject such disturbances, a camera is used to provide real-time position feedback of the beam position. A special designed forward model disturbance observer then separates the disturbance signal from the position measurements. However, due to the slow sampling speed of the camera sensor, the sampling rate of the observed disturbance is inadequate to ensure a good disturbance rejection outcome. With the algorithms of recovering fast-sampled narrow-band signals, we have opened the control system's abilities to reject high-frequency disturbances under slow vision feedback.

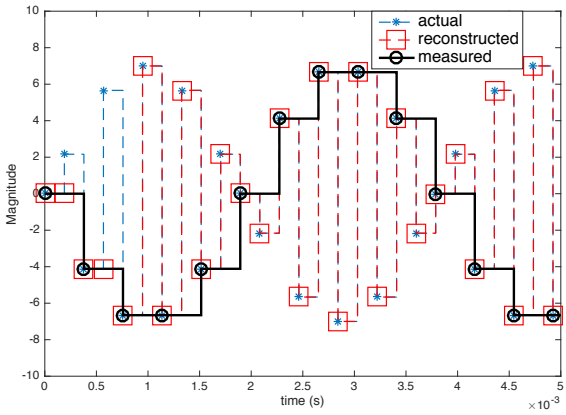


Figure 3.5: Time trace of MMP based data recovery.

3.2.1 Forward-model disturbance observer for narrow-band signals

Section 3.1 provided the block diagram of forward-model disturbance observer (FMDOB) (see Fig. 3.3) and the design of $Q(z)$ filter for disturbances with a single frequency. In this section, we extend the filter design to include disturbances with multiple frequencies.

Theorem 3.2. *Let T_s be the sampling time in Fig. 3.3 and $\{\omega_i = 2\pi f_i T_s\}_{i=1,2,\ldots,m}$ be a set of frequencies in rad/sec at which disturbance rejection is desired. Let $P_d(e^{j\omega_i})$ be the frequency response of $P_d(z)$ at ω_i, and assume that $|P_d(e^{j\omega_i})| \neq 0$, $i = 1, 2, \ldots, m$.*[2] *Let $p = 2m - 1$, and*

$$Q(z) = Q_0(z)(q_0 + q_1 z^{-1} + \cdots + q_p z^{-p}), \tag{3.21}$$

[2]Otherwise the plant will not respond to input at the target frequencies. That is, input disturbance at ω_i will not impact the plant output (inherently rejected), and output disturbance at ω_i will be immune to feedback controls because the plant does not pass through any signal components at the target frequency.

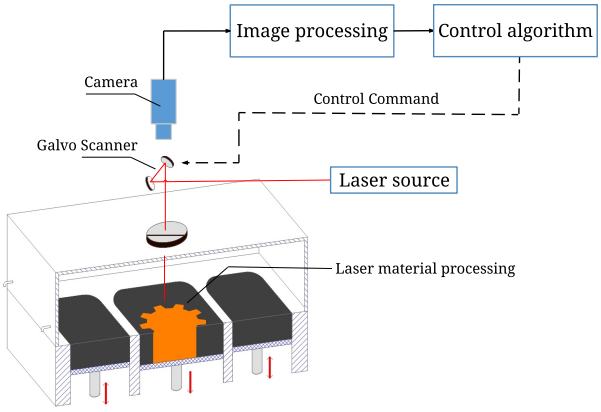


Figure 3.6: Galvo scanner system in selective laser sintering process.

with

$$\begin{bmatrix} q_0 \\ \vdots \\ q_p \end{bmatrix} = \begin{bmatrix} 1 & cos\omega_1 & \cdots & cosp\omega_p \\ 0 & sin\omega_1 & \cdots & sinp\omega_p \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & cos\omega_p & \cdots & cosp\omega_p \\ 0 & sin\omega_p & \cdots & sinp\omega_p \end{bmatrix}^{-1} \begin{bmatrix} \frac{\Re P_d(e^{j\omega_1})}{|P_d(e^{j\omega_1})|^2} \\ \frac{\Im P_d(e^{j\omega_1})}{|P_d(e^{j\omega_1})|^2} \\ \vdots \\ \vdots \\ \frac{\Re P_d(e^{j\omega_p})}{|P_d(e^{j\omega_p})|^2} \\ \frac{\Im P_d(e^{j\omega_p})}{|P_d(e^{j\omega_p})|^2} \end{bmatrix}, \tag{3.22}$$

$$Q_0(z) = 1 - \prod_{i=1}^{m} \left(\frac{1}{2} \frac{1 + 2k_{1,i}(1 + k_{2,i})z^{-1} + (1 + k_{2,i})z^{-2}}{1 + k_{1,i}(1 + k_{2,i})z^{-1} + k_{2,i}z^{-2}} \right), \tag{3.23}$$

where $\Re P_d$ *and* $\Im P_d$ *are the real part and imaginary part of* $P_d(e^{j\omega_i})$*, and*

$$k_{1,i} = -cos(\omega_i), \tag{3.24}$$

$$k_{2,i} = \frac{1 - tan(\pi\Omega_i T_s)}{1 + tan(\pi\Omega_i T_s)}. \tag{3.25}$$

Then Eq. (3.2) holds for each ω_i*, and the amplification at* $\omega \neq \omega_i$ *is controllable by choosing* Ω_i*, the 3-dB disturbance-rejection bandwidth of* $Q_0(z)$ *centered around* ω_i*.*

Proof. see A.2. □

3.2.2 Narrow-band signal recovery with the model-based predictor

Before we start discussing the model-based predictor, it is necessary to clarify that there exist some special cases in theory where the fast-sampled signal is actually not recoverable. Those signals all contains frequencies in the singular frequency set.

Singular frequency set. For narrow-band signals that has m frequency components, a frequency distribution $f = (f_1, f_2, \ldots, f_m)$ can be considered as a point in an m-dimension space $\mathbb{R}^m$. Let the slow sampling rate be f_{ss}, the singular set consists of a group of frequency configurations defined as

$$\sigma = \left\{ \; f \in \mathbb{R}^m \;\middle|\; \exists f_i, f_j, \text{ s.t. } f_i = n f_{ss}/2, \text{ or } f_i \pm f_j = n f_{ss} \; \right\}, \tag{3.26}$$

where $n \in \mathbb{Z}$, $i \neq j$ and $i, j \in \{1, 2, \ldots, m\}$.

For example, let the slow sampling rate be $f_{ss} = 2\text{kHz}$. Then a frequency distribution $f = (0.6\text{kHz}, 2\text{kHz})$ belongs to the singular frequency set because it has a frequency component (i.e $f_2 = 2\text{kHz}$) at an integer multiple of $f_{ss}/2$. A frequency distribution $f = (0.3\text{kHz}, 1.7\text{kHz})$ also belongs to the singular frequency set because $f_1 + f_2 = f_{ss}$. In general, two scenarios contribute to a singular frequency case. The first case arises when the disturbance has one or more frequency components at the Nyquist frequency of the slow sensor (i.e., $f_{ss}/2$), or its integer multiplications. In this case, there is a DC component in the frequency domain, and the intersample signal is not recoverable without its amplitude information. The second case arises when f contains pairs of frequency components in which one frequency is the alias of the other when sampled at the slow sensor speed. Thus their time-domain signals are fused together, and amplitude information of individual components is also needed to decouple them.

If the frequencies of the narrow-band signals are out of the singular frequency set, a fast signal measurements can be reconstructed by the slowly sampled $d_L[n]$ using model-based filtering, as is discussed next. Assume a multi-band signal $d_c(t)$ defined by Eq. (3.20) has m frequency components $f = (\; f_1 \quad f_2 \quad \ldots \quad f_m) \notin \sigma$. $d_L[n] = d_c(nLT_s)$ is the slow-sampled signal with sampling time LT_s, $L \in \mathbb{Z}^+$. This subsection provides the proposed approach to

recover a fast-sampled signal $d\,[n] = d_c(nT_s)$ with sampling time T_s.

Because the fast sampling time divides the slow sampling time, every L-th sample of $d[n]$ can be obtained from $d_L[n]$ directly, i.e.

$$d[nL] = d_L[n] \tag{3.27}$$

We show that the k-th intersample signal ($k = 1, 2, \ldots, L-1$) between $d[nL]$ and $d[(n+1)L]$ (denoted as $y_k[n] \triangleq d[nL+k]$) can be recovered by

$$y_k[n] = \boldsymbol{w}_k^T \cdot \boldsymbol{\varphi}_d[n] - \boldsymbol{b}^T \cdot \boldsymbol{\varphi}_y[n]. \tag{3.28}$$

where $\boldsymbol{\varphi}_d[n]$ and $\boldsymbol{\varphi}_y[n]$ are data vectors defined as

$$\boldsymbol{\varphi}_d[n] \triangleq [d_L[n], d_L[n-1], \cdots, d_L[n-(2m-1)]]^T, \tag{3.29}$$

$$\boldsymbol{\varphi}_y[n] \triangleq [y_k[n-1], y_k[n-2], \cdots, y_k[n-2m]]^T. \tag{3.30}$$

$\boldsymbol{w}_k$ and $\boldsymbol{b}$ are predictor parameter vectors

$$\boldsymbol{w}_k \triangleq \left[w_{k,0}, w_{k,1}, \cdots, w_{k,(2m-1)}\right]^T, \tag{3.31}$$

$$\boldsymbol{b} \triangleq [b_1, b_2, \cdots, b_{2m}]^T. \tag{3.32}$$

The parameter vector $\boldsymbol{b}$ is composed of the coefficients of polynomial $B(z^{-1}) = 1 + b_1 z^{-1} + \cdots + b_{2m} z^{-2m}$ which is computed from expanding the product

$$B(z^{-1}) = \prod_{i=1}^{m} (1 - 2\alpha cos(2\pi f_i L T_s) z^{-1} + \alpha^2 z^{-2}), \tag{3.33}$$

where $\alpha \in (0, 1)$ is a design parameter and z^{-1} is the one step delay operator.

The prediction formula (Eq. (3.28)) computes the k-th intersample y_k by a linear combination of $2m$ consecutive samples from $d_L[n]$, as well as by a linear combination of $2m$ previous predictions about y_k. The parameters $\boldsymbol{b}$ can be calculated by Eq. (3.33). The algorithm of obtaining vector $\boldsymbol{w}_k$ is provided in the following theorem.

Theorem 3.3. *Given the above definitions, $d[n]$ can be fully recovered from $d_L[n]$ by Eqs. (3.27) and (3.28), if f is not in the singular frequency set and $\boldsymbol{w}_k$ is from the solution of*

$$\mathbf{M}_k \begin{bmatrix} h_{k,1} \\ \vdots \\ h_{k,2m(L-1)} \\ w_{k,0} \\ \vdots \\ w_{k,2m-1} \end{bmatrix} = - \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2m} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \bar{\mathbf{b}}, \tag{3.34}$$

$$\mathbf{M}_k \stackrel{\triangle}{=} [\ \tilde{\mathbf{M}}_k \mid \boldsymbol{e}_k \quad \boldsymbol{e}_{k+L} \quad \cdots \quad \boldsymbol{e}_{k+(2m-1)L}]\ , \tag{3.35}$$

where $\mathbf{M}_k$ is a square matrix with a dimension of $2mL \times 2mL$; $\boldsymbol{e}_j$ is the elemental column vector whose entries are all zero except for the j-th entry, which equals 1; and

$$\tilde{\mathbf{M}}_k \stackrel{\triangle}{=} \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{2m} & \ddots & \ddots & 1 \\ 0 & \ddots & \ddots & a_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{2m} \end{bmatrix}_{2mL \times 2m(L-1).} \tag{3.36}$$

Parameters $[a_1, a_{2,} \cdots, a_{2m}]$ in Eqs. (3.34) and (3.36) come from the autoregressive model $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{2m} z^{-2m}$, which is computed by expanding

$$A(z^{-1}) = \prod_{i=1}^{m} (1 - 2cos(2\pi f_i T_s) z^{-1} + z^{-2}). \tag{3.37}$$

The column vector $\bar{\boldsymbol{b}}$ in the rightmost of Eq. (3.34) contains all zeros, except for the $L, 2L, \cdots, 2mL$-th entries, which equal $b_1, b_2, \cdots, b_{2m}$.

Proof. See A.3. □

Connection to autoregressive model re-parameterization: Note that Theorem 3.3 can also be viewed as a special form of Theorem 2.2. Equation (3.28) defines the structure of the new autoregressive model. The fixed/assigned parameters are collected in vector $\boldsymbol{b}^T$. The remaining parameters are in vector $\boldsymbol{w}_k^T$, which will be solved by the system of linear equation (3.34).

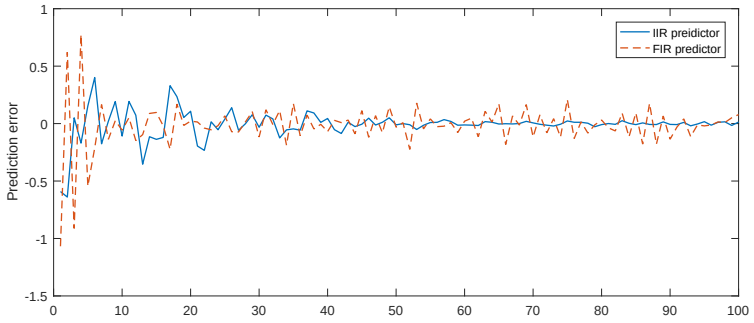
3.2.3 Choosing of the design parameter

Note that the intersample signals $y_k[n]$ are computed from not only the weighted sum of $d[n]$, but also the historical prediction values. In Eq. (3.33), α determines the weighting of input signals (sensor measurements) and historical prediction signals. As α gets closer to 1, the predictor will be more dependent on the historical predictions rather than input signals, thus less sensitive to the input noise (see Fig. 3.7a). As a trade-off, a predictor with larger α has a slower converging speed (see Fig. 3.7b). In Fig. 3.7, the slow sensor sampling speed is limited to $T_s = 0.8$ms, and $L = 3$. The disturbance has three bands at $f_1 = 187.5$Hz, $f_2 = 812.5$Hz and $f_3 = 1125$Hz, two of which (red dashed lines) are beyond the Nyquist frequency ($f_N = 625$Hz). The data vectors $\boldsymbol{\varphi}_d$ and $\boldsymbol{\varphi}_y$ are initialed as zero vectors.

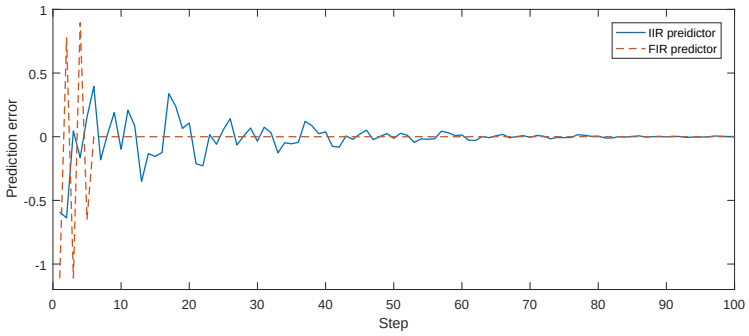
In this subsection, we discuss the effect of varying the design parameter α in Eq. (3.33). To get more insights in choosing α, one can derive the transfer function from $d_L[n]$ to $y_k[n]$ based on Eq. (3.28):

$$W_k^*(z) = \frac{w_{k,0} + w_{k,1}z^{-1} + \cdots + w_{k,(2m-1)}z^{-(2m-1)}}{1 + b_1 z^{-1} + \cdots + b_{2m}z^{-2m}}, \tag{3.38}$$

where the denominator is computed by Eq. (3.33), and the numerator can be solved by Eq. (3.34). Because filter $W_k^*(z)$ have an infinite-input response (IIR) structure, we call this design an IIR predictor. Without the denominator, $W_k^*(z)$ will be an finite-input response (FIR) predictor. Fig. 3.8 shows the bode plot of $W_k^*(z)$ with respect to different α's. Here, the configuration is the same as Fig 3.7. As is shown in the figure, a smaller α leads to a larger



(a) the input has a random noise with maximum amplitude of 0.05. The IIR predictor shows better robustness to noise.



(b) the input is noise-free, both the IIR and FIR predictors can make accurate predictions in steady state.

Figure 3.7: Prediction error of IIR ($\alpha = 0.95$) and FIR ($\alpha = 0$) predictor.

magnitude response. In the extreme case, if $\alpha = 0$, then the denominator of transfer function $W_k^*(z)$ becomes 1. Consequently, the predictor reduces to an FIR structure with a magnitude response above 0dB, and the predictions are purely dependent on input signals. Therefore, an IIR predictor with a larger α is more robust to input noise; an FIR predictor is most sensitive to input noise. In practice, in order to reduce the influence of measurement noise and increase the prediction accuracy, α is recommended to be chosen closer to 1, e.g, starting with $\alpha = 0.9$. The value can be further increased when dealing with noisy applications.

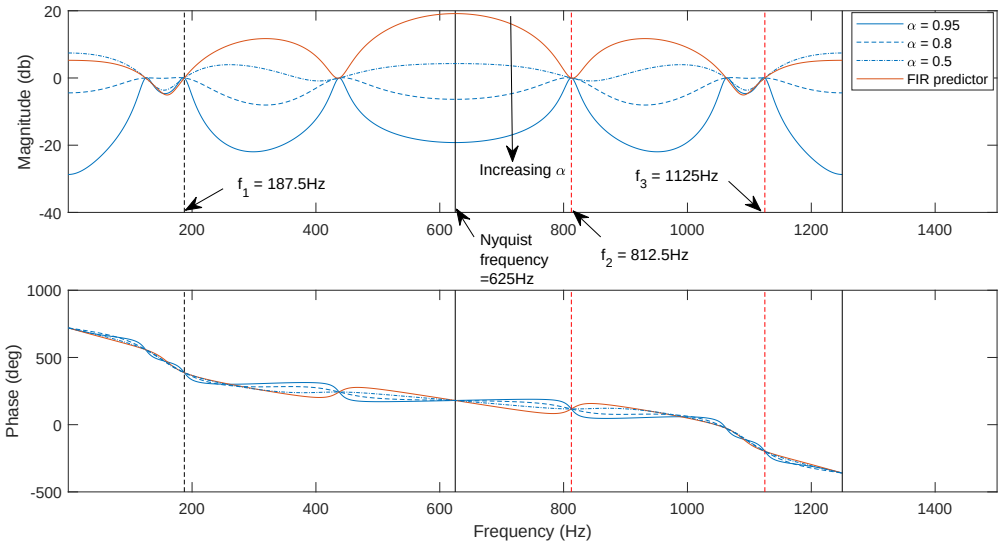


Figure 3.8: Bode plot of the IIR and FIR predictor.

3.2.4 Decoupled Solution Form and Computation Reduction

In **Theorem** 3.3, the dimension of $\boldsymbol{M}_k$ is scaled by the number of frequency components m and downsampling ratio L. When m and L are large, solving high-dimension linear system equations could be computationally expensive. However, it turns out that instead of solving Eq. (3.34), the predictor parameters can be solved from an equivalent form with reduced computation cost.

Let the augmented matrix[3] of system Eq. (3.34) be $\bar{\boldsymbol{M}} = [\ \boldsymbol{M}_k \mid \boldsymbol{b}\ ]$, where $\boldsymbol{b}$ is the summation of vectors on the right side of Eq. (3.34). Observing the sparse structure of the elemental vectors $\{\boldsymbol{e}_k, \cdots, \boldsymbol{e}_{k+(2m-1)L}\}$ in $\boldsymbol{M}_k$ in Eq. (3.35), we can swap rows of $\bar{\boldsymbol{M}}$ to yield, without changing the solutions to the associated linear system,

$$\bar{\boldsymbol{M}} = \left[\begin{array}{cc|c} \boldsymbol{A}_{2m\times 2m(L-1)} & \boldsymbol{I}_{2m\times 2m} & \boldsymbol{b}_1 \\ \boldsymbol{B}_{2m(L-1)\times 2m(L-1)} & \boldsymbol{0}_{2m(L-1)\times 2m} & \boldsymbol{b}_2 \end{array}\right], \tag{3.39}$$

where $\boldsymbol{I}$ is an identity matrix, $\boldsymbol{0}$ is a zero matrix, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are column vectors with length

[3]Given a linear system equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, the augmented matrix is defined as $\bar{\boldsymbol{A}} = [\boldsymbol{A} \mid \boldsymbol{b}]$.

$2m$ and length $2m(L-1)$. Let $\boldsymbol{h}_k = [\ h_{k,1} \quad \dots \quad h_{k,2m(L-1)}\ ]^T$, $\boldsymbol{w}_k = [\ w_{k,0} \quad \dots \quad w_{k,2m-1}\ ]^T$, then an equivalent form of Eq. (3.34) is

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{I} \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_k \\ \boldsymbol{w}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}, \tag{3.40}$$

or,

$$\boldsymbol{B}\boldsymbol{h}_k = \boldsymbol{b}_2, \tag{3.41}$$

$$\boldsymbol{w}_k = -\boldsymbol{A}\boldsymbol{h}_k + \boldsymbol{b}_1. \tag{3.42}$$

This suggests that one can solve system Eq. (3.41) first, then the predictor parameters $\boldsymbol{w}_k$ can be computed by a direct matrix-vector multiplication and a vector addition in Eq. (3.42).

For a better understanding of procedures of designing the IIR predictor, we give a step-by-step example. Consider the case where the disturbance contains one frequency component (i.e. $m = 1$) at $f = 1.2$kHz, and the fast sampling rate is $f_{sf} = 3$kHz. Without loss of generality, let the feedback sensor's sampling rate be $f_{smax} = 1.2$kHz, which is not fast enough because the disturbance occurs beyond the Nyquist frequency $f_N = f_{smax}/2 = 0.6$kHz. The proposed procedure for designing the predictor is as follows:

Choose L. This relates to f_{ss}, the sampling rate of the slow sensor, by the relationship $L = f_{sf}/f_{ss}$. Also, f_{ss} must be smaller than f_{smax}. The minimum L is then given by $L_{min} = \lceil f_{sf}/f_{smax} \rceil = 3$. Here, we choose $L = L_{min} = 3$ as an example.

Define predictor structure. The predictor is defined as (cf. Section 3.2.2)

$$d[3n] = d_L[n], \tag{3.43}$$

$$d[3n+1] = w_{1,0}d_L[n] + w_{1,1}d_L[n-1] - b_1 y_1[n-1] - b_2 y_1[n-2], \tag{3.44}$$

$$d[3n+2] = w_{2,0}d_L[n] + w_{2,1}d_L[n-1] - b_1 y_2[n-1] - b_2 y_2[n-2]. \tag{3.45}$$

where $y_1[n] = d[3n+1]$ and $y_2[n] = d[3n+2]$.

Compute parameters. Use Eq. (3.37) and Eq. (3.33) to obtain a_i's and b_i's: $a_1 = -2\cos(2\pi f T_s)$, $a_2 = 1$, $b_1 = -2\alpha\cos(6\pi f T_s)$, $b_2 = \alpha^2$.

Obtain $[w_{1,0}, w_{1,1}]$. From Eqs. (3.34-3.36), construct

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ a_1 & 1 & 0 & 0 & 0 & 0 \\ a_2 & a_1 & 1 & 0 & 0 & 0 \\ 0 & a_2 & a_1 & 1 & 0 & 1 \\ 0 & 0 & a_2 & a_1 & 0 & 0 \\ 0 & 0 & 0 & a_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{1,4} \\ w_{1,0} \\ w_{1,1} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ 0 \\ 0 \\ b_2 \end{bmatrix}. \tag{3.46}$$

Parameters $w_{1,0}$ and $w_{1,1}$ come from the unique solution of Eq. (3.46). Reorganizing the rows of corresponding augmented matrix yields

$$\tilde{\mathbf{M}} = \left[\begin{array}{cccc|cc|c} 1 & 0 & 0 & 0 & 1 & 0 & a_1 \\ 0 & a_2 & a_1 & 1 & 0 & 1 & 0 \\ \hline a_1 & 1 & 0 & 0 & 0 & 0 & a_2 \\ a_2 & a_1 & 1 & 0 & 0 & 0 & b_1 \\ 0 & 0 & a_2 & a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_2 & 0 & 0 & b_2 \end{array}\right]. \tag{3.47}$$

Then $w_{1,0}$, $w_{1,1}$ are given by

$$\begin{bmatrix} w_{1,0} \\ w_{1,1} \end{bmatrix} = -\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a_2 & a_1 & 1 \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{1,4} \end{bmatrix} + \begin{bmatrix} a_1 \\ 0 \end{bmatrix}, \tag{3.48}$$

where

$$\begin{bmatrix} h_{1,1} \\ h_{1,2} \\ h_{1,3} \\ h_{1,4} \end{bmatrix} = \begin{bmatrix} a_1 & 1 & 0 & 0 \\ a_2 & a_1 & 1 & 0 \\ 0 & 0 & a_2 & a_1 \\ 0 & 0 & 0 & a_2 \end{bmatrix}^{-1} \begin{bmatrix} a_2 \\ b_1 \\ 0 \\ b_2 \end{bmatrix}. \tag{3.49}$$

Obtain $[w_{2,0}, w_{2,1}]$. Following the same procedure as the last step, we have

$$\begin{bmatrix} w_{2,0} \\ w_{2,1} \end{bmatrix} = - \begin{bmatrix} a_1 & 1 & 0 & 0 \\ 0 & 0 & a_2 & a_1 \end{bmatrix} \begin{bmatrix} h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{2,4} \end{bmatrix} + \begin{bmatrix} a_2 \\ 0 \end{bmatrix}, \tag{3.50}$$

where

$$\begin{bmatrix} h_{2,1} \\ h_{2,2} \\ h_{2,3} \\ h_{2,4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_2 & a_1 & 1 & 0 \\ 0 & a_2 & a_1 & 1 \\ 0 & 0 & 0 & a_2 \end{bmatrix}^{-1} \begin{bmatrix} a_1 \\ b_1 \\ 0 \\ b_2 \end{bmatrix}. \tag{3.51}$$

3.2.5 Numerical verification

Consider the slow sampling time to be 0.3 milliseconds. Then the Nyquist frequency $f_N =$ 1667 Hz. In simulation, the disturbance d has three frequency components at $0.8f_N$, $1.6f_N$ and $2.3f_N$, respectively. The fast and slow sampling rates are $f_{sf} =$ 10kHz and $f_{ss} =$ 10/3kHz, respectively. Following the example procedure in section 3.2.4, we obtain parameter $\boldsymbol{b}$ in (3.33):

$$\boldsymbol{b} = \begin{bmatrix} -0.1668 & 0.7440 & 0.7068 & 0.6715 & -0.1359 & 0.7351 \end{bmatrix}.$$

and parameters $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$\boldsymbol{w}_1 = \begin{bmatrix} -0.0365 & -0.0877 & 0.1119 & -0.1501 & -0.0110 & 0.1043 \end{bmatrix},$$
$$\boldsymbol{w}_2 = \begin{bmatrix} -0.0688 & -0.0045 & 0.1522 & -0.1023 & 0.0903 & -0.0051 \end{bmatrix}.$$

Fig. 3.9 shows the time-domain disturbance reconstruction results by the IIR predictor. The dotted line represents the real-time disturbance signal. The slowly measured disturbance samples are marked with cross symbols, and the reconstructed disturbance samples are marked with circle symbols. In the simulation, we added a white noise with a noise level of 4% to the input of IIR predictor. The predictor successfully recovered the intersample data from the noisy and slow-sampled signal.

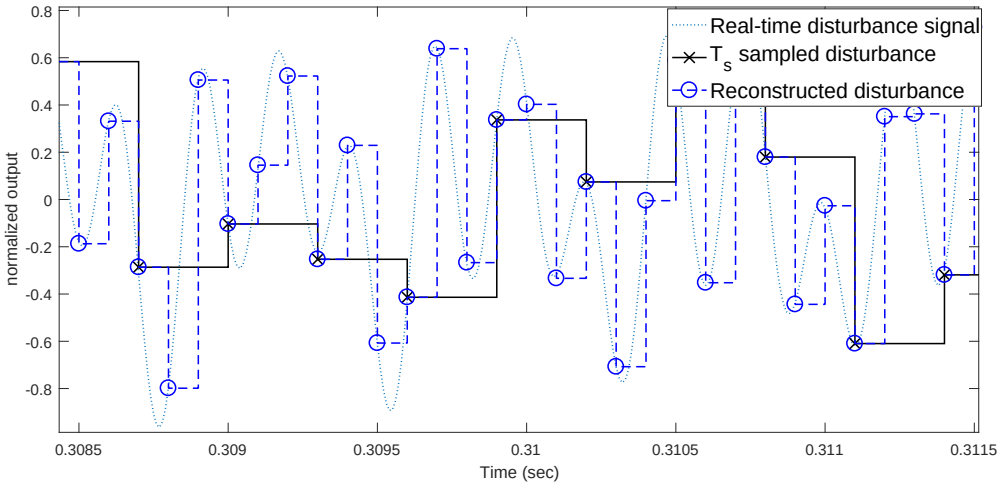


Figure 3.9: Disturbance reconstruction results by IIR predictor with $\alpha = 0.95$.

3.3 Information recovery from collaborative sensors

This section considers the real-time recovery of a fast time series by using sparsely sampled measurements from sensors whose sampling speeds are prohibitively slow originally. Specifically, when the fast signal is an autoregressive process, we propose an online information recovery algorithm that reconstructs the dense underlying temporal dynamics fully by systematically modulating two slow sensors, and by exploiting a model-based fusion of the sparsely collected data. More precisely speaking, we consider the case when $d[n]$ is a known autoregressive process and propose to obtain the sparsely sampled measurements from two sensors S_1 and S_2 with slow sampling periods MT and NT. The integers M and N are greater than one and $M \neq N$. This signal-reconstruction method is made possible by elaborately designing and re-parameterizing the autoregressive model of d[n]

3.3.1 Collaborative sensing modeling

Let $d[n]$ be a discrete time sequence with sampling time T, $d_M[n]$ and $d_N[n]$ be the discrete measurements from two sensors S_1 and S_2 with sampling times MT and NT, respectively.

The following direct connections hold:

$$d_X[n] \leftrightarrow d[Xn],\ X = M \text{ or } N. \tag{3.52}$$

In order to better describe the collaborative sampling process, we divide $d[n]$ into a list of subsequences $\{b_i\}_{i=1,2,3,\ldots}$, where b_i is referred to as the i-th batch in $d[n]$. Each batch contains L consecutive data points in $d[n]$, that is,

$$b_i[k] \leftrightarrow d[iL + k], \quad k = 1, 2, \ldots, L, \tag{3.53}$$

where $b_i[k]$ denotes the k-th data point in the i-th batch.

As a first result, when the batch size L is properly set, it can be shown that if the k-th data point in a batch is equal and aligned to a data point in $d_M[n]$ (or $d_N[n]$), then the k-th data point in the next batch will be equal and aligned to another data point in $d_M[n]$ (or $d_N[n]$):

Lemma 3.1. *Let the batch size $L = LCM(M, N)$, if $b_i[k] \leftrightarrow d_X[n]$, then $b_{i+1}[k] \leftrightarrow d_X[n+k_1]$, where $k_1 = L/X$ and $X = M$ or N.*

Proof. If $b_i[k] \leftrightarrow d_M[n]$, then combining (3.52) and (3.53), one can get $d[iL + k] \leftrightarrow b_i[k] \leftrightarrow d_M[n] \leftrightarrow d[Mn]$, hence their time stamps are equal: $t\{d[iL + k]\}/T = iL + k = Mn = t\{d[MN]\}/T$. Then it can be shown that the time stamps of $b_{i+1}[k]$ and $d_M[n + L/M]$ are equal: $t\{b_{i+1}[k]\}/T = (i + 1)L + k = M(n + L/M) = t\{d_M[n + L/M]\}/T$, where L/M is an integer. In addition, $b_{i+1}[k] = d[iL + k + L] = d[M(n + L/M)] = d_M[n + L/M]$. Thus we have $b_{i+1}[k] \leftrightarrow d_M[n + L/M]$. Analogously, $b_{i+1}[k] \leftrightarrow d_N[n + L/N]$ if $b_i[k] \leftrightarrow d_N[n]$. □

Lemma 3.1 suggests that the connections between $d_M[n]$, $d_N[n]$ and $d[n]$ are repeated over batches (see Fig. 3.10), if the batch size L is chosen as LCM(M, N). This property of repeated connections makes it possible to design a procedure to recover one batch of signal points, then use the procedure repetitively to recover other batches. With this in mind, we design our recovering algorithm under the following batch configurations.

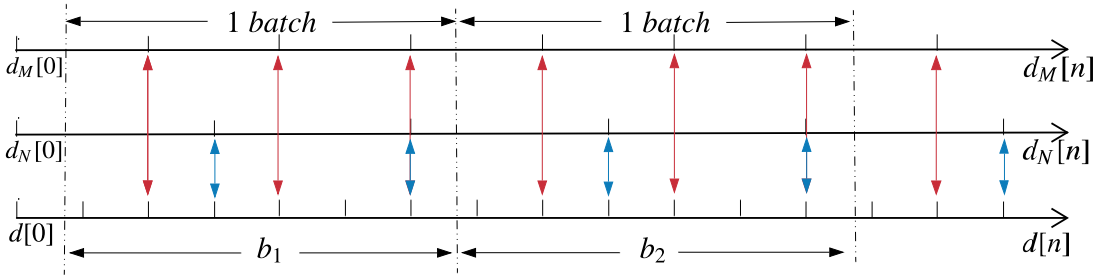


Figure 3.10: Connections between $d_M[n]$, $d_N[n]$ and $d[n]$ when $M = 2$, $N = 3$ and $L = 6$.

Definition 3.1. *A batch $b_i[k]$ (An example is shown in Fig. 3.10) is defined based on the following rules:*

1. *The first data points in $d[n]$, $d_M[n]$ and $d_N[n]$ are aligned in time, i.e., $d[0] \leftrightarrow d_M[0] \leftrightarrow d_N[0]$.*

2. *The batch size $L = LCM(M, N)$.*

3. *The last data point in a batch is aligned to both $d_M[n]$ and $d_N[n]$, i.e., $b_i[L] \leftrightarrow d_M[n_1] \leftrightarrow d_N[n_2]$.*

With the definition above, a signal batch b_i has the following properties:

1. There are L/M data points in a batch that are aligned to $d_M[n]$, with index $k \in K_M = \{M,\ 2M,\ 3M, \ldots, L\}$.

2. There are L/N data points in a batch that are aligned to $d_N[n]$, with index $k \in K_N = \{N,\ 2N,\ 3N, \ldots, L\}$.

3. There are $L - L/M - L/N + 1$ data points in a batch that are not aligned to either $d_M[n]$ or $d_N[n]$. This index set is denoted as

$$K_U = \left\{k \in Z^+ \,|\, k < L, \mathrm{mod}(k, M) \neq 0, \mathrm{mod}(k, N) \neq 0\right\} \tag{3.54}$$

For example, for $M = 3$ and $N = 2$, the batch size $L = 6$, $K_M = \{3, 6\}$, $K_N = \{2, 4, 6\}$ and $K_U = \{1, 5\}$. In the case with $M = 8$ and $N = 7$, we have $K_M = \{8, 16, 24, 32, 40, 48, 56\}$, $K_N = \{7, 14, 21, 28, 35, 42, 49, 56\}$ and 42 unmeasured data points exist in K_U.

3.3.2 Signal recovery algorithm

If the time index of the fast underlying signal $b_i[k]$ is aligned to any of the sensor measurements, i.e. $k \in K_M$ or K_N, a direct measurement is available and no data recovery is needed. However, if $k \in K_U$, $b_i[k]$ is lost in the sampling process. The following theorem shows that if $d[n]$ satisfies an autoregressive model, the lost information can be recovered by combining historical measurements form S_1 and S_2.

Theorem 3.4. *Let $d_M[n]$, $d_N[n]$, $d[n]$, and $b_i[k]$ be defined as described in section 3.3.1. If there exists a polynomial $A(z^{-1}) = 1 + \sum_{i=1}^{m} a_i z^{-i}$ ($a_m \neq 0$) such that $A(z^{-1})d[n] = 0$ at the steady state, then the k-th data point in the i-th batch can be recovered by*

$$b_i[k] = \sum_{i=0}^{t_1} w_{k,i} d_M[n_1 - i] + \sum_{j=0}^{t_2} v_{k,j} d_N[n_2 - j], \tag{3.55}$$

where t_1 and t_2 are finite integers, n_1 and n_2 denote indices of d_M and d_N such that $d_M[n_1] \leftrightarrow d_N[n_2] \leftrightarrow b_{i-1}[k]$ (such relationship is ensured by the third rule of Definition 3.1). The unknown parameters $w_{k,i}$'s and $v_{k,j}$'s come from the solution to the following system of linear equations

$$\boldsymbol{M}_k \begin{bmatrix} f_{k,1} \\ \vdots \\ f_{k,l} \\ w_{k,0} \\ \vdots \\ w_{k,t_1} \\ v_{k,0} \\ \vdots \\ v_{k,t_2} \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_m \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.56}$$

Here, $l = max\{t_1 M,\ t_2 N\} + k - m$*;* $\boldsymbol{M}_k$ *is a matrix of dimension* $(l+m) \times (l+t_1+t_2+2)$*, and is defined as*

$$\boldsymbol{M}_k = [\tilde{\boldsymbol{M}}_k\ \boldsymbol{e}_k\ \boldsymbol{e}_{k+M}\ \ldots\ \boldsymbol{e}_{k+t_1 M}\ \boldsymbol{e}_k\ \boldsymbol{e}_{k+N}\ \ldots\ \boldsymbol{e}_{k+t_2 N}], \tag{3.57}$$

where

$$\tilde{\boldsymbol{M}}_k = \begin{bmatrix} 1 & \cdots & 0 \\ a_1 & \ddots & \vdots \\ \vdots & \ddots & 0 \\ a_m & \ddots & 1 \\ 0 & \ddots & a_1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_m \end{bmatrix}_{(l+m)\times l}, \tag{3.58}$$

and $\boldsymbol{e}_i$ *is the elemental column vector whose entries are all zeros except for the* i*-th entry, which equals* 1*.*

Proof. See A.4. □

Consider an example with $M = 3$, $N = 2$ and $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2}$. The model $A(z^{-1})$ has an order of $m = 2$. Based on Definition 3.1, the batch size is chosen as $L = LCM(3, 2) = 6$, then $K_U = \{1, 5\}$. In the recovering process, data points with index $k \in K_U$ in batches of $d[n]$ will be recovered from (3.55). Here we choose $t_1 = t_2 = 1$ (there are more discussions about choosing t_1 and t_2 in the following section), then the recovering equations become:

$$\begin{aligned} b_i[k] = {} & w_{k,0} d_3[n_1] + w_{k,1} d_3[n_1 - 1] \\ & + v_{k,0} d_2[n_2] + v_{k,1} d_2[n_2 - 1], \quad k = 1, 5. \end{aligned} \tag{3.59}$$

When $k = 1$, we have $l = max\{t_1 M,\ t_2 N\} + k - m = 2$. Then parameters $w_{1,0}$, $w_{1,1}$, $v_{1,0}$,

$v_{1,1}$ can be obtained from the solution of

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ a_1 & 1 & 0 & 0 & 0 & 0 \\ a_2 & a_1 & 0 & 0 & 0 & 1 \\ 0 & a_2 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{1,1} \\ f_{1,2} \\ w_{1,0} \\ w_{1,1} \\ v_{1,0} \\ v_{1,1} \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ 0 \\ 0 \end{bmatrix}. \tag{3.60}$$

When $k = 5$ and $l = max\{t_1 M,\ t_2 N\} + k - m = 6$, parameters $w_{5,0}$, $w_{5,1}$, $v_{5,0}$, $v_{5,1}$ can be obtained from the solution of

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2 & a_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_2 & a_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & a_1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_2 & a_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_2 & a_1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{5,1} \\ f_{5,2} \\ f_{5,3} \\ f_{5,4} \\ f_{5,5} \\ f_{5,6} \\ w_{5,0} \\ w_{5,1} \\ v_{5,0} \\ v_{5,1} \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.61}$$

3.3.3 Discussion

Minimal Historical Data Used in Recovery

In Theorem 3.4, $(t_1 + 1)$ data points from $d_M[n]$ and $(t_2 + 1)$ data points from $d_N[n]$ are used in the recovery equation (3.55). In fact, the number of historical data points used in the recovery process is flexible, as we discuss next.

Corollary 3.2. *A necessary condition for the system of equations (3.56) to have a solution*

is

$$t_1 + t_2 \geq m + n_d - 2 \tag{3.62}$$

where

$$n_d = min\left\{\left\lceil\frac{t_1+1}{L/M}\right\rceil,\ \left\lceil\frac{t_2+1}{L/N}\right\rceil\right\}. \tag{3.63}$$

Proof. Recall that a solvable system of linear equations must not be overdetermined, so an obvious necessary condition for (3.56) to have solutions is

$$l + t_1 + t_2 + 2 \geq l + m \tag{3.64}$$

In addition, when $iM = jN$ holds for some $i \in [0,\ t_1]$ and $j \in [0,\ t_2]$, the corresponding columns $\boldsymbol{e}_{k+iM}$ and $\boldsymbol{e}_{k+jN}$ in matrix $\boldsymbol{M}_k$ are identical (see, e.g., the 7th and 9th columns of $\boldsymbol{M}_k$ in (3.61)), yielding redundant pairs of variables in (3.56) (say there are n_d number of them). Then, the number of independent variables becomes $l + t_1 + t_2 + 2 - n_d$ and the necessary condition (3.64) reduces to (3.62).

To more quantitatively define n_d, we recall that a signal batch could provide at most L/M measurements from sensor S_1 and L/N measurements from sensor S_2, hence the number of prior batches used in the recovery that contain measurements from S_1 (denoted as $n_{d,M}$) or S_2 (denoted as $n_{d,N}$) are

$$n_{d,M} = \left\lceil\frac{t_1+1}{L/M}\right\rceil,\ n_{d,N} = \left\lceil\frac{t_2+1}{L/N}\right\rceil. \tag{3.65}$$

It can be seen from Definition 3.1 that the condition $iM = jN$ holds only once in a single batch, then the number of redundant variable pairs n_d is the number of prior batches where measurements from both sensors are involved in the recovery process. That is, $n_d = \min\{n_{d,M},\ n_{d,N}\}$. □

Method to Reduce Computation Complexity

Taking the pseudoinverse inverse of $\boldsymbol{M}_k^\dagger$ gives a particular solution of (3.56):

$$\begin{bmatrix}\boldsymbol{f}_k\\ \boldsymbol{q}_k\end{bmatrix} = \boldsymbol{M}_k^\dagger\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{0}\end{bmatrix}, \tag{3.66}$$

where $\boldsymbol{f}_k = [f_{k,1}, \cdots, f_{k,l}]^T$, $\boldsymbol{q}_k = [w_{k,0}, \cdots, w_{k,t_1}, v_{k,0}, \cdots, v_{k,t_2}]^T$, and $\boldsymbol{a} = -[a_1, \cdots, a_m]^T$. It is worth noting that the computing time of taking pseudoinverse is sensitive with the matrix size. We discuss next an reduced-order procedure to solve (3.56) that will drastically reduce the computation load for real-time applications.

The system of linear equations (3.56) can be rewritten into the following form, where $\boldsymbol{M}_k$ is segmented into four smaller matrices with dimensions defined below.

$$\begin{bmatrix} \boldsymbol{E}_{m\times l} & \boldsymbol{D}_{m\times(t_1+t_2+2)} \\ \boldsymbol{B}_{l\times l} & \boldsymbol{C}_{l\times(t_1+t_2+2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{q}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{0} \end{bmatrix}. \tag{3.67}$$

Then the following reduced-order solution can be obtained by expanding the above the matrix equation:

$$\boldsymbol{q}_k = \left(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{B}^{-1}\boldsymbol{C}\right)^{\dagger} \boldsymbol{a}. \tag{3.68}$$

Instead of directly computing the pseudoinverse of the large matrix M_k, the reduced-order method saves computation cost by reducing the matrix dimension by l in height and width before taking the pseudoinverse. Furthermore, efficient algorithms exist for inverting the upper triangular matrix $\boldsymbol{B}$ [46].

Figure 3.11 shows the changes of the computing cost as k increases when computing the prediction parameters in a batch. Here, the signal model has an order of 6 and $M = 8$, $N = 7$, $t_1 = t_2 = 3$. The tests were performed on a computer (with Intel Core™ i7-6800K CPU and 64GB memory) running MATLAB 2017b. The test results shows that the proposed method reduces the computation costs to a significantly lower level under different configurations; furthermore, the computation cost remains largely invariant when k increases.

Remark 3.1. *The computational complexity of the prediction step (i.e., equation (3.55)) is $O(t_1 + t_2 + 2)$, that is, linear with respect to the total number of historical data used in the prediction. The computational complexity of calculating the prediction parameters (i.e., equation (3.68)) depends mainly on the complexity of inverting the $m \times (t_1 + t_2 + 2)$ matrix $(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{B}^{-1}\boldsymbol{C})$. The prediction parameters can be computed off-line and only need to be recalculated when the signal model changes.*

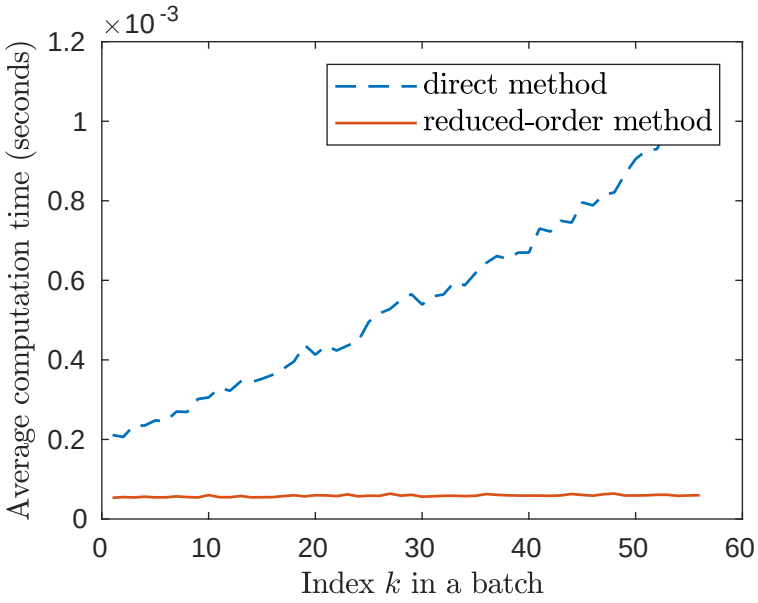


Figure 3.11: The average time for computing the system solution using the direct method and the reduced-order method.

Prediction Robustness and Model Singularity

Let $\boldsymbol{x}_m = [d_M[n_1], \ldots, d_N[n_2], \ldots]^T$ be the measurement vector and $\Delta\boldsymbol{x}$ be the measurement noise vector. Then the prediction equation (3.55) can be formulated as $d[n] = \boldsymbol{q}_k(\boldsymbol{x}_m^T + \Delta\boldsymbol{x}^T) = \boldsymbol{q}_k\boldsymbol{x}^T + \left(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{B}^{-1}\boldsymbol{C}\right)^\dagger \Delta\boldsymbol{x}^{\boldsymbol{T}}$ where $\boldsymbol{q}_k\boldsymbol{x}^{\boldsymbol{T}}$ is the true value. The prediction error cased by the measurement noise is thus

$$e = \left(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{B}^{-1}\boldsymbol{C}\right)^\dagger \Delta\boldsymbol{x}^{\boldsymbol{T}}. \tag{3.69}$$

It is well understood that if the matrix $\left(\boldsymbol{D} - \boldsymbol{E}\boldsymbol{B}^{-1}\boldsymbol{C}\right)$ is ill-conditioned (i.e., close to becoming singular), then its inversion will contain larger numbers, and consequently, the measurement noise will be amplified.

In this section, we discuss the extreme case and provide solutions to mitigate the model singularity. To see this, we first note that the signal model $A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_m z^{-m}$ defines a linear function between a data point $d[n]$ and its m consecutive prior data points:

$d[n] = -\sum_{i=1}^{m} a_m d[n-m]$. An essential part of our proposed recovery algorithm is to re-parameterize the signal model and build a new connection between $d[n]$ and prior data points available from sparse sensor measurements (recall Eqns. (3.55), (A.23), (A.26) and (A.27)). Figure 3.12 illustrates the model re-parameterizing process when recovering the first point in

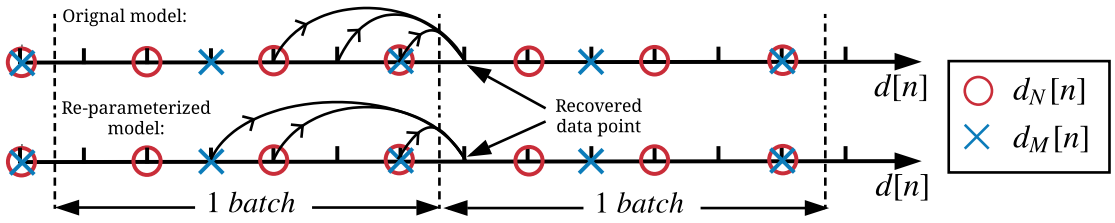


Figure 3.12: Illustration of model re-parameterizing for sparse information recovery.

a batch, under the configuration of $M = 3$, $N = 2$, $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}$, and $t_1 = t_2 = 1$. If the signal model $A(z^{-1})$ contains scarce connections between $d[n]$ and its prior data points, the new connection would be difficult or even impossible. For example, consider $A(z^{-1}) = 1 + a_6 z^{-6}$, that is, $d[n] = -a_6 d[n-6]$. As shown in Figure 3.13, when $M = 3$, $N = 2$, or $M = 4$, $N = 3$, there exist missing data points in a batch that are unrelated to any other measurements. Thus it becomes impossible to find the re-parameterized model that can recover the missing data point.

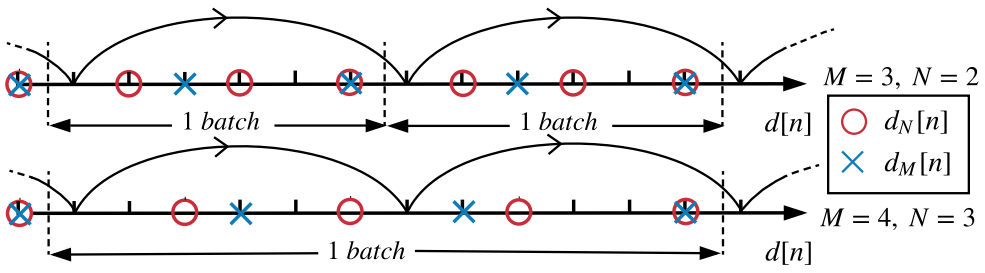


Figure 3.13: An example when a missing data point is unrelated to any measurements.

More generally, consider a signal model:

$$A_Q(z^{-1}) = 1 + \sum_{i=1}^{t} a_{iQ} z^{-iQ}, \tag{3.70}$$

where $Q \in \mathbb{Z}^+$, and at least one a_{iQ} is nonzero. Then $d[n]$ are only connected to its iQ-th prior data points:

$$d[n] = -\sum_{i=1}^{t} a_{iQ} d[n - iQ]. \tag{3.71}$$

Suppose $d[n] \leftrightarrow b_i[k]$ is the missing point in a batch that needs to be recovered (i.e. $k \in K_U$). If $d[n - iQ]$ is not picked up by the sensors for all $i = 1, 2, 3, \cdots$, then $d[n]$ is impossible to recover. Mathematically, given the batch configuration with M, N and $L = \text{LCM}(M, N)$, if Q satisfies the condition:

$$\text{Exist } k \in K_U, \text{ such that for any } i \in \mathbb{Z}, \ \text{mod}(k + iQ, L) \in K_U \tag{3.72}$$

then a signal with model $A_Q(z^{-1})$ will not be fully recoverable.

Remark 3.2. *If both (Q, M) and (Q, N) are not coprime pairs*[4]*, then condition (3.72) is satisfied.*

Proof. Because (Q, M) is not coprime, they have a common divisor (denoted as p) that is greater than 1. For any integers i, j,

$$Mj - Qi = p\left(\frac{Mj}{p} - \frac{Qi}{p}\right) \neq 1 \tag{3.73}$$

or equivalently,

$$\text{mod}(1 + iQ, M) \neq 0. \tag{3.74}$$

Let $c = \text{mod}(k + iQ, L)$, then we have $c = k + iQ - nL$, where $n \in \mathbb{Z}$. Because L is a multiple of M, $\text{mod}(c, M) = \text{mod}(k + iQ - nL, M) = \text{mod}(k + iQ, M)$. Therefore, for $k = 1 \in K_U$,

$$\text{mod}(\text{mod}(k + iQ, L), M) = \text{mod}(k + iQ, M) \neq 0. \tag{3.75}$$

[4] (A, B) is a coprime pair if and only if their greatest common divisor is 1.

Similarly, since (Q, N) is not coprime,

$$\mathrm{mod}(\mathrm{mod}(k + iQ, L), N) = \mathrm{mod}(k + iQ, N) \neq 0. \tag{3.76}$$

Combining (3.75) and (3.76), we conclude that $\mathrm{mod}(k + iQ, L) \in K_U$ for any integer i and for $k = 1$, so condition (3.72) is satisfied. □

In practice, the discussed model singularities are uncommon and can be overcome by designing the sensor pair. For example, given $M = 3$ and $N = 2$, the smallest Q such that both (Q, M) and (Q, N) are not coprime pairs is 6, then (3.70) becomes $A_Q(z^{-1}) = 1 + \sum_{i=1}^{t} a_{6i} z^{-6i}$. All signal models other than $A_Q(z^{-1})$ will be sufficient for the proposed recovering algorithm with $M = 3$ and $N = 2$. Selecting $M = 5$ and $N = 2$, on the other hand, provides a feasible solution.

Signal Pre-filtering

To improve the recovery accuracy in presence of measurement noise, we propose in this subsection a filter design to pass through signals $d_M[n]$ and $d_N[n]$ without modifying their amplitude and phase, while rejecting noise signals in a board frequency range. Given the autoregressive model $A(z^{-1})$ for $d[n]$, we could find the corresponding autoregressive model $A_M(z^{-1})$ and $A_N(z^{-1})$ for downsampled signals $d_M[n]$ and $d_N[n]$, respectively.

$$A_K(z^{-1}) = \prod_{p=0}^{K-1} A(z^{-\frac{1}{K}} \mathrm{e}^{i\frac{2\pi}{K}p}),\ K = M \text{ or } N. \tag{3.77}$$

These results are based on the transfer functions of the downsampled signals, and obtained according to Lemma 2.1. Then the AR model for the downsampled signals can be used in the filter design for model-based noise compensation.

The proposed noise attenuation is to pre-filter $d_K[n]$ ($K = M$ or N) by

$$G_K(z) = \frac{A_\alpha(z^{-1}) - A_K(z^{-1})}{A_\alpha(z^{-1})} \tag{3.78}$$

and feed the filtered results to the collaborative sensor fusion. Here, $A_K(z^{-1})$ is the internal model of $d_K[n] = d[Kn]$ and $A_\alpha(z^{-1})$ is to be designed. Such a $G_K(z^{-1})$ can pass a signal

$d_K[n]$ with an internal model $A_K(z^{-1})$ as long as the roots of $A_\alpha(z^{-1})$ are in the unit circle. To be more specific, we calculate

$$G_K(z)d_K[n] = \frac{A_\alpha(z^{-1})d_K[n] - \cancel{A_K(z^{-1})d_K[n]}}{A_\alpha(z^{-1})} = d_K[n] \tag{3.79}$$

at the steady state. Let the polynomial $A_K(z^{-1})$ have an order of m, then it can be decomposed to the product of m first-order polynomials

$$A_K(z^{-1}) = \prod_{i=1}^{m}(z^{-1} - \lambda_i e^{j\theta_i}), \tag{3.80}$$

where $\{\lambda_i e^{j\theta_i}\}_{i=1,\dots,m}$ are the roots of $A_K(z^{-1})$. For the purpose of reducing the measurement noise energy, it is important to maintain a small value to $|G_K(e^{j\omega})|$ (i.e., the magnitude of the frequency response of $G_K(z)$) at the noise frequencies. To that end, we design $A_\alpha(z^{-1})$ as

$$A_\alpha(z^{-1}) = \prod_{i=1}^{m}(z^{-1} - \alpha\lambda_i e^{j\theta_i}), \tag{3.81}$$

where $\alpha \in (0, 1)$ and is close to 1. Figure 3.14 shows the difference of $|G_K(e^{j\omega})|$under different

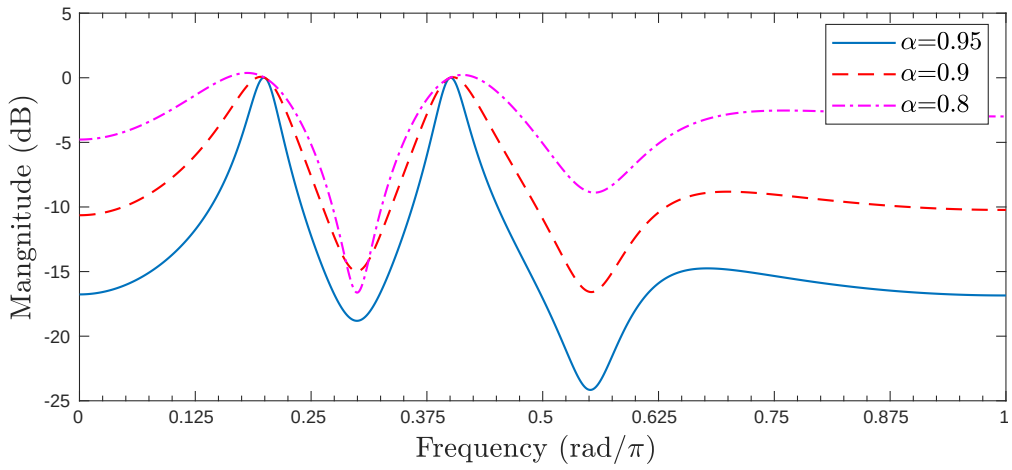


Figure 3.14: Magnitude of $G(e^{j\omega})$ when choosing different values of α in (3.81).

α selections. Here $A(z^{-1})$ has 6 roots located at $\{e^{\pm j\pi/5}, e^{\pm j2\pi/5}, 0.9e^{\pm j3\pi/5}\}$. It is shown that $|G_K(e^{j\omega})|$ is maintained under 0dB at frequencies other than the desired signal frequencies

(0.2π and 0.4π), so the noise energy can be reduced. When α is closer to 1, $|G_K(e^{j\omega})|$ will be smaller in a wide frequency range, which yields a better noise reduction rate. As a trade-off, $G_K(z)$ with α closer to 1 has a slower converging speed (see Fig. 3.15).

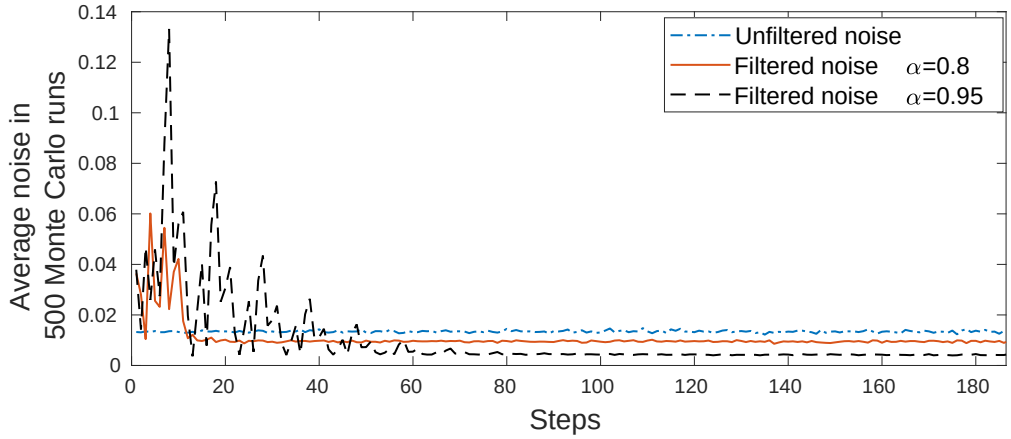


Figure 3.15: Measurement noise before and after filtering by $G(z)$ when choosing a different α.

3.3.4 Numerical verification

Figure 3.16 shows an numerical simulation of the signal recovering process. The recovered measurement $d[n]$ is sampled at 1kHz, and the two slow measurements $d_M[n]$ and $d_N[n]$ are sampled at 1/8kHz and 1/7kHz, respectively. In the simulation, the original signal $d_c(t)$ is narrow-banded and contains four frequency components at 120Hz, 167Hz, 240Hz, and 300Hz. Such frequency components are much higher than the Nyquist sampling frequencies of slow measurements, that is, 62.5Hz for $d_M[n]$ and 71.4Hz for $d_N[n]$. Hence the discretized measurements will lose the original shape of the trajectory (see the upper plot of Fig. 3.16) due to aliasing in the sampling process.

Based on the autoregressive model of a narrow-band signal $d[n]$ with n frequency com-

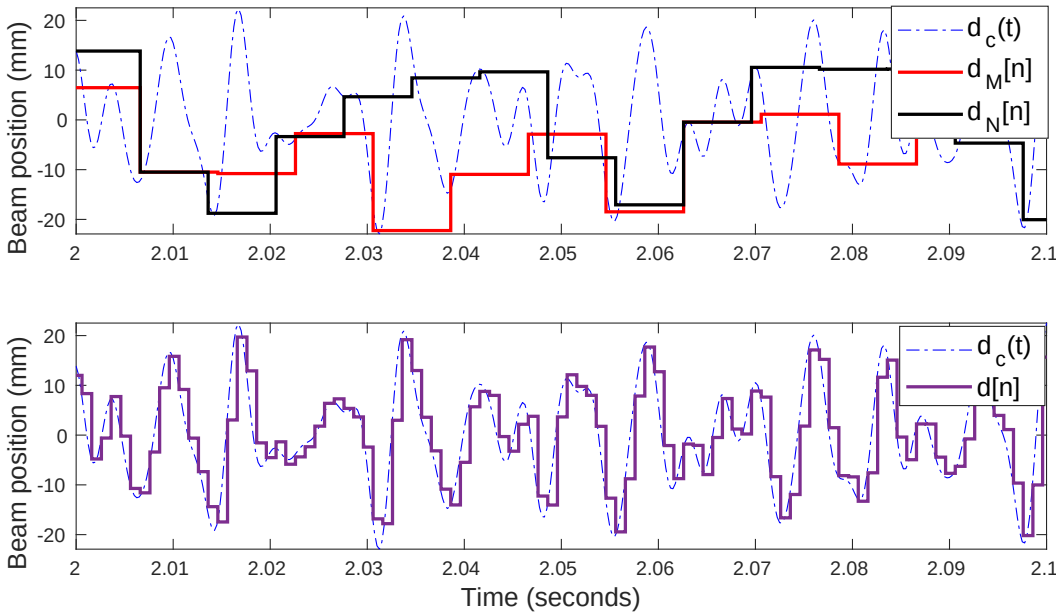


Figure 3.16: Collaborative sensor measurements and the recovered signal for optical beam scanning in additive manufacturing.

ponents f_i, $i = 1, \ldots, n$, we have $A(z^{-1})d[n] = 0$ at the steady state, where

$$A(z^{-1}) = \prod_{i=1}^{n}(1 - 2\cos(2\pi f_i T)z^{-1} + z^{-2}) \tag{3.82}$$

and T is the sampling time of $d[n]$. Substituting in the frequency values yields the polynomial model of $d[n]$: $A(z^{-1}) = 1 - 1.9619z^{-1} + 4.1664z^{-2} - 4.9797z^{-3} + 6.2201z^{-4} - 4.9797z^{-5} + 4.1664z^{-6} - 1.9619z^{-7} + z^{-8}$. Figure 3.16 shows the slow measurements $d_M[n]$ and $d_N[n]$ as well as the reconstructed signal $d[n]$. As shown in the bottom plot in Fig. 3.16, the recovered data in $d[n]$ matches with its true value.

Chapter 4

ROBOTIC TARGET FOLLOWING UNDER SLOW AND DELAYED SENSING

This chapter considers the problem of controlling a robot to follow and track a moving target based on only visual feedback. We aim to compensate the effect of slow-sampling and time delays in vision feedback. This compensation is made possible by multi-rate model-based prediction (MMP), an information recovery algorithm based on autoregressive model re-parameterization.

4.1 Formulation of the target following problem

Coordinate system definition. The target following system includes a moving target T, a follower robot R, and a vision system C statically attached to the robot. We define a static world coordinate system $\{W\}$ and attach moving coordinate systems $\{T\}$ and $\{C\}$ to the target and to the vision system, respectively. Figure 4.1 shows the relationship between the coordinate systems, where $\{C^*\}$ represents the desired position and orientation of the vision system.

We now discuss the major elements in Figure 4.1:

- The observed target pose in the camera frame ${}^C\xi_T$: There are many approaches to estimate the target pose from images. For example, with a calibrated camera and a known 3D model of the target, ${}^C\xi_T$ can be obtained from the solution of Perspective-n-Point (PnP) problem based on n 3D-to-2D point correspondences [47, 48]. With a stereo camera or an RGB-D camera, it is possiable to estimated the 3D pose of the target without using the 3D model of the target as a prior [49, 50]. There are also learning-based pose estimation proposed recently [51]. In this work, we use the PnP

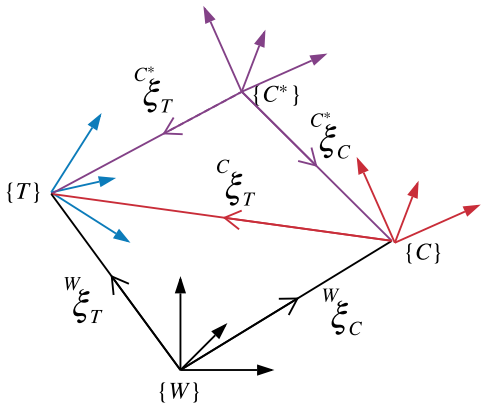


Figure 4.1: Coordinate systems of target following problem

algorithm for simplicity and will assume an estimation ${}^{C}\hat{\xi}_T$ is available hereafter.

- The camera pose in the world coordinate system ${}^{W}\xi_C$: Because the camera is rigidly attached to the robot, ${}^{W}\xi_C$ can be calculated by the forward kinematics of the robot.

- The target pose in the world frame ${}^{W}\xi_T$: This is unknown and is estimated by the vision system. Most important, we consider the case when the target is moving with unknown velocity ${}^{W}\boldsymbol{v}_T(t)$.

Data acquisition and time delay. We consider the case when the image-based pose estimation ${}^{C}\hat{\xi}_T$ has a time delay of τ seconds, and is updated every T_{ss} seconds. However, the controller of the robot is running with a much shorter period of T_{sf} seconds. We assume that the sampling rate of the imaging system can be adjusted such that

$$T_{ss} = LT_{sf}, \tag{4.1}$$

where L is an integer greater than one.

Control goal. Similar to the standard PBVS [3], the vision-based target tracking problem can be formulated as minimizing the visual feature error $\boldsymbol{e}(t)$ given by

$$\boldsymbol{e}(t) = \boldsymbol{s}(t) - \boldsymbol{s}^*(t), \tag{4.2}$$

where $\boldsymbol{s}$ represents a set of visual features and $\boldsymbol{s}^*$ is the desired value of $\boldsymbol{s}$. For the target following problem, we define the visual featuer as the pose difference between the desired and the actual camera pose $\boldsymbol{s} \triangleq {}^{C^*}\xi_C$. An estimation of the visual feature is calculated from the target pose estimation

$$ {}^{C^*}\hat{\xi}_C = {}^{C^*}\xi_T \oplus {}^{T}\hat{\xi}_C, \tag{4.3}$$

where ${}^{C^*}\xi_T$ is a user-defined value and ${}^{T}\hat{\xi}_C$ is the reverse of the pose estimation ${}^{C}\hat{\xi}_T$. The goal of target following is to control the camera velocity ${}^{C}\boldsymbol{v}_C$ such that ${}^{C}\xi_T$ will converge to ${}^{C^*}\xi_T$, that is, $\boldsymbol{s}$ converges to $\boldsymbol{s}^* = \mathbf{0}$.

4.2 Position-based visual servo with a moving target

We parameterize the feature error as $\boldsymbol{e} = {}^{C^*}\xi_C = ({}^{C^*}\boldsymbol{t}_C,\ \theta\boldsymbol{u})$, where ${}^{C^*}\boldsymbol{t}_C$ and θ are the translation vector and the rotation angles from $\{C^*\}$ to $\{C\}$, respectively. $\boldsymbol{u}$ is a unit vector representing the corresponding rotation axis. Let the relative instantaneous velocity of the camera with respect to the target be ${}^{T}\boldsymbol{v}_C \in \mathbb{R}^{6\times1}$, then the relationship between the time derivative of visual feature error $\dot{\boldsymbol{e}}$ and the relative velocity ${}^{T}\boldsymbol{v}_C$ is

$$ \dot{\boldsymbol{e}} = \boldsymbol{L}_e {}^{T}\boldsymbol{v}_C = \begin{bmatrix} {}^{C^*}\boldsymbol{R}_T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{L}_{\theta\boldsymbol{u}} {}^{C}\boldsymbol{R}_T \end{bmatrix} {}^{T}\boldsymbol{v}_C, \tag{4.4}$$

where $\boldsymbol{L}_e$ is the feature Jacobian and $\boldsymbol{L}_{\theta\boldsymbol{u}}$ is given by

$$ \boldsymbol{L}_{\theta\boldsymbol{u}} = \boldsymbol{I}_3 - \frac{\theta}{2}[\boldsymbol{u}]_\times + \left(1 - \frac{\mathrm{sinc}\theta}{\mathrm{sinc}^2\frac{\theta}{2}}\right)[\boldsymbol{u}]_\times^2. \tag{4.5}$$

Here, $\mathrm{sinc}(x)$ is the sinus cardinal such that $x\mathrm{sinc}(x) = \sin(x)$ and $\mathrm{sinc}(0)$=1. $[\boldsymbol{u}]_\times$ is a skew-symmetric matrix defined as

$$ [\boldsymbol{u}]_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}. \tag{4.6}$$

To ensure a decreasing feature error, we design a velocity controller

$$ {}^{T}\boldsymbol{v}_C = \text{-}\lambda \boldsymbol{L}_e^{-1}\boldsymbol{e} = \text{-}\lambda \begin{bmatrix} {}^{T}\boldsymbol{R}_{C^*} & \mathbf{0} \\ \mathbf{0} & {}^{T}\boldsymbol{R}_C \boldsymbol{L}_{\theta\boldsymbol{u}}^{-1} \end{bmatrix} \begin{bmatrix} {}^{C^*}\boldsymbol{t}_C \\ \theta\boldsymbol{u} \end{bmatrix}, \tag{4.7} $$

where $\lambda > 0$. Then from Eq. (4.7) and Eq. (4.4):

$$ \dot{\boldsymbol{e}} = \text{-}\lambda \boldsymbol{e}. \tag{4.8} $$

That is to say, the visual feature error will exponentially decrease to zero. Note that $\boldsymbol{L}_{\theta\boldsymbol{u}}^{-1}\theta\boldsymbol{u} = \theta\boldsymbol{u}$. Eq. (4.7) can then be simplified as

$$ \begin{cases} {}^{T}\boldsymbol{v}_{C,l} = \text{-}\lambda \cdot {}^{T}\boldsymbol{R}_{C^*}\, {}^{C^*}\boldsymbol{t}_C \\ {}^{T}\boldsymbol{v}_{C,a} = \text{-}\lambda\theta \cdot {}^{T}\boldsymbol{R}_C \boldsymbol{u} \end{cases} \tag{4.9} $$

where ${}^{T}\boldsymbol{v}_{C,l}$ and ${}^{T}\boldsymbol{v}_{C,a}$ are the linear and the angular parts of the velocity vector ${}^{T}\boldsymbol{v}_C$, respectively.

Note that in the standard PBVS, the feature Jacobian is defined with respect to the camera's velocity in the camera frame ${}^{C}\boldsymbol{v}_C$ [3], i.e., $\dot{\boldsymbol{e}} = \boldsymbol{L}_e\, {}^{C}\boldsymbol{v}_C$. Such a relationship is valid only if the target is motionless. In this work, the feature Jacobian is defined with respect to the relative velocity ${}^{T}\boldsymbol{v}_C$ between two moving frames, such that Eq. (4.4) is still valid when the target is maneuvering. To compute the velocity command ${}^{C}\boldsymbol{v}_C$, we use a velocity transformation formula

$$ {}^{C}\boldsymbol{v}_C = \begin{bmatrix} {}^{C}\boldsymbol{R}_T & \mathbf{0} \\ \mathbf{0} & {}^{C}\boldsymbol{R}_T \end{bmatrix} {}^{T}\boldsymbol{v}_C + \begin{bmatrix} {}^{C}\boldsymbol{R}_T & \text{-}\,{}^{C}\boldsymbol{R}_T \left[{}^{T}\boldsymbol{t}_C\right]_\times \\ \mathbf{0} & {}^{C}\boldsymbol{R}_T \end{bmatrix} {}^{T}\boldsymbol{v}_T. \tag{4.10} $$

Here, the target velocity ${}^{T}\boldsymbol{v}_T$ is obtained from Kalman filters, as is described in the following section.

4.3 Pose and velocity estimation

While we can estimate the target pose from the perception system, the visual measurements are usually noisy. To reduce the measurement noise and to estimate the unknown target velocity, we build a set of Kalman filters that can fit to common robotic target motions.

4.3.1 Dynamic models of the moving target

We list first the widely used constant velocity model and constant acceleration model for a moving target, then derive a constant frequency model and a compound constant frequency model.

Constant velocity model. When the acceleration is relatively small, we can assume a constant speed in the state update equation and model the acceleration as white noise, which formulates to the *discrete white noise acceleration (DWNA)* model [52]. Define the system state as $x = (\eta, \dot{\eta})$. Then the system state-space model is given by

$$x(k+1) = \begin{bmatrix} 1 & T_{ss} \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_{ss}^2 \\ T_{ss} \end{bmatrix} v(k), \tag{4.11}$$

$$z(k+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k+1) + \omega(k+1), \tag{4.12}$$

where $v(k)$ is the process noise and $\omega(k+1)$ is the measurement noise.

Constant acceleration model. When the acceleration is nearly constant, we use *discrete Wiener process acceleration (DWPA)* model that assumes a constant acceleration within each update period. Defining the system state as $x = (\eta, \dot{\eta}, \ddot{\eta})$, the system model is given as

$$x(k+1) = \begin{bmatrix} 1 & T_{ss} & \frac{1}{2}T_{ss}^2 \\ 0 & 1 & T_{ss} \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_{ss}^2 \\ T_{ss} \\ 1 \end{bmatrix} v(k), \tag{4.13}$$

$$z(k+1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(k+1) + \omega(k+1). \tag{4.14}$$

Constant frequency (CF) model. When the movement is approximately periodic, both constant velocity model and constant acceleration model will fail to timely capture the velocity and acceleration changes. This model mismatch is significant when the moving frequency is fast. The vast amount of periodic tasks can be decomposed to sinusodial motions from the Fourier theory. Consider $\eta(t) = a\sin(\Omega t + \varphi)$ and its derivative $\dot{\eta}(t) = a\Omega\cos(\Omega t +$

φ). Let the system state be $x = (\eta, \dot{\eta})$. Note that the second derivative $\ddot{\eta}(t) = -a\Omega^2 \sin(\Omega t + \varphi) = -\Omega^2 \eta(t)$. Then we can write the continuous-time state-space model as

$$\frac{d}{dt}x(t) = \begin{bmatrix} 0 & 1 \\ -\Omega^2 & 0 \end{bmatrix} x(t) \triangleq A_c x(t). \tag{4.15}$$

Discretizing Eq. (4.15) at a sampling time T_{ss} , we have

$$x(k+1) = \mathrm{e}^{A_c T_{ss}} x(k) = \begin{bmatrix} \cos(\Omega T_{ss}) & \sin(\Omega T_{ss})/\Omega \\ -\Omega \sin(\Omega T_{ss}) & \cos(\Omega T_{ss}) \end{bmatrix} x(k). \tag{4.16}$$

When the sinusoidal signal has a bias term, i.e., $\eta(t) = a \sin(\Omega t + \varphi) + b$, we can augment the system to include the bias: $x = (\eta, \dot{\eta}, b)$. The corresponding extended model with noise is

$$x(k+1) = \begin{bmatrix} \cos(\Omega T_{ss}) & \sin(\Omega T_{ss})/\Omega & 0 \\ -\Omega \sin(\Omega T_{ss}) & \cos(\Omega T_{ss}) & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_{ss}^2 \\ T_{ss} \\ T_{ss} \end{bmatrix} v(k), \tag{4.17}$$

$$z(k+1) = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} x(k+1) + \omega(k+1). \tag{4.18}$$

Compound constant frequency (CCF) model. We consider here the case when the core movement is a mixture of multiple sinusoids, i.e.,

$$\eta(t) = \sum_{i=1}^{N} a_i \sin(\Omega_i t + \varphi_i) + b, \ \Omega_i \neq \Omega_j \neq 0; \forall i \neq j \tag{4.19}$$

In order to model the kinematics of the above, let $x = (\eta_1, \dot{\eta}_1, \ldots, \eta_N, \dot{\eta}_N, b)$. The state is defined such that the i-th pair $(\eta_i, \dot{\eta}_i)$ corresponds to the i-th frequency component, and has the same discrete model as in Eq. (4.16). Each $(\eta_i, \dot{\eta}_i)$ is independent from other pairs, thus

we have the following state-space model:

$$x(k+1) = \begin{bmatrix} \boldsymbol{A}_1 & & & & \\ & \boldsymbol{A}_2 & & & \\ & & \ddots & & \\ & & & \boldsymbol{A}_N & \\ & & & & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_{ss}^2 \\ T_{ss} \\ \vdots \\ \vdots \\ T_{ss} \end{bmatrix} v(k) \tag{4.20}$$

$$z(k+1) = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & 1 \end{bmatrix} x(k+1) + \omega(k+1) \tag{4.21}$$

where $\boldsymbol{A}_i$ is a 2×2 matrix defined as

$$\boldsymbol{A}_i = \begin{bmatrix} \cos(\Omega_i T_{ss}) & \sin(\Omega_i T_{ss})/\Omega_i \\ \text{-}\Omega_i \sin(\Omega_i T_{ss}) & \cos(\Omega_i T_{ss}) \end{bmatrix}, \quad i = 1, \cdots, N. \tag{4.22}$$

4.3.2 Linear position and velocity estimation

We decouple the 3D target motions to x, y and z axes and estimate the components independently. As a result, the system order can be reduced, and we only need to consider the problem of estimating the position and velocity in one generic axis. Recall that a relative pose measurement ${}^{C}\hat{\xi}_T$ sampled at $1/T_{ss}$ Hz is available from the vision system. Combined with the known camera pose ${}^{W}\xi_C$, the target pose measurements in the world coordinate system can be obtained as ${}^{W}\xi_T = {}^{W}\xi_C \oplus {}^{C}\hat{\xi}_T$. Then the position measurement ${}^{W}\boldsymbol{t}_T = (t_x, t_y, t_z)$ can be extracted from ${}^{W}\xi_T$. For the motion in each axis, we choose the appropriate dynamic model based on an assessment of the motion type.

$$x(k+1) = Fx(x) + v(k) \tag{4.23}$$

$$z(k+1) = Hx(k+1) + w(k) \tag{4.24}$$

Denote the process noise covariance as Q and the measurement noise covariance as R. Kalman filter prediction and update steps, Eqs. (4.25-4.31), are applied to filter the noisy measurement and estimate the position and velocity for each axis. In the prediction step,

we first compute the estimated state $x(k+1|k)$ and state covariance $P(k+1|k)$.

$$\hat{x}(k+1|k) = F\hat{x}(k|k) \tag{4.25}$$

$$P(k+1|k) = FP(k|k)F' + Q \tag{4.26}$$

Here, $\bullet(k+1|k)$ indicates estimated variables given measurements up to and includeing $z(k)$. Similarly, $\bullet(k+1|k+1)$ indicates estimates given measurement includeing $z(k+1)$. The measurement prediction $\hat{z}(k+1|k)$ and the corresponding prediction covariance $S(k+1)$ are

$$\hat{z}(k+1|k) = H\hat{x}(k+1|k) \tag{4.27}$$

$$S(k+1) = R + HP(k+1|k)H' \tag{4.28}$$

In the correction step, we compute the Kalman filter gain $K(k+1)$ and the updated state estimation and covairance using latest measurement $z(k+1)$.

$$W(k+1) = P(k+1|k)H'S(k+1)^{-1} \tag{4.29}$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)(z(k+1) - \hat{z}(k+1|k)) \tag{4.30}$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \tag{4.31}$$

4.3.3 Angular position and velocity estimation

In order to build Kalman filters for estimating the rotation angles and velocities, we first extract the measured rotation matrix ${}^{W}\boldsymbol{R}_{T}$ from ${}^{W}\xi_{T}$, then convert it to Euler angles (α, β, γ). Note that there are twelve different representations of the Euler angles and any one of them can be used. Here we use the roll-pitch-yaw representation (i.e., with the rotation order "ZYX") that is popular for ships, aircraft and vehicles. Similar to the linear position estimation, we view each Euler angle as an individual component and apply Eqs. (4.25-4.31) with the appropriate dynamic model. Note that the physical meanings of the angular velocity estimates $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$ depend on the choice of Euler representation. For example, in the roll-pitch-yaw representation, $\dot{\beta}$ and $\dot{\gamma}$ are the angular velocities about the rotated y and

Dynamic model	Position	Velocity
DWNA	$(1-z^{-1})^2$	$1-z^{-1}$
DWPA	$(1-z^{-1})^3$	$(1-z^{-1})^2$
CF	$1-2\cos(\Omega T_{sf})z^{-1}+z^{-2}$	
CCF	$\prod_{i=1}^{N}(1-2\cos(\Omega_i T_{sf})z^{-1}+z^{-2})$	

Table 4.1: Polynomial models of the position and velocity profiles for different dynamic models

x-axes instead of the original ones. We obtain the angular velocity with respect to the world coordinate system ${}^{W}\boldsymbol{v}_{T,a} = (\omega_x, \omega_y, \omega_z)$ by coordinate conversion:

$$ {}^{W}\boldsymbol{v}_{T,a} = \dot{\alpha}\overrightarrow{\boldsymbol{z}} + R_z(\alpha)\left(\dot{\beta}\overrightarrow{\boldsymbol{y}} + R_y(\beta)\dot{\gamma}\overrightarrow{\boldsymbol{x}}\right). \tag{4.32} $$

Expanding Eq. (4.32), we have

$$ \begin{cases} \omega_x &= \dot{\gamma}\cos(\alpha)\cos(\beta) - \dot{\beta}\sin(\alpha) \\ \omega_y &= \dot{\beta}\cos(\alpha) + \dot{\gamma}\cos(\beta)\sin(\alpha) \\ \omega_z &= \dot{\alpha} - \dot{\gamma}\sin(\beta) \end{cases} \tag{4.33} $$

4.4 Interpolation and delay compensation

The target position and velocity estimated from the Kalman filter are only sampled at $f_{ss} = 1/T_{ss}$ Hz. They also inherit the time delays from the visual measurements. With MMP (see section 2), we can construct new data points from the slow sampled Kalman filter outputs and also compensates the time delays.

Table 4.1 summarizes the polynomial models of the position and velocity profiles for each dynamic model mentioned in section 4.3.1. These can be derived from Lemma 2.1.

4.4.1 Interpolation and delay compensation algorithm

As is shown before, the model-based information recovery can construct a connection between the historical slow-sampled data set (i.e., $d_s[n]$, $d_s[n-1]$ and so on) and a future data point $d_f[nL+k]$ in the fast-sampled sequence. This connection allows to predict the position and velocity sampled at $1/T_{sf}$ Hz using Kalman filter outputs. Furthermore, by properly adjusting the k value, the measurement delay time τ can be compensated. In more details, at each discrete step, we first calculate the index k_c such that $nL+k_c$ corresponds to the current time t_c (see Figure 4.2). Then we calculate the length of prediction steps $k_p = \text{round}(\tau/T_{sf})$ needed to compensate the delay time τ. Finally we use $k = k_c + k_p$ and apply Eq. (3.11). The algorithm to compute the fast-sampled data $\hat{d}_f$ with delay compensated, namely model-based prediction (MMP), is summarized in Algorithm 1.

Algorithm 1: Interpolation and delay compensation (one circle)

input : most recent $d_s[n]$, delay time τ and current time t_c

output: $\hat{d}_f[n]$

if $d_s[n]$ *has a new value* **then**
  update the data storage that keeps the most m recent $d_s[n]$ values $\{d_s[n], \cdots, d_s[n-m+1]\}$;
  $t_{\text{last}} \leftarrow t_c$;
end if
$k_c \leftarrow (t_c - t_{last})/T_{sf}$;
$k_p \leftarrow \text{round}(\tau/T_{sf})$;
$k \leftarrow k_c + k_p$;
From Eq. (3.12), solve $w_{k,0}, \cdots, w_{k,m-1}$;
From Eq.(3.11), compute $\hat{d}_f[n]$;

A simulated interpolation and delay compensation result is shown in Figure 4.3. In the simulation, the fast and slow sampling times are $T_{sf} = 8$ms and $T_{ss} = 48$ms, respectively.

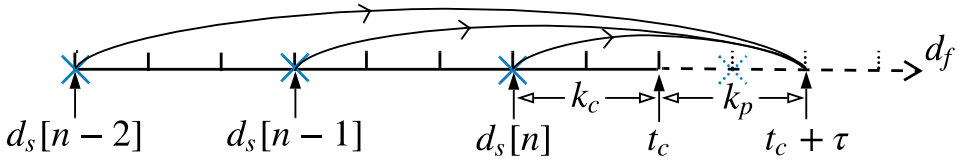


Figure 4.2: Illustration of the interpolation and delay compensation procedure. Here, $L = T_{ss}/T_{sf} = 3$.

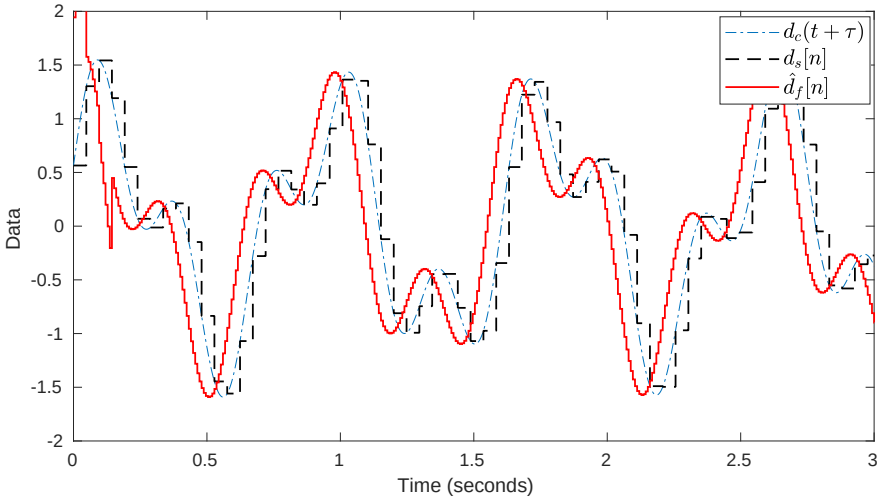


Figure 4.3: Simulated interpolation and delay compensation result.

Thus $L = T_{ss}/T_{sf} = 6$. The polynomial model used here corresponds to the CCF dynamic model with two frequencies $\Omega_1 = 2\pi \times 1.2$ rad/s and $\Omega_2 = 2\pi \times 3.1$ rad/s. Based on Table 4.1, $A(z^{-1})$ can be derived as $1 - 3.97z^{-1} + 5.94z^{-2} - 3.97z^{-3} + z^{-4}$. The results show that MMP can accurately predict the intersample data and compensate the measurement delay (56ms in this case).

4.5 Numerical simulation

In this subsection, we simulate the performances of the proposed target following algorithm under various scenarios. The simulated algorithm combines the PBVS considering a moving target (see Section 4.2), Kalman filters (see Section 4.3) and the interpolation and delay compensation using MMP (see Section 4.4). An overview of the system structure is shown in Figure 4.4. In the simulation, the target moves following a fixed 2D track as is showing in the Figure 4.5. The target started from position $(0, 0)$ and then went through the full circle multiple times. Here, the target pose measurement from the pose estimation block is simulated by adding a zero-mean Gaussian noise to the true target pose. The noises are independently distributed across each axis and have a standard deviation of 0.01 meters. To exclude the randomness in each simulation runs, we did 50 independent simulation runs for each scenario and use the averaged following error for comparison.

In the first scenario, the robot controller loop is running at 125Hz (i.e., $T_{sf} = 8$ milliseconds). However, the pose estimation from the vision system is running eight times slower and is delayed by 64 milliseconds. Figure 4.6 shows the tracking error statistics for the first four laps. Here, we also provide simulation results when using the basic PBVS (see [3]) or our modified PBVS (see Section 4.2) but without interpolation and delay compensation. In the first lap, we observed that the following error reduced from its initial value to a significant lower level. This can be seen in the error bars where the maximum errors in the first lap all start with around 3 meters and the minimum errors are below 0.1 meters. In the following laps, the simulation all transient to a steady-state where the following errors are constrained in a fixed range. The average tracking errors using the basic PBVS is 87 millimeters in the steady-state, which is significant considering that the track dimension is only 320 millimeters by 160 millimeters. Implementing our improved PBVS which considered a moving target could reduce the following error. However, the error is still large due to the slow-sampling and delays in visual feedback. Finally, using our proposed approach could reduce the errors to a small range around 3 millimeters, yielding a 95% error reduction compared to the basic

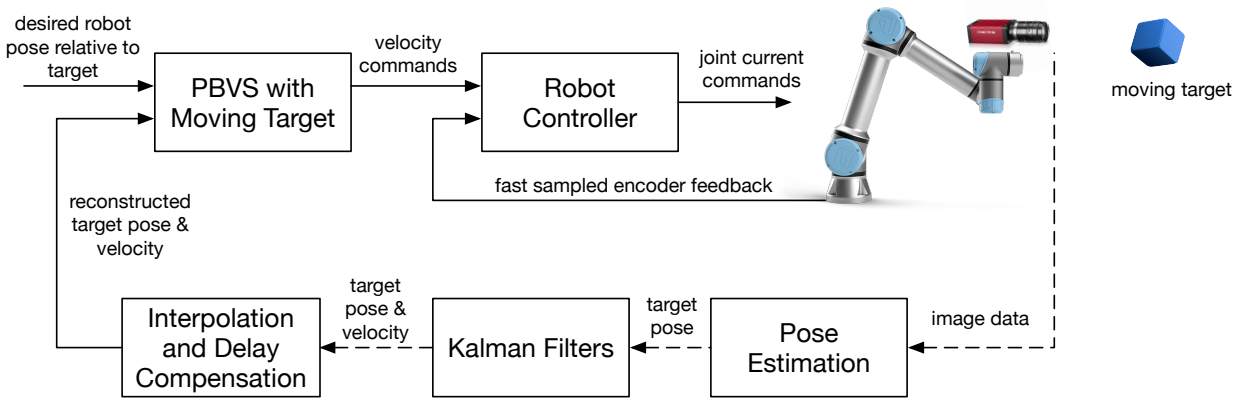


Figure 4.4: The overview of the proposed target following algorithm. The dashed lines represent signals that are updated every T_{ss} seconds while the solid lines represent fast-sampled signals that will update every T_{sf} seconds.

PBVS.

In order to analyze the impact of slow-sampling and time delay, we designed two sets of scenarios where the target all follow the same track as is shown in Figure 4.5. In the first set of scenarios, we fixed the system's sampling speed and the time delays but varied the target moving speed. The steady-state following errors for the first set of scenarios are shown in Figure 4.7. Not surprisingly, we observe that the object following performance downgrades as the target speed increases. When the target moves 12 times faster than the base level, we see that our improved PBVS actually induced more error compared with the baseline, while MMP techniques can still achieve a 80% error reduction rate.

In the second set of scenarios, we fixed the system's sampling speed and the target moving speed but varied the time delays of the vision system. As is shown in Figure 4.8, the following performance will downgrade as the delay time increase when we applied no delay compensation. With Kalman filters and MMP, the approached algorithm could still constrain the following error to a small level when delay increases.

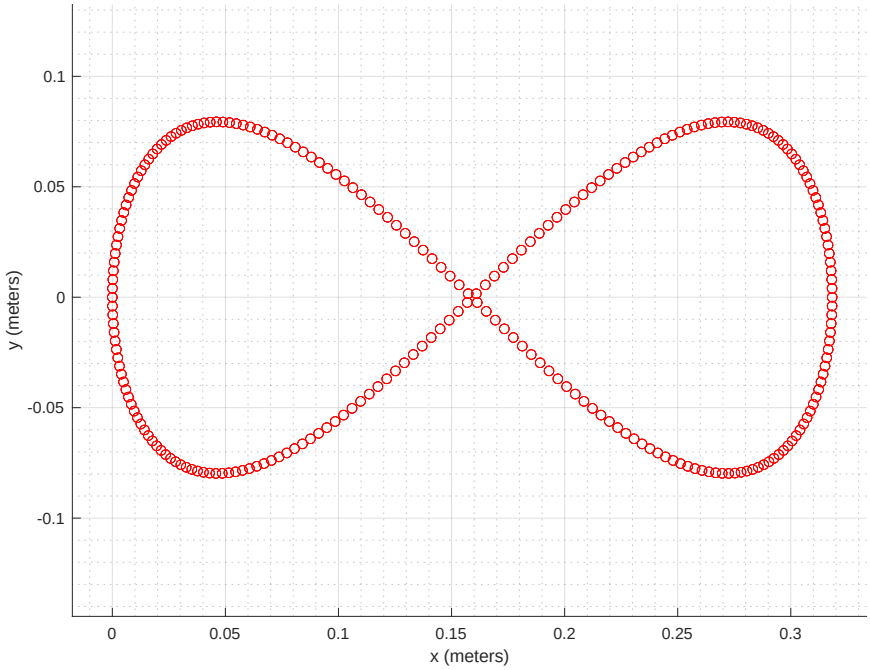


Figure 4.5: The target moving trajectory used in the simulation.

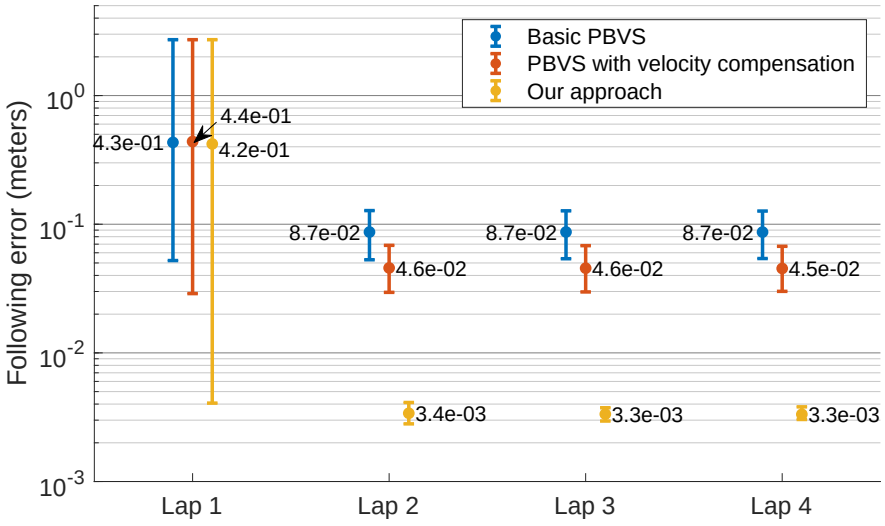


Figure 4.6: Comparison of the following errors in each lap. Each error bar indicates the maximun, minimun and mean errors in a lap.

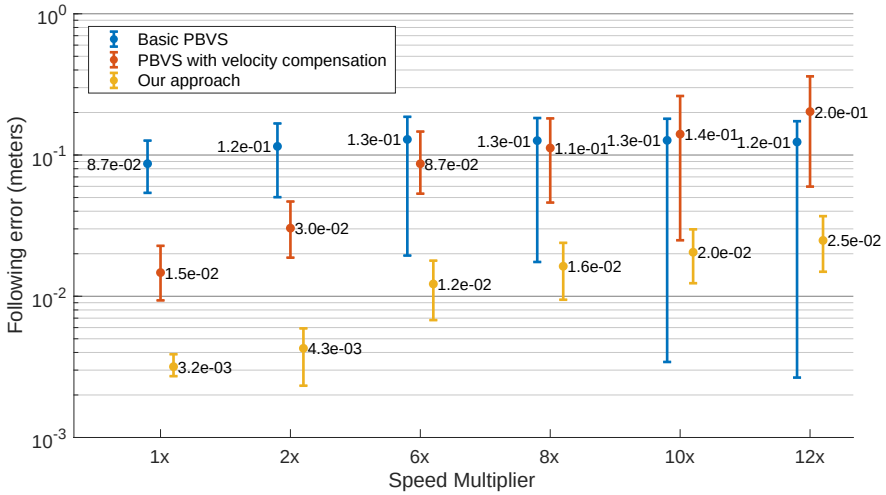


Figure 4.7: Comparison of the following error when the target is moving at different levels of speed. At the base level, target takes 2 seconds to complete one lap. Each error bar indicates the maximun, minimun and the mean values of the errors in the last lap.

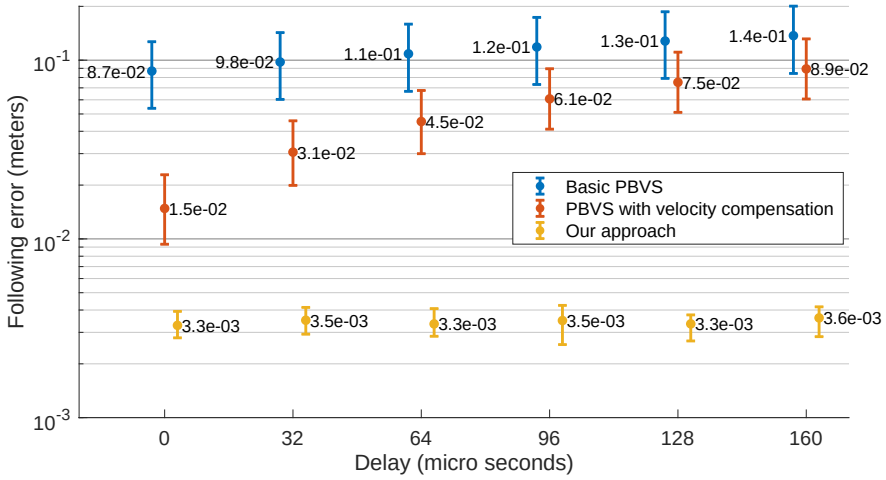
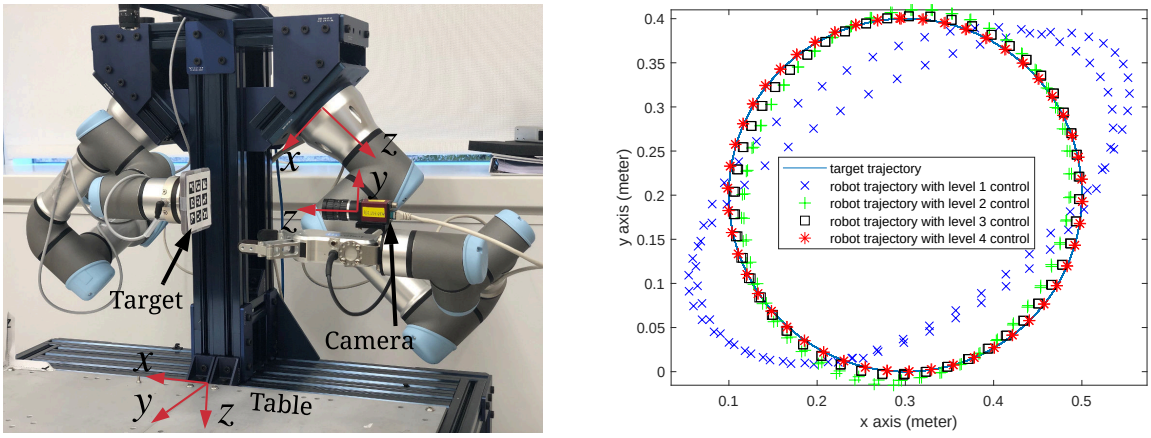


Figure 4.8: Comparison of the following error when the vision system has different levels of time dealys. Each error bar indicates the maximun, minimun and the mean values of the errors in the last lap.



(a) (b)

Figure 4.9: (a)Dual-arm robot used to test the target following algorithm. (b)Moving trajectories of the target and robot in 2D.

4.6 Experiment results

The proposed target following algorithm was tested on a dual-arm robot, as is shown in Figure 4.9a. The left arm in the figure holds a target and the right arm has a camera mounted to the end-effector. The world coordinate system $\{W\}$ is attached to the base of the right robot arm. In the experiment, we moved the target with a 2-D circle trajectory that is parallel to the W-x-y plane. Specifically, the linear x and y velocities of the target is controlled to be sinusoidal with $\pi/2$ phase difference. During the target following experiment, the end-effector's position and velocity of the left robot are assumed to be in the 3D space and are unknown to the right robot, but is used to calculate the following errors at a high sampling speed for analysis purposes. Note that the visual following errors are represented respected to the table coordinate system (see Figure 4.9a), which has a 45 degree orientation difference from the robot arm base.

Sampling speed and measurement delay. The camera used is Mako G192C from Allied Vision Technology which has 60fps at a full resolution of 1600 × 1200 pixels. ArUco

markers [53] are used for estimating the target pose in the camera frame. Due to the heavy computation cost of the marker detecting and pose estimation process, the target pose measurements can only be updated at a maximum rate of about 20Hz. On the other hand, the robot servo loop is running at a high sampling rate of 125Hz. To make the fast and slow sampling rate almost integer multiples, we triggered the camera to acquire images at a fixed rate of 17.85fps, then $L = T_{ss}/T_{sf} = 7$. The measurement delays can be monitored by adding time stamps to the images when captured. In our experiment setup, the measurement delays were about 50ms.

Two scenarios were tested using the proposed target following algorithm. In the first scenario the target moved at 0.5Hz, while in the second scenario, the target moved at a higher frequency of 1Hz. In both scenarios, the CF model was used for linear x and y axes for the Kalman filtering; DWNA model was used for other axes. Then the appropriate polynomial models were chosen from Table 4.1 and MMPs were applied for all axes. Note that if a time-invariant polynomial model is assumed, then the MMP parameters $w'_{k,i}$s can be calculated off-line to save the on-line computation cost. Figure 4.10a shows the target following errors of the linear x, y and z axes in the first scenario. Here, we tested four levels of target following algorithms. The **first** level only uses the basic PBVS algorithm which is designed based on a static target assumption [3]. The **second** level considered a moving target, and the estimated target positions and velocities from the Kalman filters are directly used. The **third** level not only considered a moving target but also compensated the delays using MMP. In order to show the importance of fast sampling rate of measurement, we down-sampled the MMP outputs in the third level to the slow sampling rate $1/T_{ss}$ Hz. Finally, in the **fourth** level, both Kalman filters and MMP are used without down-sampling. Statistic result of the target following error are shown in Figure 4.10b, where the 3σ value plus the absolute mean of the errors are computed for comparison. The same performance improvements can also be observed by comparing the moving trajectories of the target and robot (see Figure 4.9b). One can clearly observe the performance improvement when the target velocity estimation, delay compensation, and interpolation techniques are added to

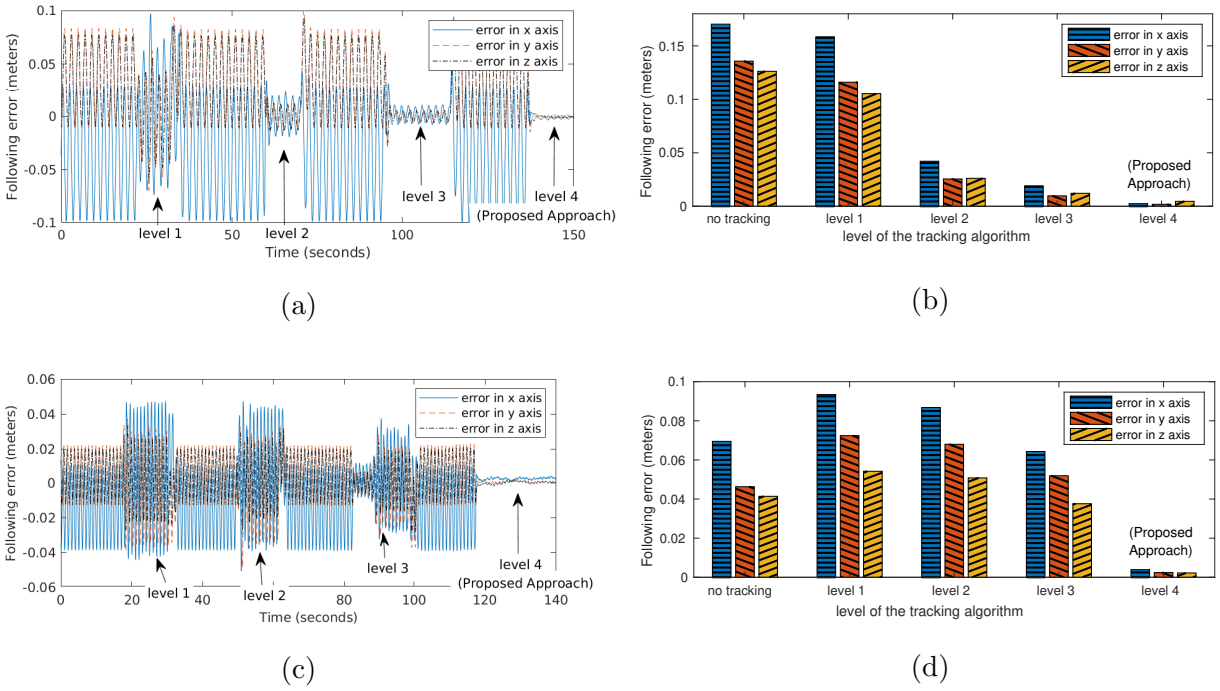


Figure 4.10: (a)Following error when target moves at 0.5Hz. (b)Following error statistics when target moves at 0.5Hz. (c)Following error when target moves at 1Hz. (d)Following error statistics when target moves at 1Hz.

the algorithm.

For the second scenario, the following error of linear x, y and z-axes are shown in Figure 4.10c and Figure 4.10d. The same four levels of target following algorithm are tested. Different from the first scenario, the first three levels actually failed to reduce the following error. This is because when the target moves at a higher frequency, the consequences of measurement delay and slow measurement sampling become more significant. As a result, it requires both delay compensation and interpolation to reduce the following error.

Chapter 5

TARGET TRACKING UNDER DELAYED AND IRREGULAR-SAMPLED SENSING

In this chapter, we present a target tracking system for a robotic air-hockey player, where the state of the puck is optimally estimated by fusing measurements from frame- and event-based vision sensors. In addition to the delay and slow-sampling of vision sensors, we additionally consider the irregular sampling interval of the measurements. The technical challenge of the problem is also amplified by the jumping velocity of the tracked object during the game. An auto-restart Kalman filter is first proposed for compensating sudden jumps in the puck state. Then a memory-enabled auto-restart Kalman filter is derived to additionally accommodate delays and sensing irregularities.

The motivation for this work is to build a robotic air-hockey player that can play with a human player. In a standard air-hockey game, a plastic puck slides on a rectangular air-hockey table; two players compete to crash their pushers (strikers) across the table to throw the puck into the opponent's goal. Constant air flows from densely spaced holes on the tabletop to suspend the puck from the table surface, so that the puck can fly from one side of the table to the other within half a second. As such, fast and accurate target tracking becomes crucial.

Our robotic air-hockey system (Fig. 5.1) consists of (1) a 6-DOF industrial manipulator with a striker mounted at its end-effector to play against a human player, and (2) a vision system to locate the puck. The vision system includes a common frame-based camera and an event-based camera. The frame-based camera comes with a 1600 by 1200 image size and a maximum image acquisition rate of 60 frame per seconds. The event-based camera has only 640 by 480 image size but can acquire data at a much faster rate. The frame-based

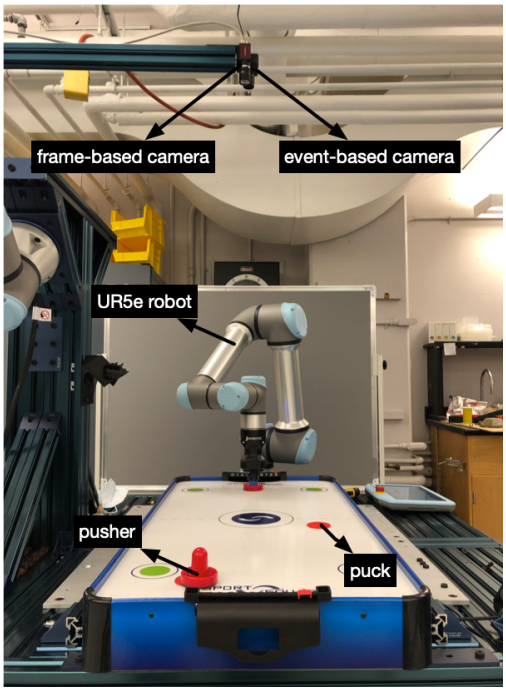


Figure 5.1: The robotic air-hockey system.

camera provides a higher spatial resolution, and thus more accurate information of the target location. However, the slow sampling speed challenges the tracking of high-speed targets. Furthermore, due to motion blurs, performance of localization degrades as the speed of the target increases. On the contrary, the event-based camera independently and asynchronously report pixels that undergo large changes of brightness. The pixel location here, combined the timestamp when the brightness changes at this pixel, constitute an event. Because the timestamp is sampled at a much higher temporal resolution (usually in microseconds), the event-based camera can localize fast-moving targets at a much faster speed. However, such a mechanism cannot detect robustly slow or static targets by design. Fig. 5.2 visualizes the data generated by both sensors when the puck moves at different speeds. By fusing data from both frame-based and event-based cameras, a new and fundamentally more robust target tracking becomes possible.

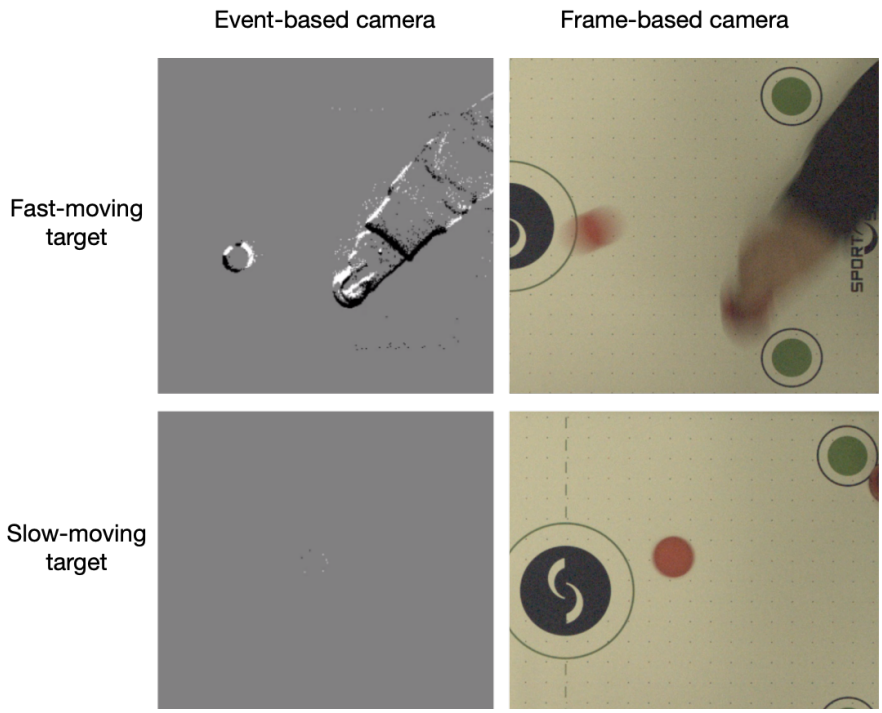


Figure 5.2: Comparison of raw data from the event-based and the frame-based cameras when the target is moving at slow and high speeds. The frames from the event-based camera are obtained by accumulating events in a fixed interval. The images are cropped for better visualization.

5.1 Problem formulation

We model the motion of the puck on the air-hockey table as a state-space dynamic system. The state of the puck consists of its position $\xi_{x/y}(t)$ and velocity $\dot{\xi}_{x/y}(t)$ in 2D:

$$x(t) = \begin{bmatrix} \xi_x(t) & \dot{\xi}_x(t) & \xi_y(t) & \dot{\xi}_y(t) \end{bmatrix}^T \tag{5.1}$$

The rotational dynamics of the puck has limited impact on the sliding trajectory, thus omitted for simplicity. The goal is to obtain the estimated state $p(x(t)|Z^t)$ at time $t = iT$ given all available measurements Z^t from both sensors at time t. As is shown in Fig. 5.3, the measurements are sampled irregularly. Each measurement is delayed by a variable period

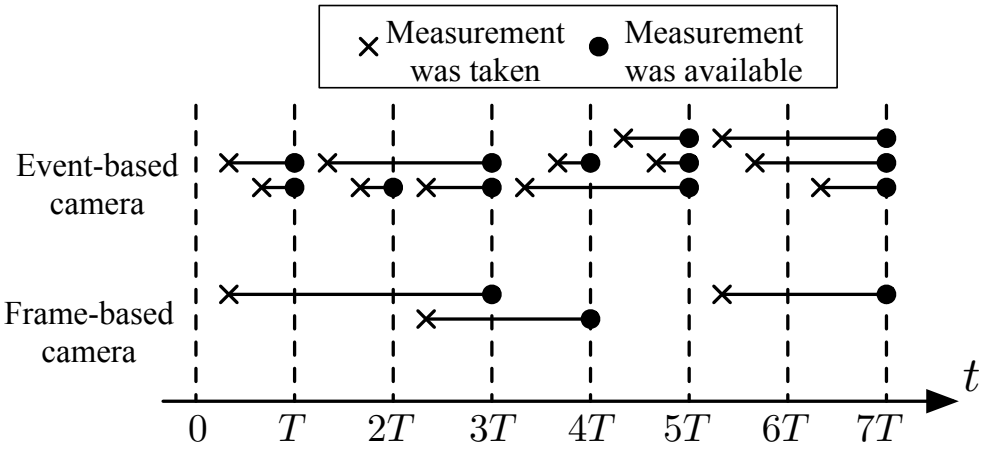


Figure 5.3: Illustration of the delayed and irregularly-sampled measurements.

before it becomes available, with the amount of delays traceable from timestamps of the sensors. Mathematically, the measurements are modeled as

$$z(t_k) = Hx(t_k - \tau_k) + \omega(t_k) \tag{5.2}$$

where $H = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$, t_k is the time when the measurement becomes available, τ_k is the delay duration and $\omega(k)$ is a zero-mean white Gaussian measurement noise with known covariance.

$$E[\omega(t_k)\omega(t_k)^T] = R(t_k) \tag{5.3}$$

5.2 System modeling

As the velocity of the target undergoes only small changes when the puck floats on air inside the air-hockey table, the acceleration of the puck can be modeled as a continuous time zero-mean white noise $\nu_c(t) = \begin{bmatrix} \nu_x(t) & \nu_y(t) \end{bmatrix}^T$, with

$$E(\nu_c(t)) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{5.4}$$

$$E(\nu_c(t)\nu_c(\tau)^T) = \begin{bmatrix} q_x & 0 \\ 0 & q_y \end{bmatrix} \delta(t - \tau) \tag{5.5}$$

where q_x and q_y are the intensities of the noises on x and y axes and $\delta(\cdot)$ is the Dirac delta function. The continuous-time motion dynamics then become

$$\dot{x}(t) = A_c x(t) + D_c \nu_c(t) \tag{5.6}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.7}$$

The discretized state equation after a time interval T is

$$x(t_k + T) = A_d(T) x(t_k) + \nu_d(t_k, T) \tag{5.8}$$

where

$$A_d(T) = \mathrm{e}^{A_c T} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

$$v_d(t_k, T) = \int_0^T \mathrm{e}^{A_c(T-\tau)} D_c \nu_c(t_k + \tau) \mathrm{d}\tau \tag{5.10}$$

From (5.5), (5.9) and (5.10), it can be shown that the covariance of the discrete-time process noise, assuming q_x and q_y to be constant from t to $t + T$, is,

$$Q(t_k, T) = Q(T) = E[v_d(t_k, T) v_d(t_k, T)^T] = \begin{bmatrix} \frac{1}{3}T^3 q_x & \frac{1}{2}T^2 q_x & 0 & 0 \\ \frac{1}{2}T^2 q_x & T q_x & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 q_y & \frac{1}{2}T^2 q_y \\ 0 & 0 & \frac{1}{2}T^2 q_y & T q_y \end{bmatrix} \tag{5.11}$$

Note that q_x and q_y are the designing parameters and should be adjusted based on the interval length T. From (5.11), the changes in velocity over interval T are in the order of $\begin{bmatrix} \sqrt{T q_x} & \sqrt{T q_y} \end{bmatrix}^T$, which can be used as a guideline for adjusting q_x and q_y.

5.3 Filtering under irregular sampling with auto-restarting

In this section, we propose optimal state estimation for the motion tracking by using the primitives of Kalman filters. Different from the standard scenarios where the prediction and correction steps are called at a fixed interval, we consider the case when the time interval T is a variable – the case in our dual-sensor design and common in visual sensing. Given the state-space model (5.8-5.11) and the measurements equation (5.2-5.3) with zero delay (i.e., $\tau_k = 0$), the minimum mean square error estimates of the state can be obtained as follows. In the prediction step, the predicted state $\hat{x}(t_k + T|t_k)$ and state prediction covariance $P(t_k + T|t_k)$ are calculated as

$$\hat{x}(t_k + T|t_k) = A_d(T)\hat{x}(t_k|t_k) \tag{5.12}$$

$$P(t_k + T|t_k) = A_d(T)P(t_k|t_k)A_d(T)^T + Q(T) \tag{5.13}$$

Similarly, the predicted measurement $\hat{z}(t_k + T|t_k)$ and measurement prediction covariance $S(t_k + T)$ are

$$\hat{z}(t_k + T|t_k) = Hx(t_k + T|t_k) \tag{5.14}$$

$$S(t_k + T) = HP(t_k + T|t_k)H^T + R(t_k + T) \tag{5.15}$$

In the correction step, we first compute the Kalman filter gain $K(t_k + T)$

$$W(t_k + T) = P(t_k + T|t_k)H^T S(t_k + T)^{-1} \tag{5.16}$$

Then the updated state estimation $\hat{x}(t_k + T|t_k + T)$ and updated covariance $P(t_k + T|t_k + T)$ can be computed as

$$\hat{x}(t_k + T|t_k + T) = \hat{x}(t_k + T|t_k) + W(t_k + T)(z(t_k + T) - \hat{z}(t_k + T|t_k)) \tag{5.17}$$

$$P(t_k + T|t_k + T) = P(t_k + T|t_k) - W(t_k + T)S(t_k + T)W(t_k + T)^T \tag{5.18}$$

5.3.1 Initialization

We initialize the state and the state covariance using a single measurement

$$z(t_0) = \begin{bmatrix} z_x(t_0) & z_y(t_0) \end{bmatrix}^T, \tag{5.19}$$

with zero as the initial estimate for velocity with associate standard deviation equal to half of known maximum speed v_m [52]. That is,

$$\hat{x}(t_0|t_0) = \begin{bmatrix} z_x(t_0) & 0 & z_y(t_0) & 0 \end{bmatrix}^T \tag{5.20}$$

$$P(t_0|t_0) = \begin{bmatrix} \sigma(t_0)^2 & 0 & 0 & 0 \\ 0 & 0.25v_m^2 & 0 & 0 \\ 0 & 0 & \sigma(t_0)^2 & 0 \\ 0 & 0 & 0 & 0.25v_m^2 \end{bmatrix} \tag{5.21}$$

where $\sigma(t_0)$ is the standard deviation of the measurement noise associated with measurement $z(t_0)$.

5.3.2 Bounce off the table boundary

Edges of the air-hockey table limit the range of motion and thus ξ_x and ξ_y in the state vector. When the puck bounces off the table edges, the velocity jumps. We propose to handle such disturbances by adding an additional correction step triggered by an estimated state off the edge limits. For a single axis $i = x$ or y, the correction step is

$$\xi_i^* = \begin{cases} 2\xi_{i,max} - \xi_i, \text{ if } \xi > \xi_{i,max} \\ 2\xi_{i,min} - \xi_i, \text{ if } \xi < \xi_{i,min} \end{cases} \tag{5.22}$$

$$\dot{\xi}_i^* = -\dot{\xi}_i, \text{ if } \xi > \xi_{i,max} \text{ or } \xi < \xi_{i,min} \tag{5.23}$$

where $\xi_{i,max}$ and $\xi_{i,min}$ are the upper and lower position limits for axis i, ξ_i^* and $\dot{\xi}_i^*$ are the corrected position and velocity. This triggered correction step, as is illustrated in Fig 5.4, models the physics of puck bouncing off the table edge, assuming an elastic collisions model.

5.3.3 Bounce off the pusher

Disturbances also occur when the puck is struck by the pusher. However, a triggered correction is not applicable here since the pusher's location is unknown. Such disturbances, if

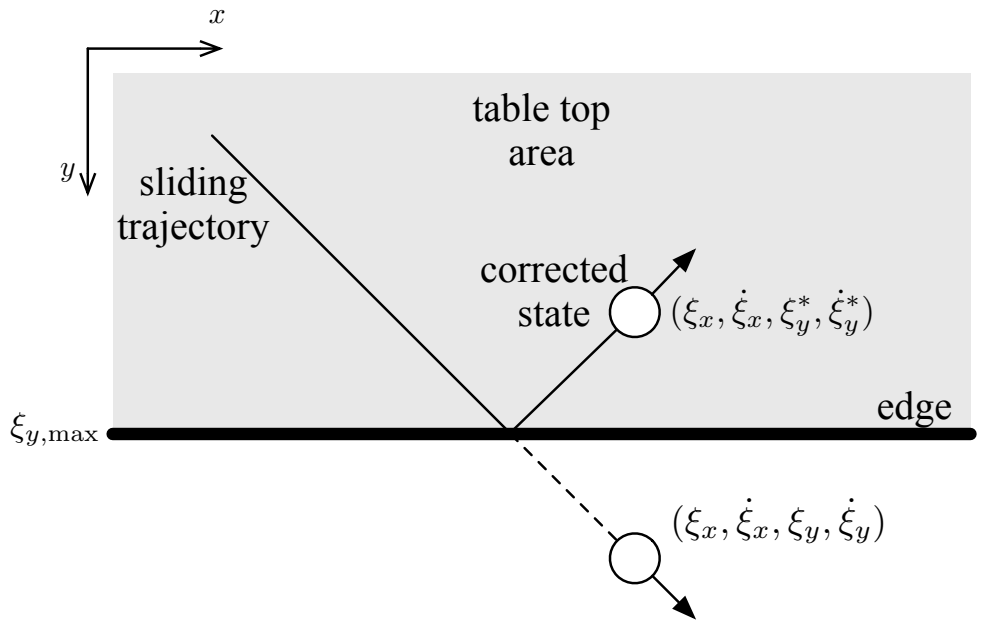


Figure 5.4: Illustration of the correction step triggered by state position violating limits.

not handled, will immediately cause large tracking errors (see the orange line in Fig. 5.5). To let the Kalman filter quickly correct the tracking error induced by the disturbances, one can choose a large process noise covariance $Q(T)$, with the sacrifice of tracking performance at other times (see the green line in Fig. 5.5). In this dissertation, we developed a method to continuously monitor the model mismatch and re-initialize the filter after detecting a mismatch. The proposed algorithm lets the Kalman filter compensate the disturbance much quicker, while maintaining a good tracking performance when there are no disturbances.

The Kalman filter innovation $\tilde{z}(t_k+T) \triangleq z(t_k+T) - \hat{z}(t_k+T)$, under the model assumption, is zero-mean and Gaussian distributed

$$\tilde{z}(t_k+T) \sim \mathcal{N}(\mathbf{0}, S(t_k+T)) \tag{5.24}$$

For each axis i, we test if the following hypothesis is true

$$H_{i0}: \ \tilde{z}_i(t_k+T) \sim \mathcal{N}(0, S_{ii}(t_k+T)) \tag{5.25}$$

The hypothesis H_{i0} is rejected if $\|\tilde{z}_i(t_k+T)\|$ is greater than a threshold λ_i. Then the false alarm probability is

$$\alpha_i \triangleq P(\|\tilde{z}(t_k+T)\| > \lambda_i \mid H_{i0}) \tag{5.26}$$

which gives

$$\lambda_i = \sqrt{S_{ii}(t_k+T)}\Phi^{-1}(1-\frac{\alpha_i}{2}) \tag{5.27}$$

where $\Phi^{-1}(x)$ is the inverse cumulative distribution function under the assumption of standard Gaussian distribution of the noises. In summary, with the desired false alarm probability α_x and α_y, (5.27) gives the threshold λ_x and λ_y. If any of the innovation components $\|\tilde{z}_x(t_k+T)\|$ and $\|\tilde{z}_y(t_k+T)\|$ is greater than the threshold, the filter will be re-initialized using the latest measurement. The false alarm probability is a designing parameter. The larger its value, the sooner the re-initialization will be triggered after the pusher strikes the puck. However, re-initialization will also be likely to be triggered by random measurement noises if α_i is larger. Fig. 5.5 shows the tracking performances of two Kalman filters with different α_i, one with $\alpha_x = \alpha_y = 0.0005$ (shown in green line) and the other with $\alpha_x = \alpha_y = 0.01$ (shown in purple line). The mean square error was based on 500 independent simulations of the same scenario. Here , the puck is sliding with constant velocity but was kicked by a player at $t = 2.5$ seconds. The Kalman filter with a larger α_i can detect the disturbance slightly faster, however, the tracking performance at other times is significantly lower than Kalman filters, due to frequent unnecessary re-initializations.

5.3.4 Variable-step Kalman filter with auto-reinitialization

Combining the above, we have the algorithm for estimating puck state under irregularly-sampled measurements but without delays. The algorithm, called auto-restart Kalman filter (ARKF) for short, is summarized below.

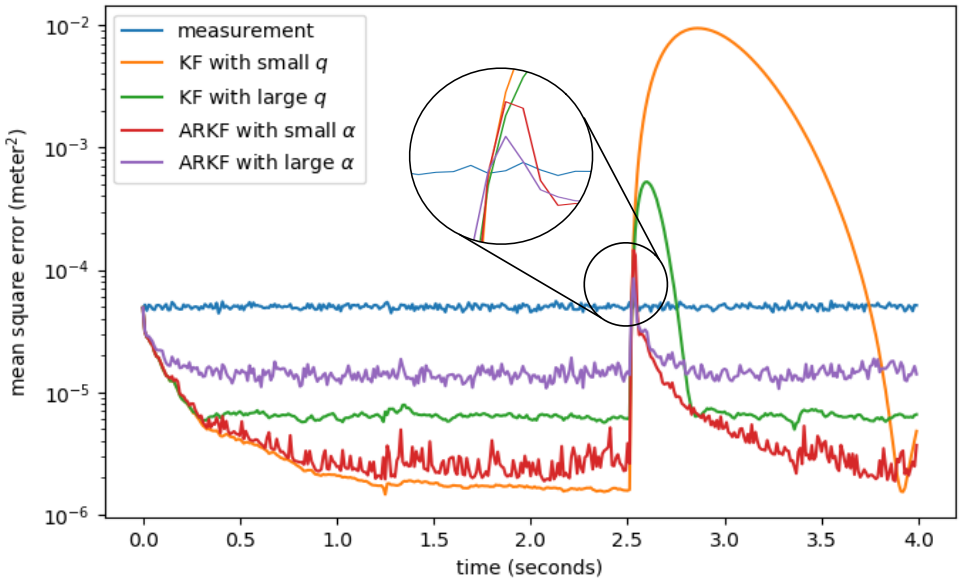


Figure 5.5: Comparison of the tracking performance under unpredictable disturbances for Kalman filters and auto-restart Kalman filters.

Algorithm 2: Auto-restart Kalman Filter (ARKF)

input : $\hat{x}(t_k|t_k)$, $P(t_k|t_k)$, $z(t_k+T)$, and step size T

output: $\hat{x}(t_k+T|t_k+T)$, $P(t_k+T|t_k+T)$

compute $\hat{x}(t_k+T|t_k)$ and $P(t_k+T|t_k)$, using (5.12-5.13);

if $\hat{x}(t_k+T|t_k)$ *off the table boundary* **then**

  correct $\hat{x}(t_k+T|t_k)$, using (5.22-5.23);

end if

compute $\hat{z}(t_k+T|t_k)$ and $S(t_k+T)$, using (5.14-5.15);

compute threshold λ_x and λ_y, using (5.27);

if $\|z(t_k+T)-\hat{z}(t_k+T|t_k)\|$ *pass the threshold* **then**

  compute $\hat{x}(t_k+T|t_k+T)$, $P(t_k+T|t_k+T)$, using (5.20-5.21);

else

  compute $\hat{x}(t_k+T|t_k+T)$, $P(t_k+T|t_k+T)$, using (5.16-5.18);

end if

5.4 Filtering under delayed and irregularly sampled measurements

In this section, we discuss the important accommodation of measurement delays that are common in visual sensing. Consider the continuous-time model (5.6) with delayed and irregularly sampled measurements from both sensors. Although the measurements are delayed and not regularly updated, we still want to obtain an estimate of the state every T seconds. We propose a memory-enabled Kalman filter that consists of an update step and a prediction step. The filter stores a limited number of past filter states in its memory. The update step is called every T seconds to update the memory based on all measurements arrived in the past T seconds. The prediction step is called to obtain an estimated state in current or future time, based on current memory data.

5.4.1 Update step

Fig. 5.6 showed the update circle for our proposed memory-enabled Kalman filter. Here, there are 4 filter states in memory and three measurements arrived in the update circle. The combined Z^* include four measurements, where $\tilde{z}(t_1^*)$ are fused from $z(t_3)$ and $z(t_1^*)$. A filter state $s(t_k)$ in the memory includes the measurement $z(t_k)$, the state estimation $\hat{x}(t_k|t_k)$ and the covariance $P(t_k|t_k)$ at a past time t_k. The memory will store all filter states in a fixed time window $[t_n - t_w, t_n]$, where t_n is the current time and t_w is the length of the time window. The length of the time window should be chosen to be greater than the maximum known sensor delay time. Assuming there are l numbers of filter states in the memory, compactly noted as

$$\{s(t_i)\}_{i=1}^{l},\ t_n - t_w < t_1 < t_2 < \cdots < t_l < t_n \tag{5.28}$$

where

$$s(t_i) \triangleq \{z(t_i), \hat{x}(t_i|t_i), P(t_i|t_i)\} \tag{5.29}$$

The new measurements since the last update step, assuming there are p of them, are $\{z(t_i^*)\}_{i=1}^{p}$. Here, t_i^* is the timestamp when the original measurement data is taken, rather than the timestamp when the processed measurement data is received.

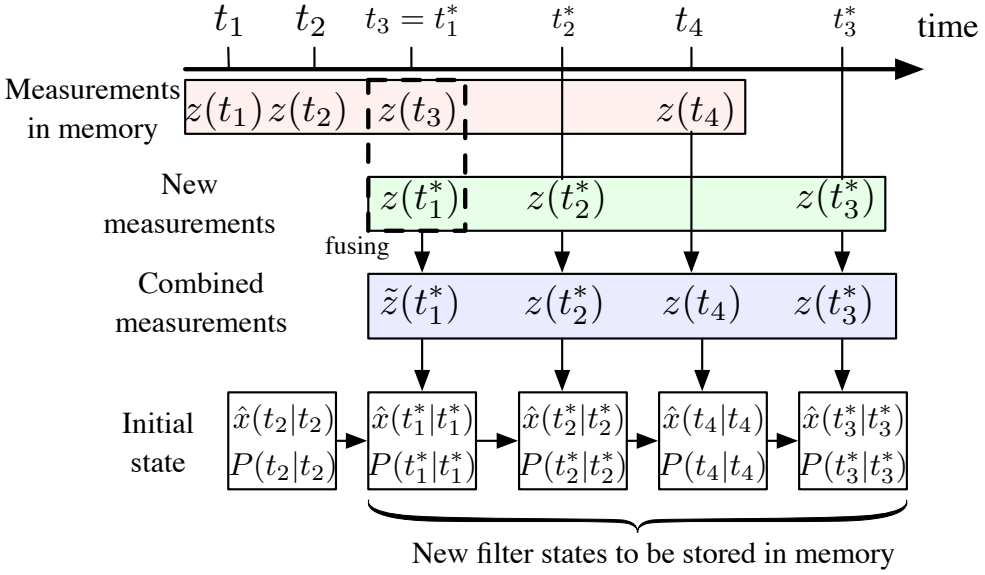


Figure 5.6: Illustration of the memory-enabled Kalman filter update circle.

In the update step, we first combine new measurements with old measurements stored in the memory, and sort them based on timestamp to get a new measurement list

$$\{z(t_1), \cdots, z(t_n), z(t_1^*), \cdots z(t_i), z(t_j^*), \cdots z(t_p^*)\} \tag{5.30}$$

Then we crop the combined measurements list to get Z^*, whose first element is the first new measurement $z(t_1^*)$

$$Z^* = \{z(t_1^*), \cdots z(t_i), z(t_j^*), \cdots z(t_p^*)\} \tag{5.31}$$

Note that it is possible that two measurements have the same timestamp, each from a different sensor. In this case, measurement fusion must be carry out to fuse two measurements into one. Denoting two measurements as z^1 and z^2 with the corresponding measurement error covariances R^1 and R^2, we have the fused measurement and covariance [54]

$$z = z^1 + R^1(R^1 + R^2)^{-1}(z^2 - z^1) \tag{5.32}$$

$$R = ((R^1)^{-1} + (R^2)^{-1})^{-1} \tag{5.33}$$

Now, we have obtained a list of measurement ordered by time. Then Kalman filters can be applied to process these measurements one by one, creating new filter states to be stored in memory. The initial state estimation and covariance would be $\hat{x}(t_n|t_n)$ and $P(t_n|t_n)$, where t_n is the first timestamp before t_1^* in (5.30). Finally, with the updated current time t_n, we discard filter states older than time $t_n - t_w$ to save memory space.

5.4.2 Prediction step

The predicted state at time t_p based on the current filter memory can be obtained from the latest filter state $s(t_l) = \{z(t_l), \hat{x}(t_l|t_l), P(t_l|t_l)\}$.

$$\hat{x}(t_p|t_l) = A_d(t_p - t_l)\hat{x}(t_l|t_l) \tag{5.34}$$

$$P(t_p|t_l) = A_d(t_p - t_l)P(t_l|t_l)A_d(t_p - t_l)^T + Q(t_p - t_l) \tag{5.35}$$

5.4.3 Memory-enabled Variable-step Kalman Filter with Auto-reinitialization

Combining the memory-based structure with the auto-restart Kalman filter, we finally get the filtering algorithm that can estimate puck state under delayed and irregularly-sampled measurements from both sensors. The algorithm, referred below as the memory-enabled auto-restart Kalman filter (M-ARKF), is summarized in Algorithm 2.

5.5 Numerical simulation

We test the tracking performance of M-ARKF in a simulated air-hockey environment. In the tested scenario, the puck slides at a constant velocity of $(\dot{\xi}_x, \dot{\xi}_y) = (0.6, 0.4)$ meters per seconds starting at $t = 0$. After the puck was bounced back from table edge, it got struck by the player at $t = 1.25$ seconds. The puck's moving trajectory is visualized in Fig. 5.7.

Fig. 5.8 shows the position measurements from both sensors, which are generated by adding Gaussian noise to the true position. In this simulation, there are 134 measurements received from the event-based camera and 58 measurements received from the frame-based camera. Each measurement is delayed by a random time sampled from a fixed range (see

Algorithm 3: Memory-enabled Auto-restart Kalman Filter (M-ARKF)

input : $\{z(t_i^*)\}_{i=1}^{p}$, t_p

output : $\hat{x}(t_p|t_l)$, $P(t_p|t_l)$

memory: $\{z(t_i)\}_{i=1}^{l}$

combine $\{z(t_i^*)\}_{i=1}^{p}$ with $\{z(t_i)\}_{i=1}^{l}$ and reorder by timestamp to obtain Z;

Extract Z^* (5.31) from Z and find the timestamp t_n right before t_1^*;

Fuse measurements in Z^* that has the same timestamp, using (5.32-5.33);

$\hat{x}(t_0|t_0) \leftarrow \hat{x}(t_n|t_n)$, $P(t_0|t_0) \leftarrow P(t_n|t_n)$, $t_0 \leftarrow t_n$;

foreach $z(t) \in Z^*$ **do**

  $\hat{x}(t|t), P(t|t) \leftarrow ARKF(\hat{x}(t_0|t_0), P(t_0|t_0), z(t), t - t_0)$;

  add $\{\hat{x}(t|t), P(t|t), z(t)\}$ to filter memory ;

  $\hat{x}(t_0|t_0) \leftarrow \hat{x}(t|t)$, $P(t_0|t_0) \leftarrow P(t|t)$, $t_0 \leftarrow t$;

end foreach

discard filter state with timestamp old than $t_n - t_w$ calculate $\hat{x}(t_p|t_l)$, $P(t_p|t_l)$, using (5.34-5.35)

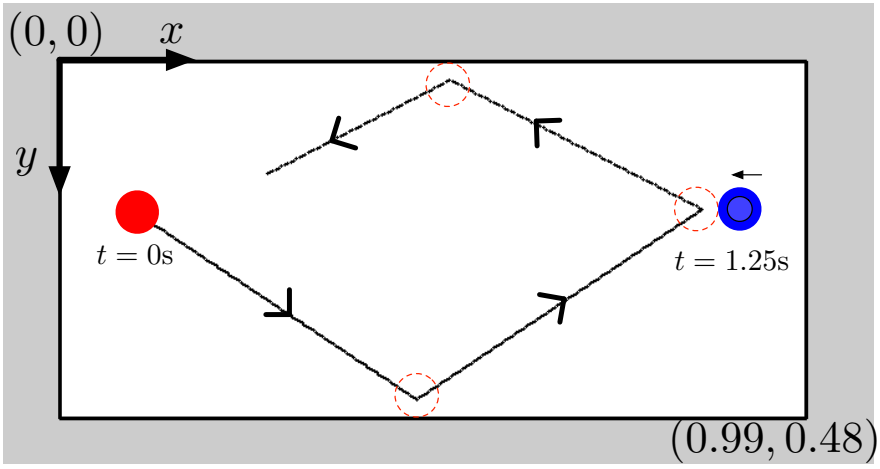


Figure 5.7: The moving trajectory of puck in the simulation. The puck was struck by pusher (visualized in blue) at $t = 1.25$ seconds.

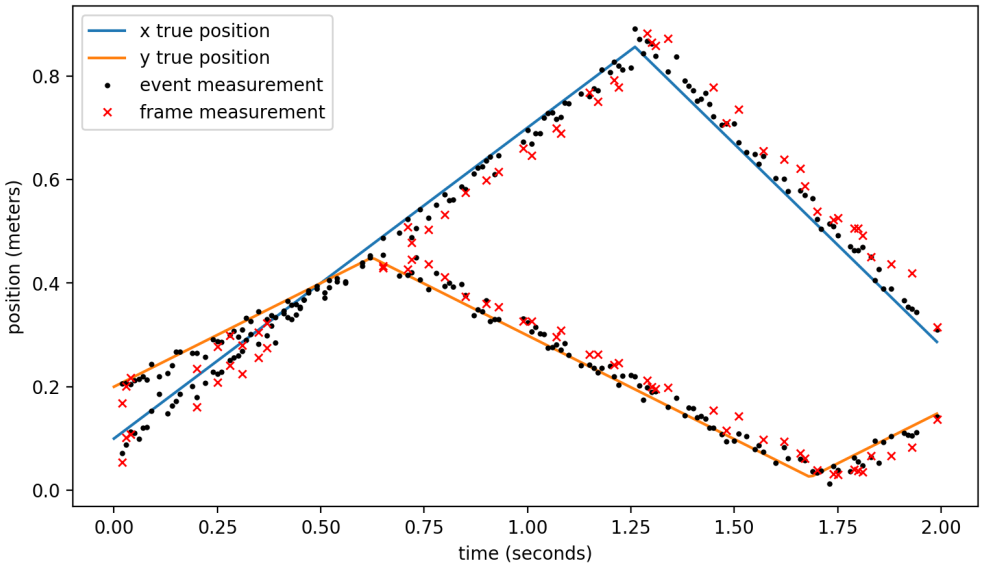


Figure 5.8: The position measurements from event-based camera and frame-based camera.

Fig. 5.9). The measurements from frame-based cameras are simulated to have more delays, mimicking the fact that the original data from fame-based cameras are bigger than those from event-based cameras and will take longer to process.

The M-ARKF was called every $T = 0.1$ seconds to process the measurements data and to predict the state at the update time. Fig. 5.10 shows the averaged position and velocity prediction error squared over 500 independent tests. The measurement noise level is also shown in the figure. The results show that M-ARKF is able to track the puck state and maintains small position and velocity errors when there is no disturbance. There are error spikes at $t = 1.25$ seconds, due to the unpredictable disturbance caused by the player striking the puck. Compared with Fig. 5.5 where the same ARKF was used to correct disturbances, the error spike is larger in M-ARKF results. This is expected because the measurements in M-ARKF tests comes with time delays, which will also delay the re-initialization process for disturbance correction.

Fig. 5.11 shows the comparison of tracking error between M-ARKF and the KF. Here, the

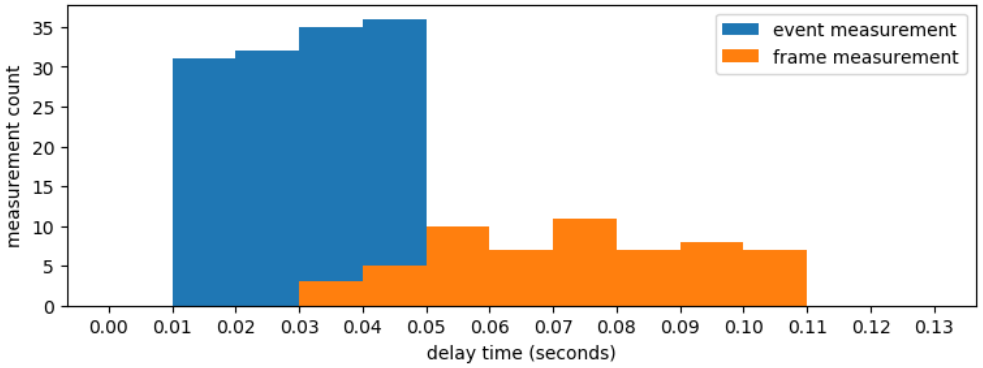


Figure 5.9: The delay time distribution for event-based camera and frame-based camera measurements.

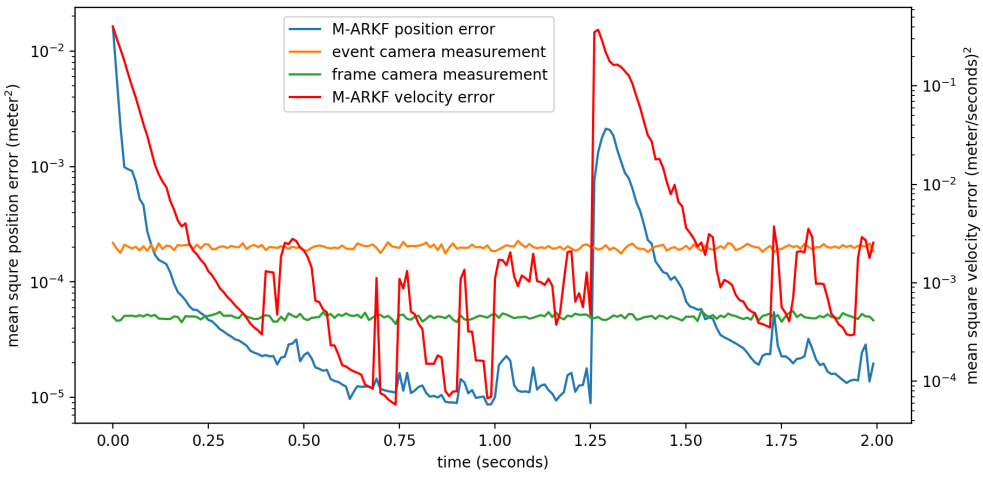


Figure 5.10: Tracking performance of M-ARKF. The mean square error was based on 500 independent runs of the simulation.

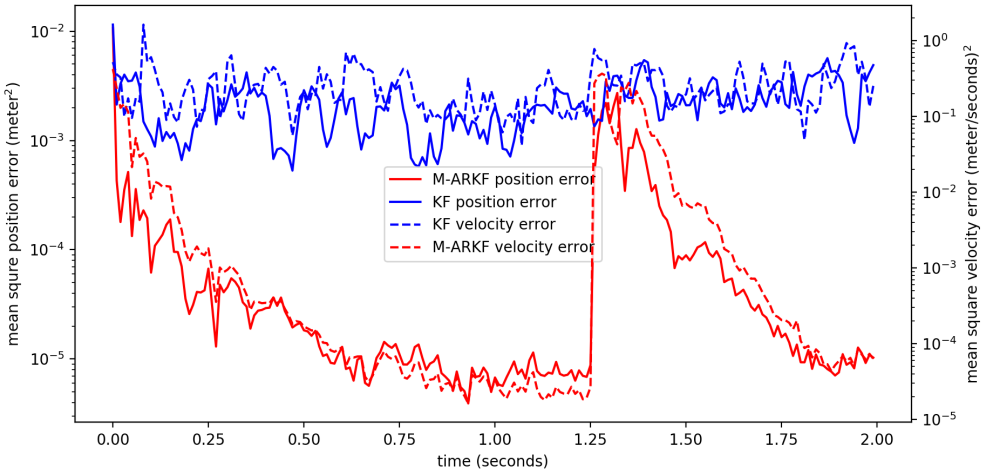


Figure 5.11: Comparison of tracking error between M-ARKF and KF.

KF have applied variable time steps during prediction and correction. However, due to the variable time-delays of both measurements, the Kalman filter have shown very poor tracking accuracy. The proposed memory-enabled auto-restart Kalman filter, however, achieved significantly better state estimation error converging. It also inherited the ability to handle the disturbance at $t = 1.25$ seconds from the design in auto-restart Kalman filter.

Chapter 6

CONCLUSIONS AND FUTURE WORKS

In this dissertation, we developed a basic model-based recovery (MR) algorithm and its two extensions. The basic MR utilized the autoregressive model of fast signals and re-parameterized it into a sparse structure. One extension of MR has an infinite-input response filter structure to reduce the measurement noise. The other extension of MR applies to collaborative sensors where two sensors sample the same signal with different sampling intervals. All of the above MR algorithms have been verified in the disturbance rejection control system. The problem of beyond-Nyquist disturbance rejection is addressed with the proposed multirate model-based disturbance observer (MR-FMDOB) design. Under this direction, the future work is to study the autoregressive model identification algorithm, such that MR can be adaptive to model variations and uncertainties.

We showed in Chapter 4 that the model-based prediction (MMP) together with Kalman filter can obtain target position and velocity estimates with slow-sampling and delay compensated. Here, MMP uses an autoregressive model of the target but the Kalman filter uses a state-space model. As is shown in Chapter 5, the Kalman filter can actually be extended to do the same task as MR. The future work is to apply a dual-rate Kalman filter to the robotic target following algorithm and compare the performance with our current approach. We also discussed a general method for robots to follow the 3D target movement with known target velocity—an extension to basic position-based visual servoing control. There are interests and values to study the corresponding extension to image-based visual servoing. The problem is estimated to be much more complicated due to the nonlinear relationship in image projection.

This dissertation also presented M-ARKF for tracking puck state in a robotic air-hockey

system. The algorithm was tested successful in simulation and experimentation to reduce the negative effect of visual sensing dynamics. This method is designed to be sufficiently general, so it can be applied to other tracking systems. M-ARKF uses the state space mode of the target to compensate the sensing dynamics. Compared with MR where autoregressive model is used, M-ARKF is more flexible for nonlinear or even statistical system models, using the same techniques in the extended Kalman filter and particle filter. However, M-ARKF has more computation cost than the MR algorithm.

BIBLIOGRAPHY

[1] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 67–79, 2005.

[2] X. Chen and M. Tomizuka, "A minimum parameter adaptive approach for rejecting multiple narrow-band disturbances with application to hard disk drives," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 2, pp. 408–415, 2012.

[3] F. Chaumette and S. Hutchinson, "Visual servo control. I. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[4] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[5] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 199–228, World Scientific, 1993.

[6] A. Castaño and S. Hutchinson, "Visual compliance: Task-directed visual servo control," *IEEE transactions on Robotics and Automation*, vol. 10, no. 3, pp. 334–342, 1994.

[7] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[8] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 814–825, 2005.

[9] G. Morel, T. Liebezeit, J. Szewczyk, S. Boudet, and J. Pot, "Explicit incorporation of 2d constraints in vision based control of robot manipulators," in *Experimental Robotics VI*, pp. 99–108, Springer, 2000.

[10] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3307–3314, 2018.

[11] F. Sadeghi, "Divis: Domain invariant visual servoing for collision-free goal reaching," in *Robotics: Science and Systems (RSS)*, 2019.

[12] M. Zhang, X. Liu, D. Xu, Z. Cao, and J. Yu, "Vision-based target-following guider for mobile robot," *IEEE Transactions on Industrial Electronics*, 2019.

[13] H. Firouzi and H. Najjaran, "Real-time monocular vision-based object tracking with object distance and motion estimation," in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 987–992, IEEE, 2010.

[14] R. C. Luo, S.-C. Chou, X.-Y. Yang, and N. Peng, "Hybrid eye-to-hand and eye-in-hand visual servo system for parallel robot conveyor object tracking and fetching," in *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2558–2563, IEEE, 2014.

[15] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[16] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "A position based visual tracking system for a 7 dof robot manipulator using a kinect camera," in *The 2012 international joint conference on neural networks (IJCNN)*, pp. 1–7, IEEE, 2012.

[17] A. Mohebbi, M. Keshmiri, and W. F. Xie, "An eye-in-hand stereo visual servoing for tracking and catching moving objects," in *Proceedings of the 33rd Chinese Control Conference*, pp. 8570–8577, IEEE, 2014.

[18] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, pp. 181–187, IEEE, 1995.

[19] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, "1 ms column parallel vision system and its application of high speed target tracking," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 650–655, IEEE, 2000.

[20] P. Chalimbaud and F. Berry, "Embedded active vision system based on an fpga architecture," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 26–26, 2007.

[21] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 2426–2429, IEEE, 2010.

[22] S. Huang, N. Bergström, Y. Yamakawa, T. Senoo, and M. Ishikawa, "High-speed visual feedback for realizing high-performance robotic manipulation," in *Selected Papers from the 31st International Congress on High-Speed Imaging and Photonics*, vol. 10328, p. 103280V, International Society for Optics and Photonics, 2017.

[23] Y. Yamakawa, Y. Matsui, and M. Ishikawa, "Development of a real-time human-robot collaborative system based on 1 khz visual feedback control and its application to a peg-in-hole task," *Sensors*, vol. 21, no. 2, p. 663, 2021.

[24] S. Namiki, K. Yokoyama, S. Yachida, T. Shibata, H. Miyano, and M. Ishikawa, "Online object recognition using cnn-based algorithm on high-speed camera imaging: Framework for fast and robust high-speed camera object recognition based on population data cleansing and data ensemble," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2025–2032, IEEE, 2021.

[25] I. A. Ivan, M. Ardeleanu, and G. J. Laurent, "High dynamics and precision optical measurement using a position sensitive detector (psd) in reflection-mode: application to 2d object tracking over a smart surface," *Sensors*, vol. 12, no. 12, pp. 16771–16784, 2012.

[26] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[27] A. J. Jerri, "The shannon sampling theorem—its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.

[28] M. Unser, "Sampling-50 years after shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.

[29] Y. Yamamoto, M. Nagahara, and P. P. Khargonekar, "Signal reconstruction via h-infinity sampled-data control theory—beyond the shannon paradigm," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 613–625, 2012.

[30] J. Tani, S. Mishra, and J. T. Wen, "Motion blur-based state estimation," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 1012–1019, May 2016.

[31] C. Wang, C.-Y. Lin, and M. Tomizuka, "Visual servoing considering sensing dynamics and robot dynamics," *IFAC Proceedings Volumes*, vol. 46, no. 5, pp. 45–52, 2013. 6th IFAC Symposium on Mechatronic Systems.

[32] C.-Y. Lin, C. Wang, and M. Tomizuka, "Pose estimation in industrial machine vision systems under sensing dynamics: A statistical learning approach," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4436–4442, IEEE, 2014.

[33] C. Wang, C.-Y. Lin, and M. Tomizuka, "Design of kinematic controller for real-time vision guided robot manipulators," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4141–4146, IEEE, 2014.

[34] X. Chen and H. Xiao, "Multirate forward-model disturbance observer for feedback regulation beyond nyquist frequency," *Systems & Control Letters*, vol. 94, pp. 181–188, 2016.

[35] H. Xiao, T. Jiang, and X. Chen, "Rejecting fast narrow-band disturbances with slow sensor feedback for quality beam steering in selective laser sintering," *Mechatronics*, vol. 56, pp. 166–174, 2018.

[36] H. Xiao, Y. Bar-Shalom, and X. Chen, "A collaborative sensing and model-based real-time recovery of fast data flows from sparse measurements," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6806–6814, 2019.

[37] H. Xiao and X. Chen, "Following fast-dynamic targets with only slow and delayed visual feedback: A kalman filter and model-based prediction approach," in *Dynamic Systems and Control Conference*, 10 2019.

[38] A. Feuer and G. Goodwin, "Generalized sample hold functions-frequency domain analysis of robustness, sensitivity, and intersample difficulties," *IEEE Trans. Autom. Control*, vol. 39, pp. 1042–1047, May 1994. 00107.

[39] D. Wang and X. Chen, "A spectral analysis of feedback regulation near and beyond nyquist frequency," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 916–926, 2018.

[40] D. Youla, J. J. Bongiorno, and H. Jabr, "Modern wiener–hopf design of optimal controllers part i: The single-input-output case," vol. 21, no. 1, pp. 3–13, 1976.

[41] V. Kucera, "Stability of discrete linear feedback systems," in *Proc. 6th IFAC World Congress, paper 44.1*, vol. 1, 1975.

[42] H. Fujioka and S. Hara, "Output regulation for sampled-data feedback systems: Internal model principle and H_∞ servo controller synthesis," *Proc. IEEE Conf. Decision and Control*, pp. 4867–4872, 2006.

[43] C. E. Garcia and M. Morari, "Internal model control. A unifying review and some new results," *Ind. & Eng. Chemistry Process Design and Develop.*, vol. 21, pp. 308–323, Apr. 1982.

[44] B. D. Anderson, "From youla kucera to identification, adaptive and nonlinear control," *Automatica*, vol. 34, no. 12, pp. 1485 – 1506, 1998.

[45] X. Chen and M. Tomizuka, "A minimum parameter adaptive approach for rejecting multiple narrow-band disturbances with application to hard disk drives," vol. 20, pp. 408–415, March 2012.

[46] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50. Siam, 1997.

[47] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate O(n) solution to the PnP problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[48] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the PnP problem: A fast, general and optimal solution," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2344–2351, 2013.

[49] B. Barrois and C. Wöhler, "3d pose estimation of vehicles using stereo camera," *Encyclopedia of Sustainability Science and Technology*, pp. 10589–10612, 2012.

[50] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.

[51] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[52] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[53] S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, pp. 481–491, 2016.

[54] L. Honglei and S. Shuli, "State estimation for a class of non-uniform sampling systems," in *2015 34th Chinese Control Conference (CCC)*, pp. 2024–2027, IEEE, 2015.

[55] P. A. Regalia, S. K. Mitra, and P. Vaidyanathan, "The digital all-pass filter: A versatile signal processing building block," *Proceedings of the IEEE*, vol. 76, no. 1, pp. 19–37, 1988.

[56] P. A. Regalia, "An improved lattice-based adaptive iir notch filter," *IEEE transactions on signal processing*, vol. 39, no. 9, pp. 2124–2128, 1991.

[57] C. E. Garcia and M. Morari, "Internal model control. a unifying review and some new results," *Industrial and Engineering Chemistry Process Design and Development*, vol. 21, no. 2, pp. 308–323, 1982.

Appendix A

A.1 Proof of Theorem 3.1

Proof. By definition, $d[nL] = d_c(nT_s) = d_L[n]$. Hence (3.9) holds. To establish (3.11), construct

$$\begin{aligned} &F_k(z^{-1})A(z^{-1}) + z^{-k}W_k(z^{-L}) = 1, \\ &W_k(z^{-L}) \triangleq w_{k,0} + w_{k,1}z^{-L} + w_{k,2}z^{-2L} + \cdots + w_{k,n_{w_k}}z^{-n_{w_k}L}, \\ &F_k(z^{-1}) \triangleq 1 + f_{k,1}z^{-1} + f_{k,2}z^{-2} + \cdots + f_{k,n_{f_k}}z^{-n_{f_k}}, \end{aligned} \tag{A.1}$$

where $w_{k,n_{w_k}} \neq 0$, $f_{k,n_{f_k}} \neq 0$, and $W_k(z^{-L})$ is obtained by replacing each z^{-1} in $W_k(z^{-1})$ with z^{-L}. As $A(z^{-1})d[n] = 0$ at steady state, it must be that $F_k(z^{-1})A(z^{-1})d[n] \to 0$, which gives, after substituting in (A.1),

$$(1 - z^{-k}W_k(z^{-L}))d[n] \to 0. \tag{A.2}$$

For $n = \tilde{n}L + k$ ($k \in \mathbb{Z}^+,\ k < L$), this implies that, at steady state,

$$\begin{aligned} d[\tilde{n}L + k] &= z^{-k}W_k(z^{-L})d[\tilde{n}L + k] \\ &= w_{k,0}d[\tilde{n}L] + w_{k,1}d[(\tilde{n} - 1)L] + w_{k,2}d[(\tilde{n} - 2)L] + \\ &\qquad \cdots + w_{k,n_{w_k}}d[(\tilde{n} - n_{w_k})L]. \end{aligned} \tag{A.3}$$

But by definition $d[(\tilde{n} - i)L] = d_L[\tilde{n} - i]$. Hence, with a change of notations, the result simplifies to the asserted (3.11).

Consider solving (A.1), which is a special constrained Diophantine equation. Matching the coefficients of z^{-i}'s ($i = 1, 2, \ldots, m + n_f$), one can obtain $m + n_{f_k}$ linear equations with the $n_{f_k} + n_{w_k} + 1$ parameters of $F_k(z^{-1})$ and $W_k(z^{-L})$ as the unknowns. A solution thus

exists if

$$n_{f_k} + n_{w_k} + 1 \geq m + n_{f_k}. \tag{A.4}$$

Additionally in (A.1), the highest order of z^{-1} must be the same in $F_k(z^{-1})A(z^{-1})$ and $z^{-k}W_k(z^{-1})$, namely,

$$m + n_{f_k} = k + Ln_{w_k}. \tag{A.5}$$

Hence the minimum-order solution is achieved with

$$n^*_{w_k} = m - 1,\ n^*_{f_k} = L(m-1) - m + k. \tag{A.6}$$

Under (A.6), the coefficients of z^{-l}'s, $l \in \{1, 2, \ldots, L(m-1)+k\}$, in $A(z^{-1})F_k(z^{-1}) + z^{-k}W(z^{-L})$ are given by

$$w_{k,p} + \sum_{i,j=0,1,\ldots}^{i+j=pL+k} a_i f_{k,j} : \text{for } z^{-pL-k},\ p = 0, 1, \ldots, m-1$$
$$\sum_{i,j=0,1,\ldots}^{i+j=l} a_i f_{k,j} : \text{for } z^{-l},\ l \neq pL + k\ \forall p \in \{0, 1, \ldots, m-1\},$$

where $f_{k,0} = 1$ and $a_0 = 1$. All the above coefficients must be zero for (A.1) to hold. Confining so yields, after some matrix algebra, yields Eq. (3.12). □

A.2 Proof of Theorem 3.2

Proof. For each ω_i, Eq. (3.2) has the solution

$$Q(\mathrm{e}^{j\omega_i}) = \frac{1}{P_d(\mathrm{e}^{j\omega_i})} = \frac{\overline{P_d(\mathrm{e}^{j\omega_i})}}{|P_d(\mathrm{e}^{j\omega_i})|^2}, \tag{A.7}$$

i.e.

$$\begin{cases} \Re Q(\mathrm{e}^{j\omega_i}) = \frac{\Re P_d(\mathrm{e}^{j\omega_i})}{|P_d(\mathrm{e}^{j\omega_i})|^2} \\ \Im Q(\mathrm{e}^{j\omega_i}) = -\frac{\Im P_d(\mathrm{e}^{j\omega_i})}{|P_d(\mathrm{e}^{j\omega_i})|^2} \end{cases}, i = 1, 2, \ldots, m. \tag{A.8}$$

Define first

$$Q^*(z) = q_0 + q_1 z^{-1} + q_p z^{-p} \tag{A.9}$$

such that $P_d(e^{j\omega_i})Q^*(e^{j\omega_i}) = 1$, then by Eq. (A.8), we must have, for $i = 1, 2, \ldots, m$,

$$\begin{cases} q_0 + q_1 \cos\omega_i + \cdots + q_p \cos p\omega_i = \frac{\Re P_d(e^{j\omega_i})}{|P_d(e^{j\omega_i})|^2}, \\ q_1 \sin\omega_i + \cdots + q_p \sin p\omega_i = -\frac{\Im P_d(e^{j\omega_i})}{|P_d(e^{j\omega_i})|^2}. \end{cases} \tag{A.10}$$

There are m such equation sets, or $2m$ linear equations. Since $\omega_i \in (0, \pi)$, and $\omega_i \neq \omega_j$ if $\forall i \neq j$, those linear equations are independent from each other. Then we have $2m$ linearly independent equations and $p + 1 = 2m$ unknowns, and q_i's can be uniquely solved from Eq. (3.22).

The first element in the Q filter in Eq. (3.21), or $Q_0(z)$ in Eq. (3.23), is a multi-band bandpass filter that has m narrow passbands centered at ω_i. It is produced by $1 - Q_1(z)$, where $Q_1(z)$ is constructed by m cascaded lattice-based band-stop filters [55, 56] whose bandwidths are related to $k_{2,i}$'s defined by Eq. (3.25). One can show that $Q_0(e^{j\omega_i}) = 1$ at each center frequency ω_i. Combining Eq. (3.23) and Eq. (A.9) then results in the proposed structure of $Q(z)$ in Eq. (3.2).

Because $Q_0(e^{j\omega_i}) = 1$ and $P_d(e^{j\omega_i})Q^*(e^{j\omega_i}) = 1$, the disturbance rejection requirement (Eq. (3.2)) is satisfied. In addition, when $\omega \neq \omega_i$, $|Q_0(e^{j\omega})|$ can be made arbitrarily small by reducing the bandwidth $B_{w,i}$. Thus $|1 - P_d(e^{j\omega})Q(e^{j\omega})|$ can be controlled to be approximately 1 if $\omega \neq \omega_i$, avoiding large noise amplification. □

A.3 Proof of Theorem 3.3

Proof. In order to establish and validate Eq. (3.28), we construct a polynomial equation

$$H_k(z^{-1})A(z^{-1}) + z^{-k}W_k(z^{-L}) - B^*(z^{-L}) = 1, \tag{A.11}$$

where $A(z^{-1})$ is defined by Eq. (3.37), and

$$H_k(z^{-1}) = 1 + h_{k,1}z^{-1} + \cdots + h_{k,2m(L-1)}z^{-2m(L-1)}, \tag{A.12}$$

$$W_k(z^{-L}) = w_{k,0} + w_{k,1}z^{-L} + \cdots + w_{k,2m-1}z^{-(2m-1)L}, \tag{A.13}$$

$$B^*(z^{-L}) = b_1 z^{-L} + b_2 z^{-2L} + \cdots + b_{2m} z^{-2mL}. \tag{A.14}$$

The coefficients of $B^*(z^{-L})$ are the same as those in $B(z^{-1})$ (computed by Eq. (3.33)).

Based on the internal signal model [57] of $d[n]$, $A(z^{-1})d[n] = 0$ at the steady state. Combining this with Eq. (A.11) yields

$$\left(1 - z^{-k}W_k(z^{-L}) + B^*(z^{-L})\right) d[n] = H_k(z^{-1})A(z^{-1})d[n] = 0, \tag{A.15}$$

which gives

$$\begin{aligned} d[n] &= z^{-k}W_k(z^{-L})d[n] - B^*(z^{-L})d[n] \\ = w_{k,0}d[n-k] &+ w_{k,1}d[n-k-L]\cdots + w_{k,2m-1}d[n-k-(2m-1)L] \\ -b_1 d[n-L] &- b_2 d[n-2L] - \cdots - b_{2m}d[n-2mL]. \end{aligned} \tag{A.16}$$

Replacing n with $nL + k$, we have

$$\begin{aligned} d[nL+k] = w_{k,0}d[nL] &+ \cdots + w_{k,2m-1}d[(n-(2m-1))L] \\ -b_1 d[(n-1)L+k] &- \cdots - b_{2m}d[(n-2m)L+k]. \end{aligned} \tag{A.17}$$

Recalling $d_L[n] = d[nL]$ and $y_k[n] \triangleq d[nL+k]$, it follows that Eq. (A.17) can be written as Eq. (3.28).

Now consider solving Eq. (A.11). Expanding the equation and collecting the coefficients of z^{-i}'s ($i = 1, 2, \ldots, 2mL$), one can get $2mL$ linear equations with $2mL$ unknowns, which can be written in matrix form as Eq. (3.34). □

A.4 Proof of Theorem 3.4

Proof. To see first (3.55), we construct

$$F_k(z^{-1})A(z^{-1}) + z^{-k}W_k(z^{-M}) + z^{-k}V_k(z^{-N}) = 1, \tag{A.18}$$

where

$$F_k(z^{-1}) = 1 + f_1 z^{-1} + \cdots + f_l z^{-l}, \tag{A.19}$$

$$W_k(z^{-M}) = w_{k,0} + w_{k,1}z^{-M} + \cdots + w_{k,t_1}z^{-t_1 M}, \tag{A.20}$$

$$V_k(z^{-N}) = v_{k,0} + v_{k,1}z^{-N} + \cdots + v_{k,t_2}z^{-t_2 N}. \tag{A.21}$$

Multiplying both sides of (A.18) with $d[n]$ and dropping the trivial term $F_k(z^{-1})A(z^{-1})d[n]$, we have

$$d[n] = z^{-k}W_k(z^{-M})d[n] + z^{-k}V_k(z^{-N})d[n], \tag{A.22}$$

namely,

$$d[n] = \sum_{i=0}^{t_1} w_{k,i}d[n-k-iM] + \sum_{j=0}^{t_2} v_{k,j}d[n-k-jN]. \tag{A.23}$$

Let $d[n]$ be the k-th data point of the i-th batch, i.e. $d[n] \leftrightarrow b_i[k]$, then based on the batch definition (see (3.53)), we have $d[n-k] \leftrightarrow d[iL] \leftrightarrow b_{i-1}[k]$. Recall that the indices n_1 and n_2 are chosen such that $d_M[n_1] \leftrightarrow d_N[n_2] \leftrightarrow b_{i-1}[L]$. Thus we get $d[n-k] \leftrightarrow d_M[n_1] \leftrightarrow d_N[n_2]$, or $(n-k)T = n_1MT = n_2NT$ based on their time-stamp equivalence. Now the time stamps of the summation terms in (A.23) are

$$\begin{aligned} t\{d[n-k-iM]\} &= (n-k-iM)T \\ &= (n_1-i)MT = t\{d_M[n_1-i]\}, \end{aligned} \tag{A.24}$$

$$\begin{aligned} t\{d[n-k-jN]\} &= (n-k-jN)T \\ &= (n_2-j)NT = t\{d_N[n_2-j]\}. \end{aligned} \tag{A.25}$$

Thus we get

$$d[n-k-iM] \leftrightarrow d_M[n_1-i], \tag{A.26}$$

$$d[n-k-jN] \leftrightarrow d_N[n_2-j]. \tag{A.27}$$

In other words, (3.55) will be satisfied as long as (A.23), or its equivalent from (A.18) is satisfied.

Now consider solving (A.18). Expanding the equation and collecting the coefficients of z^{-i}'s ($i = 1, 2, \ldots, l+m$), one can get $(l+m)$ linear equations with $(l+t_1+t_2+2)$ unknowns, which can be written in the matrix form as (3.56). □