

©Copyright 2016

Arom Boekfah

# Output-Boundary Regulation Using Event-Based Feedforward for Nonminimum-Phase Systems

Arom Boekfah

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

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Santosh Devasia, Chair

Brian Fabien

Joseph Garbini

Program Authorized to Offer Degree:  
Mechanical Engineering

University of Washington

**Abstract**

Output-Boundary Regulation Using Event-Based Feedforward for Nonminimum-Phase Systems

Arom Boekfah

Chair of the Supervisory Committee:  
Professor Santosh Devasia  
Mechanical Engineering

This research addresses a disturbance-rejection problem for nonminimum-phase systems. Exact disturbance cancellation can be obtained by using inversion-based feedforward, but preview information of disturbances is required for nonminimum-phase systems. The output-boundary regulation (OBR) approach is proposed when sufficient preview information is not available. In particular, OBR rapidly transitions the output to a desired value so that the effects of disturbances are maintained within given boundaries. The main contributions of this research are to develop a rapid output transition technique using inversion-based feedforward and to quantify manageable types of disturbances for the proposed OBR using an event-based feedforward approach. Simulation results for linear and nonlinear nonminimum-phase system examples are presented to illustrate the proposed OBR approach.



## TABLE OF CONTENTS

	Page
List of Figures . . . . .	iii
List of Tables . . . . .	viii
Chapter 1: Introduction . . . . .	1
1.1 Research Goal and Impact . . . . .	1
1.2 What is the Challenge? . . . . .	3
1.3 Research Plan and Timeline . . . . .	3
Chapter 2: Limitations of Feedback and Feedforward for Nonminimum-Phase Systems . . . . .	5
2.1 Feedback-Based Control . . . . .	5
2.2 Feedforward-Based Control . . . . .	6
2.3 Event-Based Control . . . . .	9
2.4 Feedforward-Based Output Transition . . . . .	12
2.5 Main Contributions of this Research . . . . .	14
Chapter 3: Preliminary Test of the Idea Using an AFM Example . . . . .	15
3.1 Problem Formulation . . . . .	16
3.2 Method and Results . . . . .	22
Chapter 4: Output-Boundary Regulation Using Event-Based Feedforward for Linear Systems . . . . .	36
4.1 Problem Formulation and Solution Approach . . . . .	37
4.2 Output-Transition Control . . . . .	38
4.3 Performance with Multiple Transitions . . . . .	48
4.4 Numerical Example . . . . .	53
Chapter 5: Extension to Nonlinear Nonminimum-Phase Systems . . . . .	64
5.1 Problem Formulation and Solution Approach . . . . .	65

5.2	Output-Transition Input . . . . .	69
5.3	Simulation Results and Discussion . . . . .	79
Chapter 6:	Conclusions . . . . .	99
6.1	Future Works . . . . .	101
	Bibliography . . . . .	102
Appendix A:	Perturbation Analysis for a Linear System Example . . . . .	109
A.1	Effects of Zero Locations . . . . .	109
A.2	Constraints on the Input and Maximum Allowable Power Values . . . . .	111
A.3	Effects of Noise . . . . .	116
A.4	Effects of Sampling Time . . . . .	117
Appendix B:	Perturbation Analysis for a Nonlinear System Example . . . . .	121
B.1	Effects of Zero Locations . . . . .	121
B.2	Constraints on the Input and Maximum Allowable Power Values . . . . .	121
B.3	Effects of Noise . . . . .	122
B.4	Effects of Sampling Time . . . . .	123
Appendix C:	Additional Information . . . . .	126
C.1	MATLAB Code for a Linear System Example . . . . .	127
C.2	MATLAB Code for a Nonlinear System Example . . . . .	133

## LIST OF FIGURES

Figure Number	Page
1.1 Scan Frequency versus scan size with a variety of positioning systems and control techniques: stiff samples (triangles) and soft samples (asterisks). The straight lines are least-square-error fits of the data points. . . . .	2
1.2 Research Plan . . . . .	4
2.1 Root locus of the example nonminimum-phase system $G_1 = -(s - 1)/(s + 2)$ using the proportional gain. . . . .	6
2.2 Closed-loop system using LQR with the cost functional $J_{LQR} = \int_0^\infty (y'y + u'Ru)dt$ for the example system $G_1 = -(s - 1)/(s + 2)$ . Left plot shows the closed-loop poles using different input weight $R$ . Right plot shows the corresponding step response (with zero initial state $x(0) = 0$ ). . . . .	7
2.3 Closed-loop system using LQR with the cost functional $J_{LQR} = \int_0^\infty (y'y + u'Ru)dt$ for the example system $G_2 = -(s - 1)/(s + 2)(s + 3)$ . Left plot shows the closed-loop poles using different input weight $R$ . Right plot shows the corresponding step response. . . . .	7
2.4 Schematic of system. A disturbance $d$ adds to the plant output $z$ to yield the output $y$ , which needs to be maintained at the nominal value $y_{nom} = 0$ . . . .	8
2.5 Exact disturbance rejection using noncausal inverse input for the example system $G_1 = -(s - 1)/(s + 2)$ . (Top) Disturbance $d$ . (Middle) Output $y = z + d$ . (Bottom) Input $u$ . (Left plots) Noncausal inversion (the input, represented by the dashed line, starts before the transition occurs). (Right plots) (Unstable) Causal exact inversion leads to unbounded input (the input also diverges in the wrong direction). . . . .	9
2.6 Approximation inverse of the example system $G_1 = -(s - 1)/(s + 2)$ with minimum phase systems. (Dashed lines) Method 1: ignoring nonminimum-phase zero $G_{approx,1} = 1/(s + 2)$ . (Dashed-Dotted lines) Method 2: replacing nonminimum-phase with minimum phase zeros $G_{approx,2} = (s + 1)/(s + 2)$ . . . .	10
2.7 A double integrator system $\ddot{x} = u + d$ with $d = 0.05$ using the event-based control . . . . .	12
3.1 Schematic of proposed boundary regulation for the scalar output case. . . .	15
3.2 Schematic diagram of AFM vertical positioning. . . . .	16
3.3 Magnitude $ G(i\omega) $ of piezo dynamics $G(s)$ . . . . .	19

3.4	Sample profile example for one scan period at the scan frequency of 8 Hz. Left: sample profile $z_s$ versus time. Right: sample profile $z_s$ versus scan line $x_p$ . . . . .	23
3.5	Limits of feedback. Left plot: (dots) variation of closed loop poles with input weight $r$ ; (asterisks) open-loop poles; (diamonds) closed loop pole at $r = 16$ ; and (circles) reflection of zeros about imaginary axis. Right plot: step responses, with steady state output normalized to one — (solid line) $r = 16$ ; (dashed line) lower input cost with $r = 1$ ; and (dotted line) open loop response. . . . .	25
3.6	Results for one scan period at the scan frequency of 8 Hz without boundary regulation. Left plot: vertical positions versus time. Right plot: vertical positions versus scan line $x_p$ . Top: (solid line) sample profile $z_s$ ; (dashed line) vertical position of the piezo $z_p$ . Bottom: (solid line) cantilever deflection $z_c$ ; (dotted line) sample deformation $z_d$ ; (dashed line) the region's boundary [0 10] nm. . . . .	26
3.7	Piezo transition results. Left: output $z_p$ for 1 ms: (dashed line) approach 1 (step input); (solid line) approach 2 (exact inverse). Right: input and output during the transition time of 0.02 ms. Top right: inputs: (dashed line) $u_{ff,\Delta}$ in Eq. (3.27); (solid line) $0.1 \times u_{inv}$ in Eq. (3.29). Bottom right: output $z_p$ : (dash-dotted line) ramp signal; (dashed line) output $z_p$ with $u_{ff,\Delta}$ ; (solid line) desired piezo position $z_{p,d}$ and output $z_p$ with $u_{inv}$ . . . . .	31
3.8	Results. Left: approach 1 (step input) at 31 Hz with the transition-check interval $t_{ff} = 0.17$ ms. Right: approach 2 (exact inverse) at 56 Hz with the transition-check interval $t_{ff} =$ the transition time $t_{tran} = 0.02$ ms. Top: sample profile $z_s$ . Middle: cantilever deflection $z_c$ : (solid line) cantilever deflection with boundary regulation; (dotted line) cantilever deflection without boundary regulation; (dashed line) the region's boundary [0 10] nm. Bottom: Inputs: (solid line) additional feedforward input $u_{ff}$ ; (dashed line) feedback and reference DC gain feedforward input $0.1u$ , with $u$ from Eq. (3.19). . . . .	33
3.9	Magnification of Fig. 3.8 approach 2 (exact inverse). Left: from Fig. 3.8(d) and (f). Right: magnification of the left plots between 2.2 ms and 2.6 ms. Top: cantilever deflection: (solid line) cantilever deflection $z_c$ ; (dashed line) region's boundary [0 10] nm; (dotted line) transition trigger values [2.5 7.5] nm. Bottom: additional feedforward input $u_{ff}$ . . . . .	34
3.10	Maximum and minimum cantilever deflection $z_c$ at different scan frequencies: (cross) maximum $z_c$ ; (plus sign) minimum $z_c$ ; (dashed line) the region's boundary. Top: without boundary regulation. Middle: with boundary regulation, approach 1 – step input (the transition-check interval $t_{ff} = 0.17$ ms). Bottom: with boundary regulation, approach 2 – exact inverse (the transition-check interval $t_{ff} = 0.02$ ms). . . . .	35

4.1	Schematic of output-boundary regulation. The output $y$ is maintained within given bounds $[-\bar{Y}, \bar{Y}]$ . If the output-error is outside a transition-trigger region $(-\bar{y}, \bar{y}) \subset (-\bar{Y}, \bar{Y})$ , then the proposed OBR approach triggers a transition of the output $y$ back towards the nominal value $y_{nom} = 0$ . . . . .	36
4.2	Normalized optimal single plant output transition and the optimal input. The optimal plant output $z = \bar{z}_{ff}$ is (i) transitioned from (steady-state value) 0 to the nominal (steady state) value $\bar{z} = \bar{y}$ during the transition time $T_{tt}$ and (ii) maintained constant afterward. The normalized undershoot $z_{us}$ and overshoot $z_{os}$ are shown to occur at time instants $t = T_{us}$ and $t = T_{os}$ . . . . .	39
4.3	Flexible structure example. . . . .	53
4.4	Disturbance $d$ with a disturbance ramp-rate of $\dot{D}^*$ . . . . .	55
4.5	The nominal transition of plant output from $z(0) = 0$ to $z(T_{tt}) = 1$ with a transition time of $T_{tt} = 1$ s. (Left) Optimal desired trajectory and nominal single-transition input $\bar{u}_{T_{tt},nom}$ . (Right) States $x_{1,nom}$ and $x_{2,nom}$ . . . . .	57
4.6	OBR performance with higher trigger frequency. OBR with a minimum trigger time of (Left): $T_{trig} = 1$ s and (Right): $T_{trig} = 0.1$ s. (Top): OBR input $u_{ff,k}$ . (Middle and Bottom) Output $y$ . . . . .	59
4.7	Simulation results with different minimum trigger time $T_{trig}$ . . . . .	60
4.8	Example results. (Left): DC-gain feedforward (without OBR). (Right): Step-input OBR. (Top): Disturbance $d$ . (Middle) Input $u$ . (Left) Input $u_{ref}$ , Eq. (4.78). (Right) Input $u_{st,k}$ , Eq. (4.79). (Bottom): Output $y$ . . . . .	61
4.9	Simulation results. Variation $\Delta\dot{D}^*$ in the acceptable ramp rate $\dot{D}^*$ with parametric uncertainties $\Delta\beta = \Delta\zeta_{p,1} = \Delta\zeta_{p,2} = \Delta\omega_{z,1} = \Delta\omega_{z,2} = \Delta\omega_{p,1} = \Delta\omega_{p,2}$ . . . . .	63
5.1	Schematic of proposed OBR. The output-tracking error $e$ is reset to zero, i.e., $e = 0$ at time instants $t_k$ , with $k = 1, 2, \dots$ ), and separated by at-least time $T_{trig}$ , i.e., $t_{k+1} - t_k \geq T_{trig}$ , when the output-tracking error $ e(t_k) $ is greater than a set (transition-trigger) value $\bar{e}$ , which is less than the error-boundary value $\bar{E}$ , i.e., $ e(t_k)  = \bar{e} < \bar{E}$ . . . . .	64
5.2	Schematic of output tracking for the nonlinear system $G$ . . . . .	65
5.3	Transition polynomial: (left) main polynomial $\hat{y}_{tran}$ , (middle) flexible polynomial $\tilde{y}_{tran}$ , (right) transition polynomial $y_{tran} = \hat{y}_{tran} + \tilde{y}_{tran}$ . . . . .	67
5.4	Single transition from the initial output $y(T_i)$ to the final output $y(T_f)$ within the same transition time $T_{tt} = T_f - T_i$ for all cases: (left) stable-inversion approach (with pre- and post-actuation inputs), (middle) state-transition-based approach (without post-actuation input), (right) output-transition-based approach (with post-actuation input), (top) output $y$ , and (bottom) Euclidean norm of the state $\ x\ _2$ . . . . .	68

5.5	Internal state $\eta$ transition to $\eta_f$ is completed: by the output transition time $t = T_f$ with S-OBR (top) and after the output transition at time $t = T_f$ with O-OBR (bottom). In both cases, the output remains at $y_f$ after time $t = T_f$ till another transition is required. . . . .	74
5.6	Single output transition using S-OBR and O-OBR for the decoupled internal dynamics system example. . . . .	88
5.7	Results from O-OBR iteration process: (left plot) root mean squared (RMS) differences of stable $\eta_s$ and $\eta_u$ unstable internal states and (right plot) parameter $p_{1,m}$ in Eq. (5.100). . . . .	91
5.8	Single output transition using S-OBR and O-OBR for the coupled internal dynamics system example. . . . .	92
5.9	Undershoot of the output $y_{us}$ using different transition time $T_{tt}$ for a given trigger value $\bar{e} = 0.04$ : (top plots) S-OBR, (bottom plots) O-OBR, (left plots) undershoot of the output $y_{us}$ satisfies Eq. (5.129) with allowable change of the desired trajectory $\sup_{t \in [T_i, T_f]} \{y_d(t)\} = \bar{e}$ , and (right plots) undershoot of the output $y_{us}$ exceeds Eq. (5.129), therefore transition time $T_{tt}$ is too small. . . . .	94
5.10	Transition time $T_{tt}$ and undershoot of the output $y_{us}$ for a given trigger value $\bar{e}$ : (top) allowable undershoot $y_{us}$ in Eq. (5.128) with the corresponding transition time $T_{tt}$ and (bottom) actual undershoot of the output $y_{us}$ for a given trigger value $\bar{e}$ . . . . .	95
5.11	Comparative evaluation using S-OBR (dashed lines) and O-OBR (dotted lines): (top) transition time $T_{tt}$ with a given trigger value $\bar{e}$ and (bottom) maximum slope $\alpha_1$ of the desired trajectory in Eq. (5.120) with varying a trigger value $\bar{e}$ : (left) limits of bandwidth $f_{max}$ and (right) limits of input $u_{max}$ . . . . .	96
5.12	Example tracking results for the coupled internal dynamics system with a fixed transition time $T_{tt} = 1s$ using: (left plots) S-OBR and (right plots) O-OBR. . . . .	97
6.1	Output transition for linear systems: (left) SST and (right) OOT. . . . .	99
6.2	Output transition for nonlinear systems: (left) S-OBR and (right) O-OBR. . . . .	100
A.1	Effects of zero locations when using SST: (top plots) input $u$ , (bottom plots) output $z$ , (left plots) varying unstable zero $m_{z,u}$ , and (right plots) varying stable zero $m_{z,s}$ . . . . .	110
A.2	Effects of unstable zero locations $m_{z,u}$ using SST to: (top) the undershoot of the plant output $z_{us}$ , (middle) input magnitude $u^*$ , (bottom) the maximum disturbance ramp-rate $\dot{D}^*$ . The trigger time $T_{trig} = 1s$ . . . . .	111
A.3	Effects of stable zero locations $m_{z,s}$ using SST to: (top) the undershoot of the plant output $z_{us}$ , (middle) input magnitude $u^*$ , (bottom) the maximum disturbance ramp-rate $\dot{D}^*$ . The trigger time $T_{trig} = 1s$ . . . . .	112

A.4	Effects of zero locations when using OOT: (top plots) input $u$ , (bottom plots) output $z$ , (left plots) varying unstable zero $m_{z,u}$ , and (right plots) varying stable zero $m_{z,s}$ . . . . .	113
A.5	Effects of unstable zero locations using OOT to: (top) the undershoot of the plant output $z_{us}$ , (middle) input magnitude $u^*$ , (bottom) the maximum disturbance ramp-rate $\dot{D}^*$ . The trigger time $T_{trig} = 1s$ . . . . .	114
A.6	Effects of stable zero locations using OOT to: (top) the undershoot of the plant output $z_{us}$ , (middle) input magnitude $u^*$ , (bottom) the maximum disturbance ramp-rate $\dot{D}^*$ . The trigger time $T_{trig} = 1s$ . . . . .	115
A.7	Comparison of the single output transition between SST and OOT: (top) undershoot of the output $z_{us}$ (bottom) maximum input $u$ for single output transition. . . . .	116
A.8	Comparison of the single output transition between SST and OOT: (top) trigger value obtained from Eq. (4.57) in Lemma 2 and (bottom) estimation $\bar{y}/T_{tt}$ of disturbance ramp-rate $\dot{D}^*$ that can be handled between using SST and OOT. . . . .	117
A.9	Comparison of maximum input and power between using SST and OOT (top) maximum input $U_k^*$ and (bottom) maximum power $P_{max}$ required for different disturbance ramp rate $\dot{D}^*$ . . . . .	118
A.10	Tracking results with noise $N$ added to the output $y$ (the trigger time $T_{trig} = 1s$ ) with magnitude of: (left) trigger value ( $1 \times \bar{y}$ ) and (right) twice trigger value ( $2 \times \bar{y}$ ). . . . .	119
A.11	Effects of noise on the disturbance ramp rate $\dot{D}^*$ (the trigger time $T_{trig} = 1s$ ). . . . .	119
A.12	Tracking results with different sampling time $T_s$ (the trigger time $T_{trig} = 1s$ ): (left) sampling time $T_s = 4 \times 10^{-2}s$ and (right) sampling time $T_s = 10^{-5}s$ . . . . .	120
A.13	Effects of sampling time $T_s$ (the trigger time $T_{trig} = 1s$ ) on the maximum disturbance ramp-rate $\dot{D}^*$ . . . . .	120
B.1	Effects of zero locations on the slope of desired trajectory $\alpha_1$ : (top plots) input $u$ , (bottom plots) output $y$ , (left plots) S-OBR, and (right plots) O-OBR. . . . .	122
B.2	Comparison of maximum input and power between using S-OBR and O-OBR (top) maximum input $U_{max}$ and (bottom) maximum power $P_{max}$ required for different disturbance ramp rate $\dot{D}^*$ . . . . .	123
B.3	Effects of noise: (left plots) smaller noise magnitude and (right plots) larger noise magnitude. . . . .	124
B.4	Effects of sampling time using a fixed time step MATLAB ode1 function. . . . .	125

## LIST OF TABLES

Table Number	Page
2.1 Comparison of approximate inverse methods and a noncausal inverse approach at different frequencies of the desired trajectory. . . . .	11
3.1 Natural frequencies and damping ratios of $G(s)$ . . . . .	18
4.1 Comparative evaluation. Transition-trigger bound $\bar{y}$ , normalized undershoot $ z_{us} $ , input magnitude $u^*(T_{tt})$ , and allowable disturbance ramp-rate $\dot{D}^*$ . . . .	62
C.1 Summary of the proposed approach for linear and nonlinear systems . . . . .	126

## ACKNOWLEDGMENTS

I would like to thank my advisor Professor Santosh Devasia for everything during my Ph.D. study. This dissertation would not have been done without his invaluable, constant, and patient advice. I am very fortunate to have Professor Santosh Devasia as my advisor, thanks to Bantoon Srisuwan, Wanwisa Kisalang, and Dr. Jiradech Kongthon who introduced me to Professor Santosh Devasia. I would like to thank my committee members Professor Brian Fabien, Professor Joseph Garbini, and Professor Howard Chizeck for very useful suggestions and comments.

I would like to thank my family for their supports, Dr. Siriphan Manochewa for suggestions on writing and presentation, Nathan Banka for proofreading the abstract page, Dr. Thongchai Phairoh for writing recommendation letters for me many times, Wichien Uasomskul and Dr. Jackrit Suthakorn for supporting me to study abroad, and all of my friends during my study in the United States.

I would like to thank Mahidol University, Thailand, and the Royal Thai Government for the Royal Thai Government Scholarship between 2009 and 2015, Louis & Katherine Marsh Endowed Fund in Engineering for the fellowship during 2012, and the National Science Foundation (CMMI 1301452) for the financial support since 2013.

## DEDICATION

To my father who has passed away and my mother.

## Chapter 1

## INTRODUCTION

**1.1 Research Goal and Impact**

The goal of this research is to reject disturbances for nonminimum-phase systems especially when disturbances are not known ahead of time. Disturbance cancellation for nonminimum-phase systems is important in many critically technological applications, for example, nanopositioning systems, wind turbines, and vertical take-off and landing (VTOL) aircraft.

*1.1.1 Piezo-based positioning systems – Atomic Force Microscopes (AFMs)*

Precision positioning is important, e.g., in Atomic Force Microscopes (AFMs) when using nonminimum-phase nanopositioners during the imaging of soft samples such as cells. Too much output force can damage the soft sample and too less output force can lead to loss of contact [1,2]. Although control has helped to improve AFM performance, scanning over soft samples remains about a hundred times slower than the scanning for comparably-sized stiff samples (see Fig. 1.1), even with a variety of different control methods [1]. The scan frequency limit is caused by the inability to maintain small applied forces in the presence of disturbances arising from rapid changes in the sample height and/or material properties during high-frequency scanning — large forces lead to damage of soft samples. For example, a large sample, such as a live eukaryotic cell, can be imaged at the scan frequency of 3 – 4 Hz [3]. If control schemes can be developed to compensate for disturbances and maintain a small AFM force (the output) in a desired range during high-frequency scanning, then high-frequency AFM operation is possible. The resulting higher scan frequency operation will enable the investigation of fast biological processes such as lamellipodia formation during cell migration, which requires both high resolution and relatively large scan areas [4–6]. Thus, there is a need to increase the operating speed (scan frequency) of AFMs to investigate

nanoscale dynamic processes in soft samples.

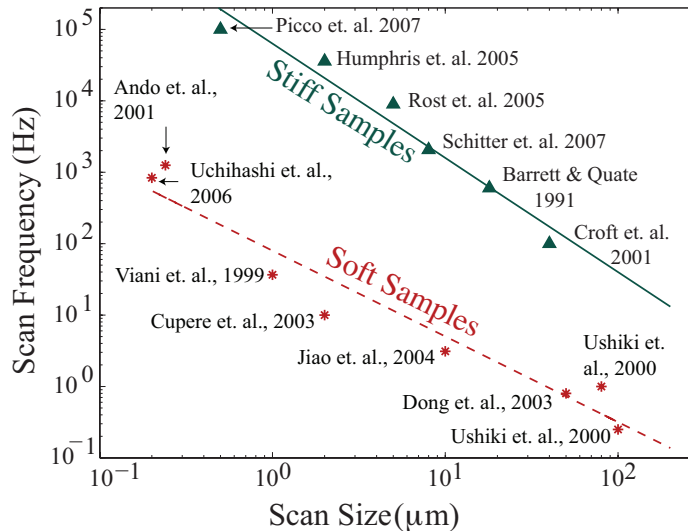


Figure 1.1: Scan Frequency versus scan size with a variety of positioning systems and control techniques: stiff samples (triangles) and soft samples (asterisks). The straight lines are least-square-error fits of the data points.

### 1.1.2 Wind Turbines – preview control is needed to regulate the wind disturbance

The turbine controller needs to regulate turbine speeds according to wind disturbances. Wind gusts are not known in advance and preview information is needed to accurately compensate for the wind disturbance. Feedback normally leads to a delay between the time that a wind disturbance arrives and the time that the the control actuator begins to mitigate resulting loads [7]. This delay could cause damage to the structure due to component bending [8]. Preview feedforward controller can improve the system performance when combined with feedback. Therefore, preview information of the wind disturbance needs to be measured, e.g., by using light detection and ranging (LIDAR) systems. However, measured wind disturbances can change during the preview time. Precision control of wind turbines provides better performance of the wind turbine system.

### *1.1.3 Vertical Take-Off and Landing (VTOL) Aircraft*

Tracking control for nonlinear nonminimum phase VTOL aircraft is challenging. Nonminimum phase behavior of VTOL aircraft occurs from the small body force generating moment [9]. Preview information of the desired trajectory is needed for exact output tracking of VTOL aircraft using a stable inversion technique [10]. However, wind disturbances can affect tracking performance of VTOL aircraft, since preview information of wind disturbances is normally not known ahead of time. Improving control can make VTOL aircraft track more aggressive output trajectory.

Improving disturbance cancellation can lead to better performance for nonminimum-phase systems.

### **1.2 What is the Challenge?**

Current methods cannot address the disturbance rejection problems for nonminimum-phase systems if preview information of disturbances is not available. Feedback has limits for nonminimum-phase systems. Feedforward can overcome the limits of feedback, but preview information is required for exact disturbance cancellation.

### **1.3 Research Plan and Timeline**

The research timeline is shown in Fig. 1.2. Preliminary test of the idea using an AFM example was presented in ASME 2013 Dynamic Systems Control Conference (DSCC) [11] and this paper was selected as a DSCC 2013 Best Paper Award Finalist in Theory. Performance quantification using a simplified flexible structure example was presented in 2015 American Control Conference (ACC) [12] and was published in IEEE Transactions on Control Systems Technology [13]. The extension of theoretical works from linear to nonlinear nonminimum phase systems was submitted to ASME 2016 Dynamic Systems and Control Conference (DSCC) [14].

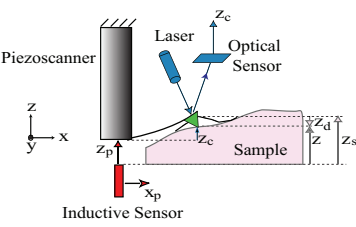
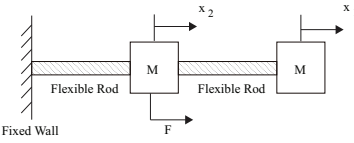
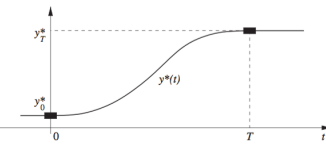
1) Preliminary idea	2) Theory	3) Extension of theoretical works
<p style="text-align: center;">Done</p> 	<p style="text-align: center;">Done</p> 	<p style="text-align: center;">Done</p>  <p style="text-align: center;">[15]</p>
<p style="text-align: center;">- Preliminary test of the idea</p> <p style="text-align: center;">ASME DSCC 2013 (Conference)</p>	<p style="text-align: center;">- Performance quantification using a simplified flexible structure example</p> <p style="text-align: center;">ACC 2015 (Conference) IEEE TCST 2016 (Journal)</p>	<p style="text-align: center;">- Extension to nonlinear systems</p> <p style="text-align: center;">ASME DSCC 2016 (Conference) (Submitted)</p>

Figure 1.2: Research Plan

## Chapter 2

## LIMITATIONS OF FEEDBACK AND FEEDFORWARD FOR NONMINIMUM-PHASE SYSTEMS

### 2.1 Feedback-Based Control

A feedback-based approach for precision output trajectory tracking (e.g., a regulator approach in [16,17]) has fundamental limitations for nonminimum-phase systems [18–20].

#### 2.1.1 Feedback Limitations for Nonminimum Phase Systems

The feedback limitation arises because of the need to maintain stability. For example, high controller gains can yield better disturbance rejection. However, high gains can lead to instability for non-minimum phase systems since the closed loop poles tend to the open-loop “unstable” zeros in the right-half plane (RHP). For example, a root locus plot for a nonminimum phase system example,

$$G_1 = \frac{y(s)}{u(s)} = \frac{-(s-1)}{s+2} \quad (2.1)$$

where  $y$  and  $u$  are the output and input of the system  $G_1$ , is shown in Fig. 2.1 and the closed loop system is unstable when the controller gain  $K < -2$  or  $K > 1$ . Moreover, high gains can lead to overshoot and undershoot problems for nonminimum-phase system [20]. Stability of the closed-loop system can be ensured with appropriate feedback design, for example, when using a linear quadratic regulator (LQR) to design a stabilizing feedback by minimizing the cost functional

$$J_{LQR} = \int_0^{\infty} (y'y + u'Ru)dt \quad (2.2)$$

and varying the input weight  $R$ . A root locus plot using LQR for the example system  $G_1$  is shown in Fig. 2.3 (left plot). In the limit when the input cost tends to zero, some of the closed loop poles move towards a reflection of the open-loop RHP zeros about the imaginary

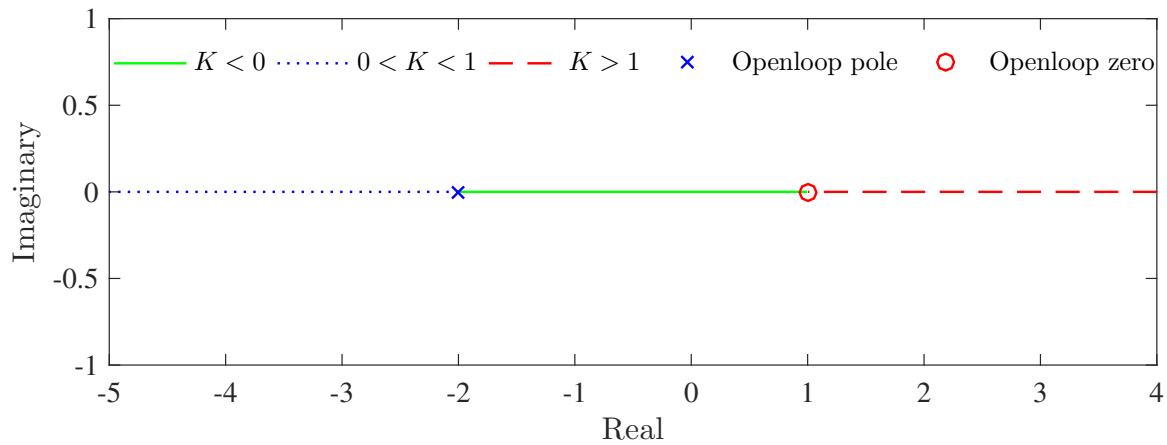


Figure 2.1: Root locus of the example nonminimum-phase system  $G_1 = -(s - 1)/(s + 2)$  using the proportional gain.

axis, e.g., see Fig. 2.3 (left plot) the second order nonminimum phase system example

$$G_2 = \frac{y(s)}{u(s)} = \frac{-(s - 1)}{(s + 2)(s + 3)}. \quad (2.3)$$

This limits the achievable settling time even with large inputs [19] as also shown in Figs. 2.2 and 2.3 (right plots). Therefore, feedback has limitations because of stability issues for nonminimum-phase systems.

## 2.2 Feedforward-Based Control

Inversion-based feedforward methods can overcome some of the limitations of feedback for nonminimum-phase systems (since feedforward does not directly influence system stability [21]), and achieve precision tracking of a desired output that can exactly cancel known disturbances and/or track a desired output, e.g., [22–25]. However, such (inversion-based) feedforward approaches tend to be non-causal for nonminimum-phase systems, and rely on the use of a-priori (preview) information of the disturbance. Therefore, inversion-based feedforward approaches are not suited for the situation when the disturbance is not known a-priori and the system is nonminimum phase.

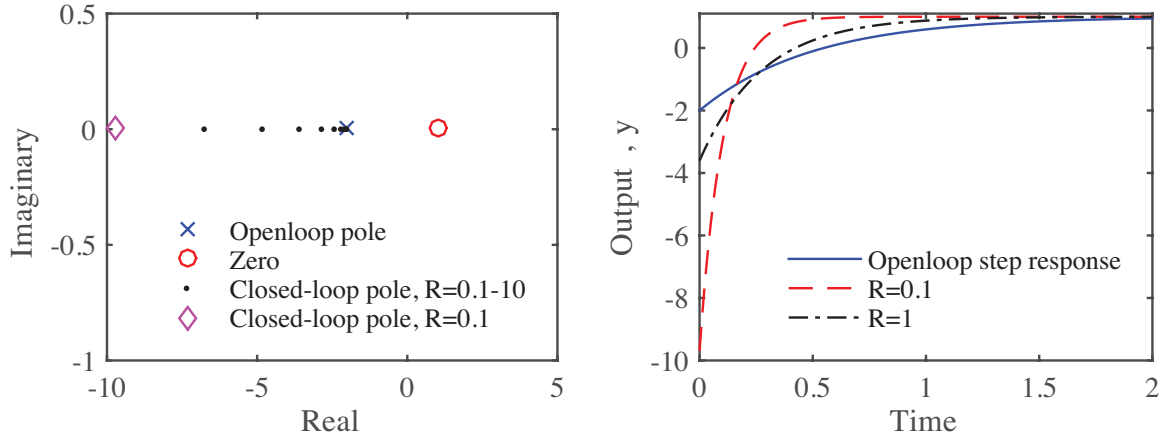


Figure 2.2: Closed-loop system using LQR with the cost functional  $J_{LQR} = \int_0^\infty (y'y + u'Ru)dt$  for the example system  $G_1 = -(s-1)/(s+2)$ . Left plot shows the closed-loop poles using different input weight  $R$ . Right plot shows the corresponding step response (with zero initial state  $x(0) = 0$ ).

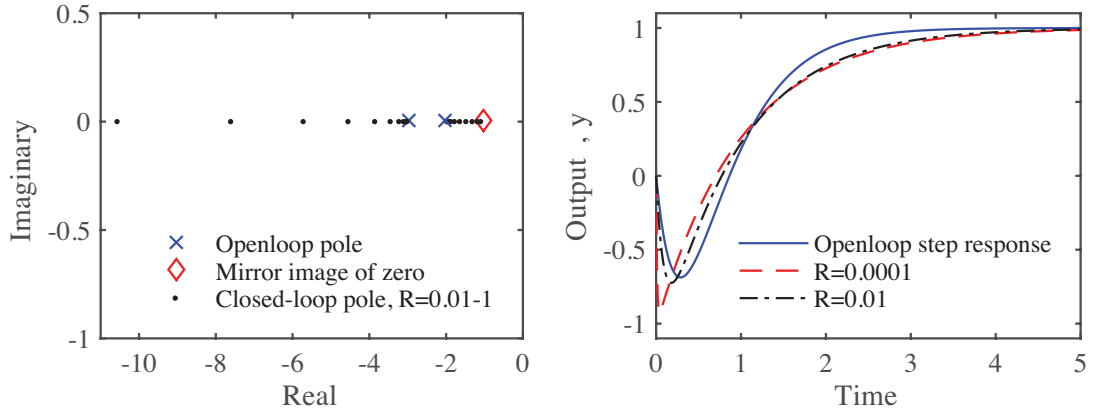


Figure 2.3: Closed-loop system using LQR with the cost functional  $J_{LQR} = \int_0^\infty (y'y + u'Ru)dt$  for the example system  $G_2 = -(s-1)/(s+2)(s+3)$ . Left plot shows the closed-loop poles using different input weight  $R$ . Right plot shows the corresponding step response.

### 2.2.1 Inversion-Based Feedforward Limitations for Nonminimum Phase Systems

Consider a system shown in Fig. 2.4, a disturbance  $d$  is added to the scalar output  $z$  of the plant  $G$  to yield the output  $y$ . For example, the plant output  $z$  needs to cancel

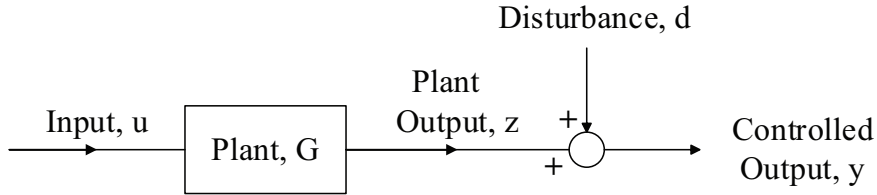


Figure 2.4: Schematic of system. A disturbance  $d$  adds to the plant output  $z$  to yield the output  $y$ , which needs to be maintained at the nominal value  $y_{nom} = 0$ .

the disturbance  $d$  to maintain the output error  $y$  at zero. The noncausal inverse can be used to achieve the exact cancellation of the disturbance provided preview information of the disturbance is available. However, preview information of the disturbance might not be known for some cases, e.g., a VTOL aircraft under wind disturbances, and the use of exact inverse will make the input  $u$  become unbounded. Results for the example system  $G_1 = -(s - 1)/(s + 2)$  using noncausal inverse are shown in Fig. 2.5 (left plots) and using exact inverse are shown in Fig. 2.5 (right plots). Note that an approximate inverse approach can be used with known desired trajectory (using minimum phase approximation without pre-actuation time). Figure 2.6 shows the example when the disturbance  $d$  is changed within a transition time  $T_{tt}$  of 1 s with the use of noncausal inverse and two approximation models

$$G_{approx,1} = \frac{1}{s + 2} \quad (2.4)$$

by ignoring the nonminimum phase zero and

$$G_{approx,2} = \frac{s + 1}{s + 2} \quad (2.5)$$

by replacing a nonminimum-phase zero with a minimum phase zero. Approximation methods can be used at low operating frequencies, i.e., slow desired trajectory (see variations on the maximum output error  $\max |y|$  versus the transition time  $T_{tt}$  (the time that the disturbance  $d$  changes from 0 to 1, e.g., see Fig. 2.6 (top plot) for the transition time of 1 s) in Table 2.1).

The limits of the inversion-based approach, when sufficient preview of the disturbance is not available, motivate the development of boundary-regulation rather than exact com-

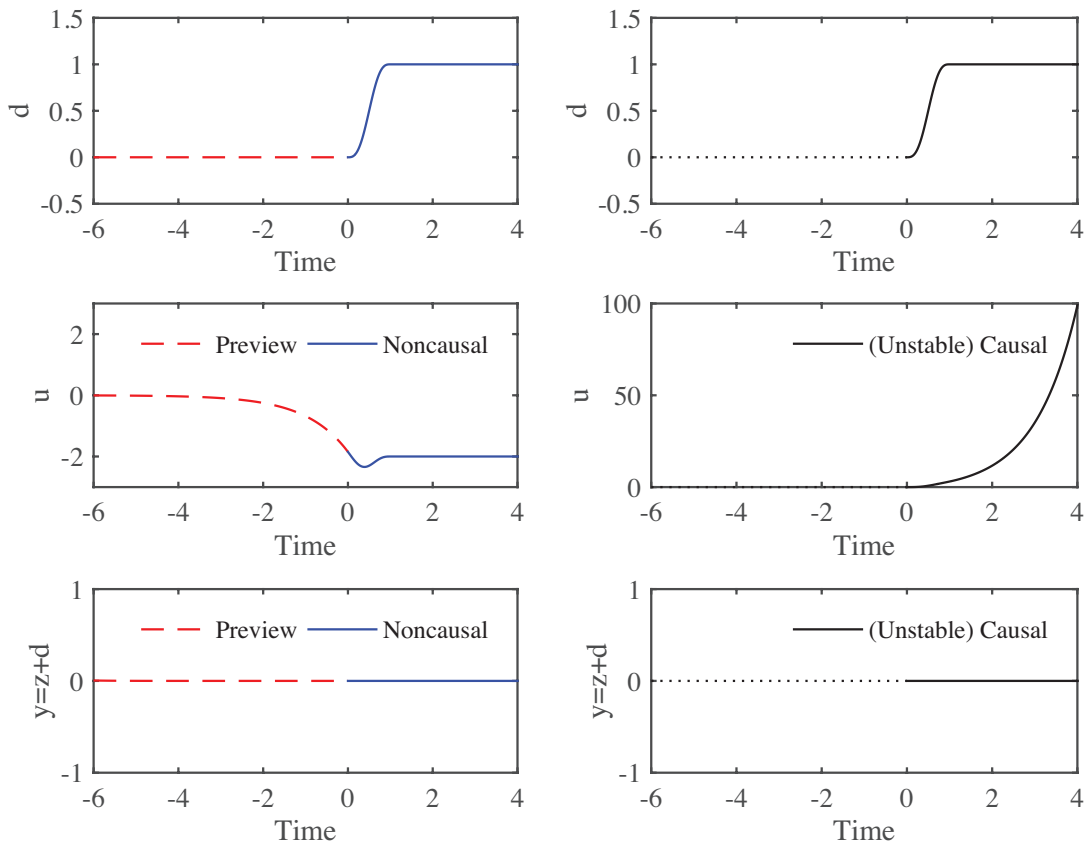


Figure 2.5: Exact disturbance rejection using noncausal inverse input for the example system  $G_1 = -(s - 1)/(s + 2)$ . (Top) Disturbance  $d$ . (Middle) Output  $y = z + d$ . (Bottom) Input  $u$ . (Left plots) Noncausal inversion (the input, represented by the dashed line, starts before the transition occurs). (Right plots) (Unstable) Causal exact inversion leads to unbounded input (the input also diverges in the wrong direction).

penetration. In particular, precomputed feedforward inputs are used to rapidly transition the output to a desired value so that effects of disturbances are maintained within given boundaries. The idea is similar to an event-based control approach [26–31].

### 2.3 Event-Based Control

Event-based control stabilizes systems through control actions (such as resetting the system state to zero) taken at specified events, e.g. when the output crosses specified values [28,29].

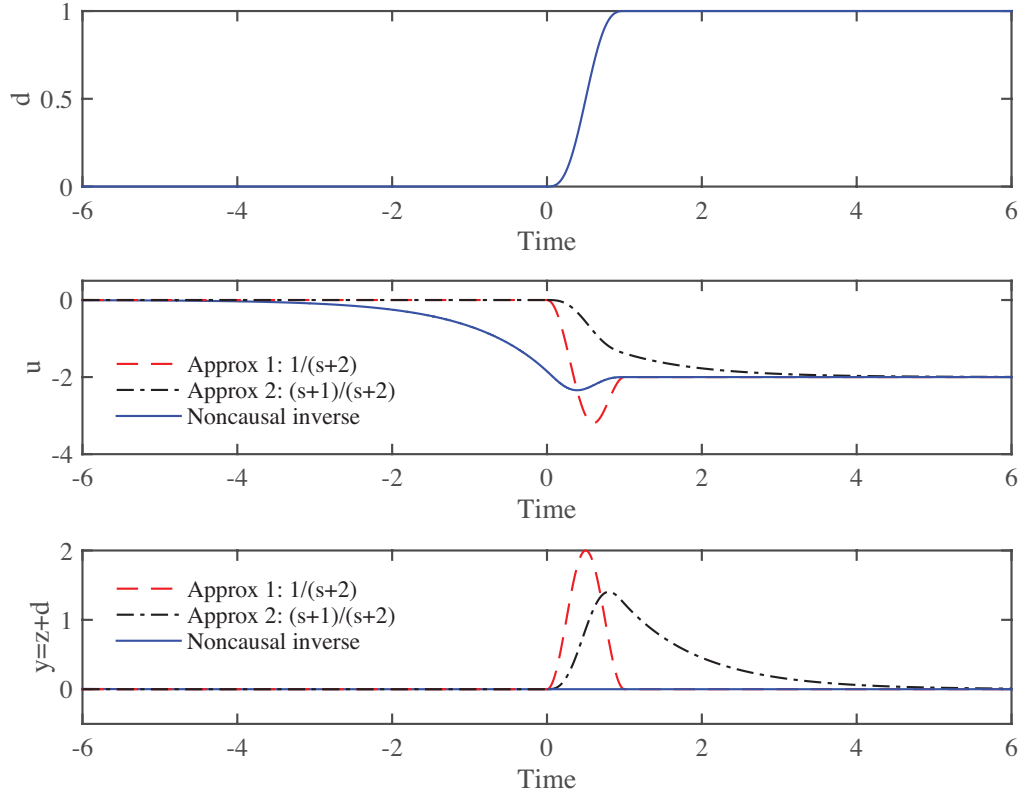


Figure 2.6: Approximation inverse of the example system  $G_1 = -(s - 1)/(s + 2)$  with minimum phase systems. (Dashed lines) Method 1: ignoring nonminimum-phase zero  $G_{approx,1} = 1/(s + 2)$ . (Dashed-Dotted lines) Method 2: replacing nonminimum-phase with minimum phase zeros  $G_{approx,2} = (s + 1)/(s + 2)$ .

Event-based feedback can also be used to reduce communication and control tasks while maintaining sufficient closed-loop performance [30]. For example, a double integrator system is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(t) \quad (2.6)$$

where  $d$  is a disturbance and the output  $y = x$  [28]. The objective is to maintain the output  $y$  at zero when the disturbance is known only when the output hits the trigger value, e.g., at time  $t_o$  and  $t_i$  shown in Fig. 2.7 (top plot). Otherwise, the control input  $u(t) = -d(t)$

Table 2.1: Comparison of approximate inverse methods and a noncausal inverse approach at different frequencies of the desired trajectory.

	Controlled output, $\max( y )$		
	$T_{tt} = 0.1$ s	$T_{tt} = 1$ s	$T_{tt} = 10$ s
(i) Approximation 1: $G_{approx,1} = 1/(s + 2)$	20 (0.1 s)	2 (0.95 s)	0.20 (8.6 s)
(ii) Approximation 2: $G_{approx,2} = (s + 1)/(s + 2)$	1.9 (3.7 s)	1.4 (4.21 s)	0.37 (10.1 s)
(iii) Noncausal inverse: $G = -(s - 1)/(s + 2)$	0.007 (6.1 s)	0.005 (6.8 s)	$2.1 \times 10^{-4}$ (14 s)

Preview time for using noncausal inverse is 6 s.

Settling times, when the output changes from 0 to 0.95, are shown in parentheses. Settling times of the desired trajectory are 0.08 s ( $T_{tt} = 0.1$  s), 0.80 s ( $T_{tt} = 1$  s), and 7.98 s ( $T_{tt} = 10$  s).

can be used with the known disturbance. The event-based control law can be given by

$$u(t) = \begin{cases} -\text{sgn}(y(t)) & , \text{ if } |y| > 1 \\ -L\hat{x}(t) & , \text{ if } |y| \leq 1 \end{cases} \quad (2.7)$$

where the gain  $L = [1 \ 2]$  and the generator state  $\hat{x}$  is given by

$$\dot{\hat{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad (2.8)$$

with initialized  $\hat{x} = x(t_i)$ , e.g., see Fig. 2.7 when  $|y| = 1$ . Example results with constant disturbance  $d = 0.05$  are shown in Fig. 2.7. The performance of event-based stabilization was compared to continuous time stabilization in [26, 32–34], and experimental results of event-based control for reference tracking and disturbance compensation of linear systems were studied in [31, 35, 36]. In addition to the linear case, stabilization of nonlinear systems using event-based control was studied in [37–39]. Tracking problems of nonlinear systems using event-based control were described in [40, 41]. Typically, the control inputs used

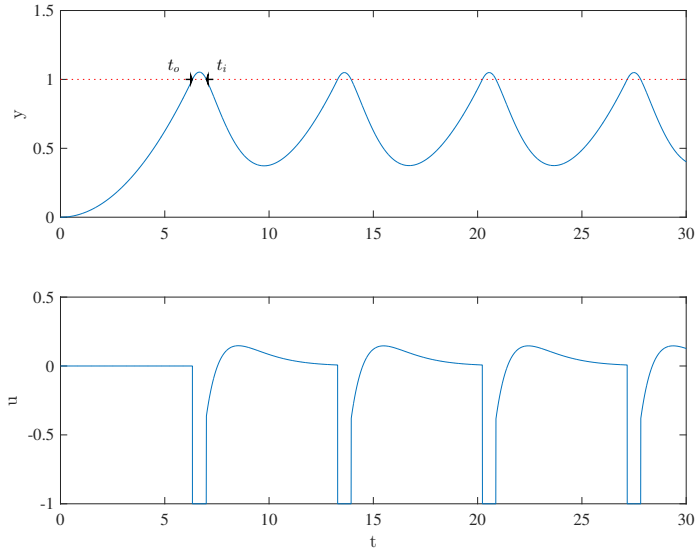


Figure 2.7: A double integrator system  $\ddot{x} = u + d$  with  $d = 0.05$  using the event-based control

in event-based control is the input that would have been generated by a continuous-time feedback controller [28], which could have stabilized the system, i.e., in the non-event-based control case. However, in general, feedback-based controllers (and the resetting using such feedback-based controllers) tend to have fundamental limitations for nonminimum-phase systems [42]. Recent work has shown that the use of feedforward can improve performance of event-based control [31]. This research expands on this use of feedforward to improve the resetting of the system for nonminimum-phase systems.

## 2.4 Feedforward-Based Output Transition

To obtain faster output transition, only the output  $y$  of the system can be transitioned during a transition interval rather than transitioning the entire state  $x$  of the system.

### 2.4.1 Optimal Output-Transition (OOT) for Linear Systems

Rapid transition of the output (as it nears the output boundary) can be achieved by using optimal (say, bounded input, minimal time) solutions to the standard state transition (SST) problem to change the system state  $x$  from its current value, e.g.,  $x(0)$  at time  $t = 0$  (close

to the output boundary) to a nominal state  $x(T_{tt}) = \bar{x}$  at time  $t = T_{tt}$ , away from the boundary. The transition time can be reduced further if only the system output needs to be transitioned from one value to another rather than the transition of the entire system state. For example, the optimal output transition problem is to change the output  $y$  from an initial value (say,  $y(t) = \underline{y}$ ) to a final value (say,  $y(t) = 0$  for all time  $t > T_{tt}$ ) while minimizing a given cost function. Extensions of the output transition problem, with pre- and post-actuation, was posed in [43], which led to the minimum-time solution from a pre-specified class of output trajectories. The general problem of optimal output transition (OOT) was posed and solved in [44], connections of the approach with the input shaping method was studied in [45, 46], and extensions of the OOT approach for nonlinear systems and actuator-redundant systems were recently developed in [47, 48]. This OOT approach to design feedforward is adapted in this research to rapidly transition the output to the nominal value for solving the output-boundary regulation problem.

#### 2.4.2 Output-Transition-Based OBR (O-OBR) Approach for Nonlinear Systems

An important issue in the OBR design is to rapidly transition the output away from the specified boundary at each output-reset event. However, as opposed to the linear case, computing solutions to optimal control problems for nonlinear systems can be non-trivial, e.g., [47]. For example, constraints for a given change in the output setpoint to be feasible has been studied recently in [49]. Another approach for changing the output set point (i.e., a single output transition) is to specify a sufficiently-smooth output-transition trajectory, e.g., using polynomials, and then use the stable-inversion approach to find the input [50]. However, this approach is not applicable with general output trajectories for nonminimum-phase systems if preview information about the transition is not available [22] since the resulting input is non-causal and requires pre-transition input before the output starts to change. The pre-transition input requirement can be avoided by appropriately modifying the desired trajectory [15, 51]. This polynomial-based approach [15, 51] achieves a transition from an initial system state  $x(T_i) = x_i$  to a specified final system state  $x(T_f) = x_f$ . A faster output transition, away from the boundary, can be achieved if only the system output  $y$

needs to be transitioned, e.g., from  $y(T_i) = y_i$  to a specified final system state  $y(T_f) = y_f$  rather than the entire system state  $x$ , e.g., [47, 52]. The main contribution of the current work is to extend the polynomial-based state-transition approach in [15, 51] to achieve an output transition (i.e., transition the output  $y$  to reset the error  $e$  to zero) within a prescribed time interval and then use post-transition input to eventually transition the system state  $x$  to the final desired value  $x_f$ . Although the current approach is not formulated to handle output constraints such as reducing overshoot in minimum-time problems (e.g., as studied in [53]), the added flexibility due to the potential use of post-actuation can lead to the reduction of the transition time.

A faster output transition can be obtained by using post-actuation and therefore disturbance rejection performance can be improved for nonminimum-phase systems.

## ***2.5 Main Contributions of this Research***

The main contributions of this research are:

1. to develop the output transition approach using inversion-based feedforward to rapidly bring the output to a desired value so that the effects of disturbances are bounded within given boundaries when using event-based control and
2. to quantify the types of disturbances that can be handled by the proposed OBR using an event-based feedforward approach.

## Chapter 3

**PRELIMINARY TEST OF THE IDEA USING AN AFM EXAMPLE**

This research<sup>1</sup> proposes a new control framework for nonminimum-phase systems — boundary regulation. In particular, the research will investigate the use of pre-computed feedforward trajectories (which can still achieve more rapid output transition than feedback methods) to enhance disturbance rejection even when a-priori knowledge of the disturbance is not available. The main idea is to use feedforward to transition the output of interest (say  $z_c$ ) away from the boundary of the desired region, e.g.,  $z_c \in [\underline{Z}_c, \overline{Z}_c]$ , for the scalar output case, as shown in Fig. 3.1. When the output is near the desired region’s boundary, e.g., when it reaches transition-trigger values  $z_c = \overline{z}_c$  or  $z_c = \underline{z}_c$ , the output is rapidly transitioned away from the region’s boundary ( $z_c = \overline{Z}_c, z_c = \underline{Z}_c$ ) using feedforward. A drawback of fast transitions is that large undershoot and peaking can occur when rapidly transitioning the output of a nonminimum-phase system. Therefore, it is important that the transition action is taken before the output reaches the region’s boundary to maintain boundary regulation,  $z_c \in [\underline{Z}_c, \overline{Z}_c]$ . The proposed approach is illustrated with simulation results for an AFM imaging example, and the increase in scan frequency with and without the proposed approach is evaluated.

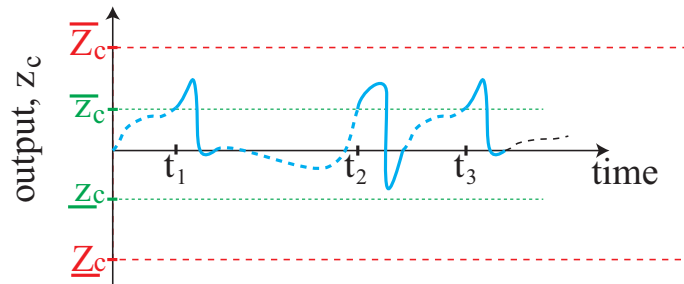


Figure 3.1: Schematic of proposed boundary regulation for the scalar output case.

---

<sup>1</sup>Material in this chapter was presented in ASME 2013 Dynamic Systems Control Conference (DSCC) [11].

### 3.1 Problem Formulation

The positioning problem in AFM is described in this section.

#### 3.1.1 AFM Positioning Scheme

The force  $f_s$  applied on the sample can be controlled using vertical positioning of the AFM probe (cantilever). The schematic diagram of AFM vertical positioning using a piezo positioner is shown in Fig. 3.2. The vertical position of the piezo  $z_p$  above the sample base is

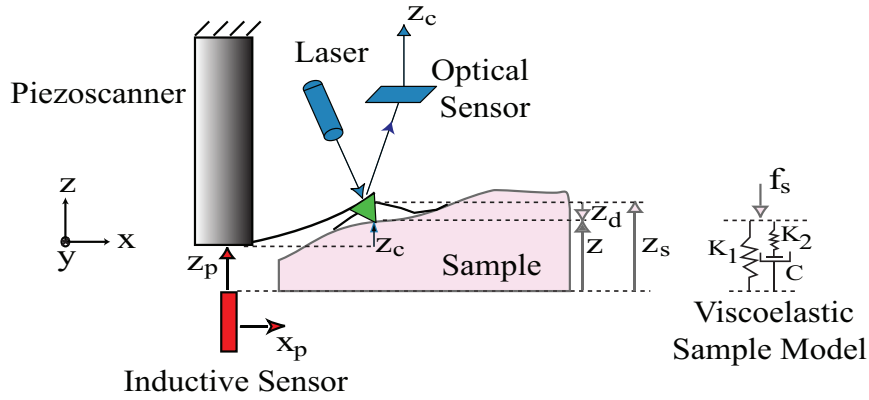


Figure 3.2: Schematic diagram of AFM vertical positioning.

measured with an inductive sensor and the deflection  $z_c$  of the AFM cantilever is measured with an optical sensor. The sample height  $z_s$  is deformed by  $z_d$  due to the force  $f_s$  between the cantilever tip and the sample. The net position  $z$  of the cantilever tip measured with respect to the sample base is given by

$$z(t) = z_p(t) + z_c(t). \quad (3.1)$$

#### 3.1.2 Cantilever and Sample Models

The cantilever deflection  $z_c$  depends on the force  $f_s$  between the cantilever tip and the sample,

$$z_c(s) = G_c(s)f_s(s) \approx \frac{1}{K_c}f_s(s) \quad (3.2)$$

where the dynamics  $G_c$  of the cantilever is assumed to be a constant (with cantilever stiffness  $K_c$ ) as in previous works (e.g., [54]) since the cantilever bandwidth is substantially higher than the piezo dynamics.

The sample height  $z_s$  (which varies with lateral position  $x$ ) gets compressed to  $z$  due to deformation  $z_d$  caused by the sample force  $f_s$  with

$$z_d(s) = \left[ \frac{K_2 + cs}{K_1 K_2 + (K_1 + K_2)cs} \right] f_s(s) = [G_s(s)] f_s(s) \quad (3.3)$$

where the sample is modeled with the standard viscoelastic Kelvin model [55] shown in Fig. 3.2. At relatively-high scan frequencies (large  $s$ , e.g., at 6.0 Hz [56]), recent work has shown that cell walls can transition into a glassy state. The sample model can be simplified to

$$G_s(s) \approx \left[ \frac{1}{K_1 + K_2} \right] = \frac{1}{K_s} \quad (3.4)$$

where  $K_s$  is the effective sample stiffness, which leads to the simplification, from Eqs. (3.2) and (3.4),

$$z_d(t) = \frac{1}{K_s} f_s(t) = \frac{K_c}{K_s} z_c(t). \quad (3.5)$$

It is noted that the sample stiffness  $K_s$  can vary with the lateral position  $x$  over the sample surface due to variations in the sample height and sample material properties. Moreover, the sample stiffness can vary nonlinearly with the sample deformation due to contact effects, e.g., [57–60] and surface forces [61]. In typical imaging applications, the sample stiffness  $K_s$  is nonlinear and unknown, which makes it difficult to estimate the sample deformation  $z_d$  for a given sample force  $f_s$ . Nevertheless, if the sample force can be maintained to be small, then the sample deformation is also small.

### 3.1.3 Piezo Model

The vertical position  $z_p$  of the piezo tip can be adjusted through the piezo input  $u$ . For small motions (compared to the range of the piezo), the nonlinearities in the piezo dynamics in the vertical direction  $z$  could be neglected and represented by a linear (but nonminimum

phase) dynamics  $G(s)$

$$G(s) = \frac{z_p(s)}{u(s)} = \frac{K \left( \frac{\omega_{p1}^2 \omega_{p2}^2}{\omega_z^2} \right) [s^2 - 2\zeta_z \omega_z s + \omega_z^2]}{\left[ s^2 + 2\zeta_{p1} \omega_{p1} s + \omega_{p1}^2 \right] \left[ s^2 + 2\zeta_{p2} \omega_{p2} s + \omega_{p2}^2 \right]} \quad (3.6)$$

with the output piezo position  $z_p$  in nm and the input  $u$  in volts. In the following, the low-frequency (DC) gain  $K$  of the system is chosen to be 10 (nm/V) which is a typical displacement-per-volt range for vertical piezo positioners. Similarly, model  $G$  parameters that are representative of AFM piezo dynamics are given in Table 3.1.

Table 3.1: Natural frequencies and damping ratios of  $G(s)$ .

	$\omega$ (rad/s)	$\zeta$
Zeros	$\omega_z = 2\pi 4000$	$\zeta_z = 0.1$
First resonance poles	$\omega_{p1} = 2\pi 2000$	$\zeta_{p1} = 0.1$
Second resonance poles	$\omega_{p2} = 2\pi 6000$	$\zeta_{p2} = 0.1$

Note that the second natural frequency  $\omega_{p2}$  of vibration in the model  $G$  in Eq. (3.6) is chosen to be three times the first natural frequency  $\omega_{p1}$  of vibration, and the zero  $\omega_z$  is interlaced between the poles of the system, as in typical flexible structure models. The frequency response of the example system (with the above parameter values) is shown in Fig. 3.3.

The above model of the piezoscanner dynamics: (a) is stable; (b) has zeros on the right half of the complex plane, i.e., is nonminimum phase, as in typical SPM positioning systems, e.g., [62, 63]; and (c) has relative degree two. The relative degree, the difference between the order of the transfer function's denominator and numerator, being two implies that the input voltage  $u$  can directly change the second derivative of the output [64], i.e., the input  $u$  directly changes the acceleration of the output position  $z_p$  as in typical mechanical systems. Note that the use of filters to smooth the input or the output tends to increase the relative degree of the system.

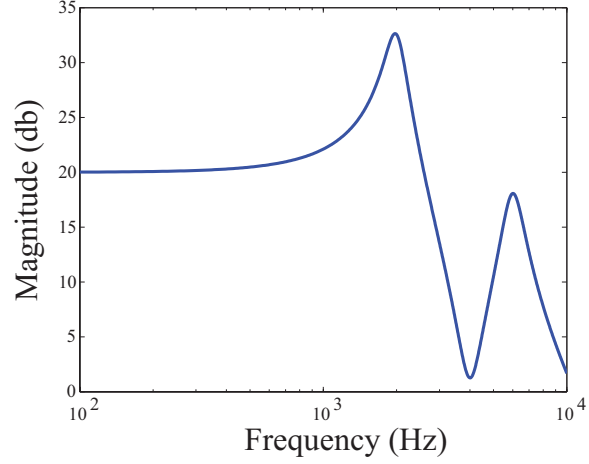


Figure 3.3: Magnitude  $|G(i\omega)|$  of piezo dynamics  $G(s)$ .

A state space model of the piezo dynamics (in modal, observer canonical form) is

$$\begin{aligned}\dot{X}(t) &= AX(t) + Bu(t) \\ z_p(t) &= CX(t)\end{aligned}\tag{3.7}$$

where

$$A = \begin{bmatrix} -2\zeta_{p1}\omega_{p1} & 1 & 0 & 0 \\ -\omega_{p1}^2 & 0 & 0 & 0 \\ 0 & 0 & -2\zeta_{p2}\omega_{p2} & 1 \\ 0 & 0 & -\omega_{p2}^2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -2.6707 \times 10^4 \\ 1.3156 \times 10^9 \\ 2.6707 \times 10^4 \\ 2.3717 \times 10^9 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}.\tag{3.8}$$

**Remark 1.** (*Effect of Sample Force on Piezo*): In general, the sample force  $f_s$  can affect the piezo position  $z_p$  [65]. However, when imaging soft samples, the force  $f_s$  applied on the sample needs to be maintained small — in the pico-newton range to avoid sample damage [66–68]. This relatively small force (as opposed to potentially large forces allowed when imaging stiff samples) does not significantly affect the piezo position  $z_p$  and is neglected in the model.

### 3.1.4 Estimating Sample Topography

During imaging in the contact mode, the cantilever tip position  $z$  (in Eq. (3.1)) is atomically close to the deflected surface, i.e.,

$$z(t) = z_p(t) + z_c(t) = z_s(t) - z_d(t) \quad (3.9)$$

which results in the following expression for the sample height  $z_s$  of the sample surface, from Eq. (3.5)

$$z_s(t) = z_p(t) + z_c(t) + z_d(t) = z_p(t) + z_c(t) \left(1 + \frac{K_c}{K_s}\right). \quad (3.10)$$

Note that both the piezo tip position  $z_p$  and the cantilever deflection  $z_c$  can be measured [69] or estimated [54,70]. In contrast, the sample deformation  $z_d$  (with potentially unknown variation in sample properties such as stiffness) is not easily estimated nor measured. However, if the sample deformation is small, then the sample height  $z_s$  can be approximated from AFM measurements as (from Eq. (3.10))

$$z_s(t) \approx z_p(t) + z_c(t). \quad (3.11)$$

**Remark 2.** (*Small Sample Deformation in Stiff Samples*): For stiff samples (large  $K_s$ ) such as silicon-based calibration gratings, the sample deformation  $z_d$  tends to be small even with relatively-large sample forces  $f_s$ . Therefore, the sample topography can be estimated with the approximation in Eq. (3.11) that neglects the sample deformation.

**Remark 3.** (*Small Sample Deformation in Soft Samples*): For relatively soft samples (small  $K_s$ ), sample force  $f_s$  needs to be small to ensure small sample deformation  $z_d$  to estimate the sample topography with Eq. (3.11).

### 3.1.5 The Tracking Requirement for Soft Samples

The sample force  $f_s$  should be kept small when imaging soft samples [66]. For example, when imaging soft animal cells, which do not have the relatively-stiff cell wall present in plant cells, previous animal cell studies report that the AFM-applied force should be less

than 100 (pN) to avoid sample damage [67, 68]. This implies that the cantilever deflection needs to be kept small since a small sample force requirement, e.g.,

$$|f_s(t)| < \bar{F} \quad (3.12)$$

can be translated into a small cantilever deflection  $z_c$  requirement (from Eq. (3.2)), i.e.,

$$|z_c(t)| < \frac{\bar{F}}{K_c} = \bar{Z}_c \quad (3.13)$$

where the cantilever deflection (from Eq. (3.10))

$$z_c(t) = \frac{K_s}{K_s + K_c} [z_s(t) - z_p(t)] \quad (3.14)$$

depends on the piezo dynamics  $X$ , and the sample height  $z_s$  and sample deformation  $z_d$  act as unknown output disturbances.

With a relatively soft cantilever (stiffness 0.01 (N/m)), keeping forces below 100 (pN) will require maintaining the cantilever deflection to be below  $\bar{Z}_c = 10$  (nm). If the nominal force (during imaging) is kept at half of the maximum possible force, i.e., the nominal cantilever deflection is  $z_{c,nom} = 5$  (nm), then the variation in the cantilever deflection needs to be maintained less than 5 (nm), i.e.,

$$|z_c(t) - z_{c,nom}| < 5 \text{ (nm)} \quad (3.15)$$

to avoid excessive forces ( $z_c < 10$  nm) and to maintain contact between the sample and the surface ( $z_c > 0$ ). The cantilever deflection  $z_c$  can be controlled by adjusting the piezo tip position  $z_p$ . In particular, when the cantilever is perfectly maintained at the nominal cantilever deflection, i.e.,

$$z_c = z_{c,nom},$$

the desired (reference) vertical position of the piezo  $z_{p,ref}$  can be obtained from Eq. (3.14) as

$$z_{p,ref}(t) = z_s(t) - \left[1 + \frac{K_c}{K_s}\right] z_{c,nom}. \quad (3.16)$$

**Remark 4.** (*Need for Tracking during Imaging*): Since sample profile can be determined using Eq. (3.11) as in Remark 2, high precision in the tracking (of the sample surface with

*the piezo position) is not needed for imaging if sample and/or cantilever damage is not a concern. However, tracking of the sample surface is critical to limit forces applied to the sample and to reduce sample deformation (and damage) when imaging soft samples.*

### *3.1.6 Challenges in use of Feedforward in AFM*

For known disturbances, feedforward, which does not share the stability limitations of feedback [21], can be added to improve tracking performance of feedback systems [71]. For example, if the trajectory to be tracked is known (e.g., the lateral scan trajectories in an AFM the  $x-y$  directions in Fig. 3.2), then inversion-based feedforward methods can be used to precision positioning of the piezo — even for nonminimum-phase systems [1]. However, the difficulty with the vertical-positioning problem is that the sample surface is unknown. An innovative approach developed in [72] was to use the previous scanned line to infer about the sample topography in the next scan line and use this information to add feedforward inputs to achieve tracking of the sample surface with the piezo. An extension of this idea was developed as the zoom approach, which iteratively finds a feedforward input to achieve vertical positioning while the scan area is increased (or decreased) [69]. Nevertheless, when the surface properties change suddenly, e.g., due to a stiffness change or topography variations, there can be substantial increase in the applied force that cannot be predicted with such iterative methods. Thus, unknown variations in sample properties such as stiffness and topography limit the use of prior knowledge in feedforward-based corrections which motivate the development of boundary regulation.

## **3.2 Method and Results**

This section describes the boundary-regulation approach, and comparatively evaluates maximum scan frequencies, with and without, the boundary regulation approach.

### 3.2.1 Sample Profile for Simulations

Let the scan line  $x_p$  in Fig. 3.2, in the time domain, of length  $L = 30 \mu\text{ m}$  for a scan frequency of  $\omega_h$  (rad/s) be described by

$$x_p(t) = \frac{L}{2} \sin(\omega_h t + \phi) + \frac{L}{2}. \quad (3.17)$$

The sample profile  $z_s$  over the scan line  $x_p$ , where the maximum sample height  $H = 1 \mu\text{ m}$ , is given by

$$z_s(t) = \begin{cases} \frac{H}{2} \sin(\omega_v t + \phi) + \frac{H}{2} & , \text{ for } \begin{cases} \frac{L}{8} < x_p < \frac{3L}{8} \\ \frac{5L}{8} < x_p < \frac{7L}{8} \end{cases} \\ 0 & \text{otherwise.} \end{cases} \quad (3.18)$$

The frequency  $\omega_v = 4 \times \omega_h$  and the angle  $\phi = 3\pi/2$  radian. It is noted that the sample profile  $z_s$  was smoothed by a first order filter with a break frequency of 1,000 kHz twice to obtain a smooth sample profile in time. A sample profile example at the scan frequency  $f$  of 8 Hz ( $\omega_h = 2\pi \times 8$  rad/s) for one scan period ( $T = 1/8$  s = 125 ms) is shown in Fig. 3.4. Additionally, the ratio  $\alpha = K_s/K_c$  of the sample stiffness  $K_s$  to the cantilever stiffness  $K_c$

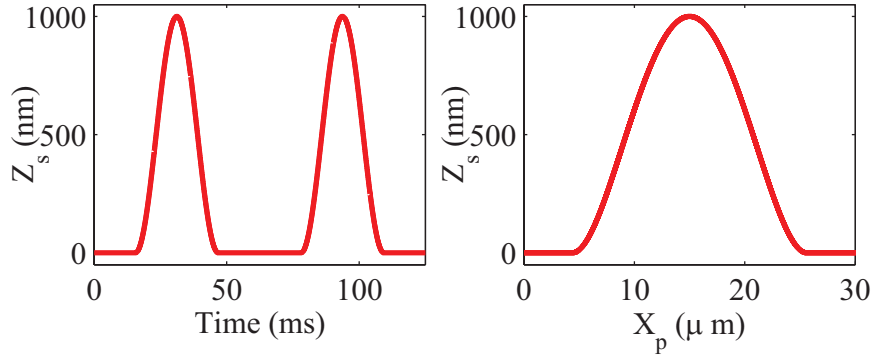


Figure 3.4: Sample profile example for one scan period at the scan frequency of 8 Hz. Left: sample profile  $z_s$  versus time. Right: sample profile  $z_s$  versus scan line  $x_p$ .

was chosen to be a constant,  $\alpha = 5$  [56], and is assumed to be known when simulating the system.

**Remark 5.** (*Varying Sample Stiffness*): In general, the sample stiffness will vary over the sample surface depending on variations in material properties and the sample profile. It tends to be unknown.

### 3.2.2 Without Boundary Regulation

For the case without boundary regulation, the control input  $u$  is chosen to be a combination of feedback and feedforward

$$u(t) = -K_{fb}X(t) + u_{ref}(t) \quad (3.19)$$

resulting in a closed-loop system (from Eq. (3.7))

$$\dot{X}(t) = [A - BK_{fb}]X(t) + Bu_{ref}(t) = A_{cl}X(t) + Bu_{ref}(t) \quad (3.20)$$

where  $u_{ref}$  is the reference feedforward input to the closed-loop system and  $K_{fb}$  is the feedback gain.

#### Feedback Design

The feedback controller  $K_{fb}$  can be designed using linear quadratic regulators (with small weights on the control effort) to optimize the transient response in the piezo positioning, i.e., to minimize the cost function

$$J(u) = \int_0^{\infty} [ru^2(\tau) + z_p^2(\tau)] d\tau. \quad (3.21)$$

Even with small input costs (small  $r$  in Eq. (3.21)), performance improvement with feedback is limited for nonminimum phase systems [18]. For example, the variation of the closed-loop poles of the piezo system (Eq. (3.20)) when input weight  $r$  (in Eq. (3.21)) changes from 1 to 1,000 is plotted in Fig. 3.5. As the input cost  $r$  is decreased, two poles tend to infinity; however, the other two poles go from open-loop location (asterisks) towards the reflection of the (unstable) system zeros about the imaginary axis (circle). The feedback gain is chosen as  $r = 16$  to increase the distance of the closed-loop-system poles from the imaginary axis which tends to decrease the settling time. As seen in Fig. 3.5, the settling time with the closed-loop system (with input cost choice  $r = 16$ ) is smaller than the open-loop system and is even smaller than the case with lower input cost,  $r = 1$ .

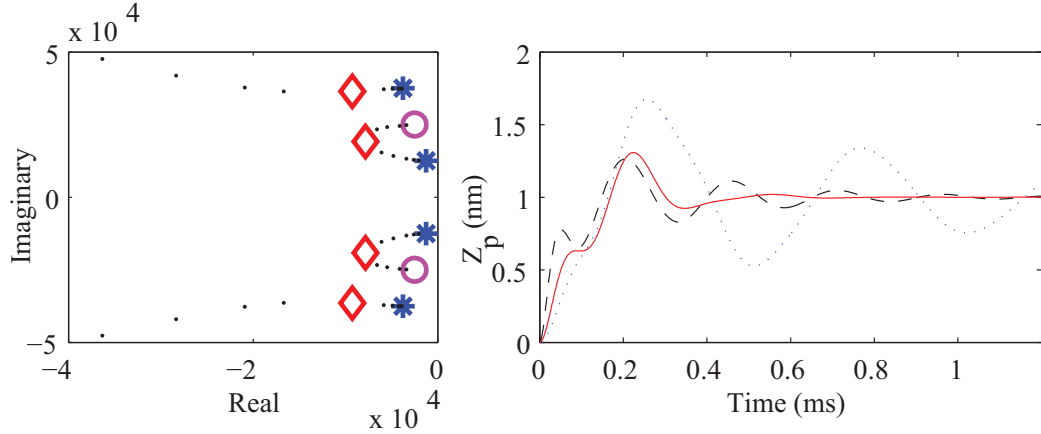


Figure 3.5: Limits of feedback. Left plot: (dots) variation of closed loop poles with input weight  $r$ ; (asterisks) open-loop poles; (diamonds) closed loop pole at  $r = 16$ ; and (circles) reflection of zeros about imaginary axis. Right plot: step responses, with steady state output normalized to one — (solid line)  $r = 16$ ; (dashed line) lower input cost with  $r = 1$ ; and (dotted line) open loop response.

#### *Reference DC-gain Feedforward*

Although the sample surface is not known a priori, current estimates of the sample height (e.g.,  $z_s$  from Eq. (3.10)) can be used to find feedforward to adjust the piezo position to reduce the force on the sample. In particular, the reference input  $u_{ref}$  in Eq. (3.19) is chosen to be reference piezo position  $z_{p,ref}$  in Eq. (3.16) scaled by the inverse of the DC-gain ( $-CA_{cl}^{-1}B$ ) of the closed-loop system (Eq. 3.20), i.e.,

$$\begin{aligned}
 u_{ref}(t) &= \frac{1}{(-CA_{cl}^{-1}B)} [z_{p,ref}(t)], \\
 &= \frac{1}{(-CA_{cl}^{-1}B)} \left[ z_s(t) - \left( 1 + \frac{K_c}{K_s} \right) z_{c,nom} \right]. \quad (3.22)
 \end{aligned}$$

This reference feedforward  $u_{ref}$  attempts to correct the steady state error, and in this sense, plays the role of an integral controller.

### Results Without Boundary Regulation

The tracking with feedback and DC-gain feedforward as described in Eqs. (3.20) and (3.22) are shown in Fig. 3.6, which presents the positioning results for scanning the example sample at 8 Hz over one scan period ( $T = 125$  ms). Even without the boundary-regulation

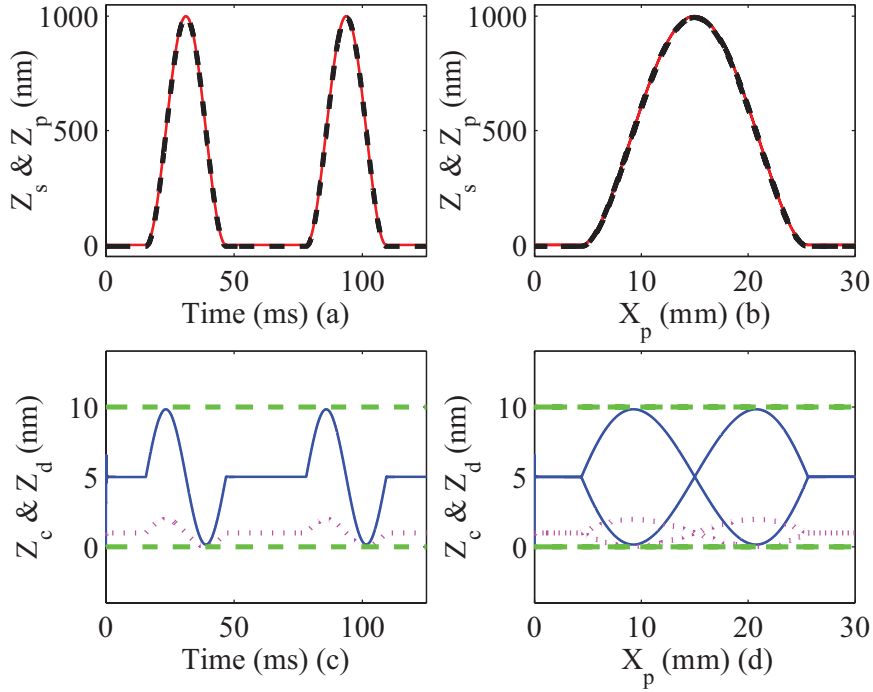


Figure 3.6: Results for one scan period at the scan frequency of 8 Hz without boundary regulation. Left plot: vertical positions versus time. Right plot: vertical positions versus scan line  $x_p$ . Top: (solid line) sample profile  $z_s$ ; (dashed line) vertical position of the piezo  $z_p$ . Bottom: (solid line) cantilever deflection  $z_c$ ; (dotted line) sample deformation  $z_d$ ; (dashed line) the region's boundary [0 10] nm.

approach, the applied sample force remains sufficiently small when the scan frequency is small, as seen in Fig. 3.6. However, the cantilever deflection  $z_c$  (shown in Fig. 3.6 (c) and (d)) is excessive, i.e.  $z_c > 10$  nm when the scan frequency  $f$  becomes larger, e.g.,  $f > 8$  Hz. It is noted that when the cantilever deflection  $z_c$  is negative, contact is lost with the sample, and the models used for simulations are not valid.

### 3.2.3 With Boundary Regulation

The boundary regulation approach is studied in this section. Two approaches to design the feedforward input to transition the output away from the boundary are investigated: (i) using step inputs; and (ii) using exact inversion of the system dynamics.

#### Transition Triggering

The cantilever deflection  $z_c$ , is monitored at discrete transition-condition check instants  $t_k = kt_{ff}$  spaced evenly at the transition-check interval  $t_{ff}$ . If the cantilever deflection at  $z_c(t_k)$  is outside the transition trigger values, i.e.,  $z_c(t_k) < \underline{z}_c$  or  $z_c(t_k) > \bar{z}_c$  as in Fig. 3.1, then an additional feedforward input  $u_{ff}$  is used to transition the cantilever deflection  $z_c$  back to the nominal value  $z_{c,nom}$ . The resulting control law in Eq. (3.19) becomes

$$u(t) = -K_{fb}X(t) + u_{ref}(t) + u_{ff}(t). \quad (3.23)$$

The nominal cantilever deflection  $z_{c,nom} = 5$  nm and the desired-cantilever-deflection region is  $[0 \ 10]$  nm (i.e.,  $[\underline{Z}_c \ \bar{Z}_c]$  in Fig. 3.1). In the following simulations, the transition trigger values  $\underline{z}_c$  and  $\bar{z}_c$  were chosen to be half the distance from  $z_{c,nom}$  to the boundary  $[0 \ 10]$  nm, i.e.,  $\underline{z}_c = 2.5$  nm and  $\bar{z}_c = 7.5$  nm.

If the cantilever deflection  $z_c(t_k)$  at a transition-condition check instants  $t_k$  is outside the transition-trigger region, i.e.,

$$z_c(t_k) < \underline{z}_c, \quad \text{or} \quad z_c(t_k) > \bar{z}_c$$

then the feedforward input  $u_{ff}$  is used to increase or reduce the cantilever deflection by the distance  $\Delta z_c = 2.5$  nm of the transition trigger values ( $\underline{z}_c, \bar{z}_c$ ) from the nominal value  $\bar{z}_{c,nom}$ . The corresponding change  $\Delta z_p$  needed in the piezo position  $z_p$  can be found as (similar to Eq. (3.16))

$$\Delta z_p = \left(1 + \frac{K_c}{K_s}\right) \Delta z_c \quad (3.24)$$

yields  $\Delta z_p = 3.0$  nm.

*Approach 1: Step Input as Feedforward*

In this approach, the additional displacement in the piezo position  $z_{p,d}$ , needed to maintain the cantilever deflection in the desired region, is potentially updated by  $\Delta z_p$  (defined in Eq. (3.24)) at every transition-condition check instants  $t_k$  as

$$\begin{aligned} z_{p,d}(t_k) &= z_{p,d}(t_k) - \Delta z_p && \text{if } z_c(t_k) < \underline{z}_c, \\ &= z_{p,d}(t_k) + \Delta z_p && \text{if } z_c(t_k) > \bar{z}_c, \\ &= z_{p,d}(t_{k-1}) && \text{otherwise} \end{aligned} \quad (3.25)$$

with an initial value  $z_{p,d}(0) = 0$ . Then, the additional feedforward input  $u_{ff}$ , to achieve this step change in the piezo position, is found using the DC-gain of the system as

$$u_{ff}(t) = \frac{[z_{p,d}(t_k)]}{(-CA_{cl}^{-1}B)} \quad \forall \quad t_k \leq t < t_{k+1}. \quad (3.26)$$

Note that the feedforward input changes in discrete steps with this approach, and should give a fast rise time depending on the closed-loop properties. For example, with the input  $u_{ff}$  chosen as one of these steps, i.e.,

$$u_{ff,\Delta} = \frac{[\Delta z_p]}{(-CA_{cl}^{-1}B)} \quad (3.27)$$

the resulting change in piezo position  $z_p$  (from an initial value of zero) is shown in Fig. 3.7 (left). Note that the settling time is 0.44 ms — the transition-check interval  $t_{ff}$  (for this step-input feedforward case) is chosen to be initial time 0.17 ms when the piezo position  $z_p$  reaches the desired final position  $\Delta z_p$ .

*Approach 2: Exact Inverse as Feedforward*

To increase the rate at which the output is transitioned away from the boundary, the steady-state feedforward input in Eq. (3.26) was augmented with an inversion-based feedforward input  $u_{tran}$  to speed up the initial transition, i.e.,

$$\begin{aligned} u_{ff}(t) &= u_{tran}(t) + \frac{[z_{p,d}(t_{k-1})]}{(-CA_{cl}^{-1}B)} && \text{for } t_k \leq t < [t_k + t_{tran}], \\ u_{ff}(t) &= \frac{[z_{p,d}(t_k)]}{(-CA_{cl}^{-1}B)} && \text{for } [t_k + t_{tran}] \leq t < t_{k+1}. \end{aligned} \quad (3.28)$$

Conceptually, the additional feedforward input  $u_{tran}$  can be chosen to be the exact inverse input  $u_{inv}$  to achieve as fast a transition as needed by tracking a sufficiently smooth

fast trajectory — within actuator saturation limits. For example, the vertical piezo position  $z_p$  could be driven from 0 *nm* to  $\Delta z_p = 3$  *nm* (in Eq. (3.24)) in say a transition time  $t_{tran}$  of

$$t_{tran} = 0.02 \text{ ms},$$

which is about one tenth of the transition-check interval  $t_{ff} = 0.17$  ms with the step input in approach 1. The corresponding piezo trajectory is found by smoothing a ramp signal (represented by a dash-dotted line in Fig. 3.7 (right, bottom)) twice by a first-order filter with a break frequency of 175 *kHz* to obtain the twice differentiable desired trajectory  $z_{p,d}$  for the piezo position. Since the piezo dynamics has the relative degree two, the expression for the second time derivative of the output position  $z_p$  can be written as (from Eq. (3.7))

$$\begin{aligned} \ddot{z}_p(t) &= (CA^2)X(t) + (CAB)u_{inv}(t) \\ &= A_y X(t) + B_y u_{inv}(t). \end{aligned} \quad (3.29)$$

Matching this expression to the second time derivative of the desired piezo position, i.e.,  $\ddot{z}_p = \ddot{z}_{p,d}$  leads to the inverse system (and the inverse input)

$$\begin{aligned} \dot{\eta}(t) &= [A_{inv}]\eta(t) + [B_{inv}]Y_d(t) \\ u_{inv}(t) &= [C_{inv}]\eta(t) + [D_{inv}]Y_d(t) \end{aligned} \quad (3.30)$$

where  $\eta$  is the internal dynamics and

$$A_{inv} = T_\eta[A - (BB_y^{-1}A_y)]T_r^{-1}, \quad (3.31)$$

$$B_{inv} = [T_\eta[A - (BB_y^{-1}A_y)]T_l^{-1} \vdots T_\eta(BB_y^{-1})],$$

$$C_{inv} = -B_y^{-1}A_yT_r^{-1},$$

$$D_{inv} = [-B_y^{-1}A_yT_l^{-1} \vdots (B_y^{-1})],$$

$$Y_d(t) = \begin{bmatrix} z_{p,d}(t) \\ \dot{z}_{p,d}(t) \\ \ddot{z}_{p,d}(t) \end{bmatrix} \quad (3.32)$$

and the transformation matrix  $T$  is given by

$$T = \begin{bmatrix} C \\ CA_{cl} \\ T_\eta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ -2.5133 & 0.001 & -7.5398 & 0.001 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.33)$$

$$T^{-1} = \left[ T_r^{-1} \mid T_l^{-1} \right] = \left[ \begin{array}{cc|cc} 1.5000 & 0.1989 & -0.0002 & -0.0002 \\ 0 & 0 & 1 & 0 \\ -0.5000 & -0.1989 & 0.0002 & 0.0002 \\ 0 & 0 & 0 & 1 \end{array} \right]. \quad (3.34)$$

**Remark 6.** (*Unstable Internal Dynamics*): The eigenvalues of  $A_{inv}$  in Eq. (3.31), i.e., the poles of the inverse system are also the zeros of the original system, and have a positive real part. Therefore the inverse system is unstable.

Bounded inverses can be found to the unstable inverse system [1]. However, this requires a pre-actuation time  $t_{pre}$  that depends on the distance  $d_{zeros}$  of the right-half-plane *unstable* zeros of the original system from the imaginary axis, and can be estimated as [73]

$$t_{pre} = \frac{10}{d_{zeros}} = \frac{10}{\zeta_z \omega_z} = \frac{10}{(0.1)(2\pi 4000)} = 3.98 \text{ ms} \quad (3.35)$$

where  $d_{zeros} = \zeta_z \omega_z$  from Eq. (3.6). The estimated 3.98 ms pre-actuation time needed with the causal inverse is almost an order of magnitude larger than the settling time of 0.44 ms with the step response. While it yields precision tracking of a fast-transition trajectory, the large pre-actuation time makes the approach not suitable to track rapidly changing external disturbances, e.g., scanning rapidly over a sample slope.

To avoid the need for pre-actuation time, direct causal inverse is proposed here to exactly track the desired output and achieve rapid output-transition from the boundary. A problem with this approach is that the associated internal dynamics is unstable, and can increase rapidly over time leading to rapidly increasing inverse input magnitudes. Nevertheless, since the inverse input is only used for a short period of time  $t_{tran} = 0.02$  ms, the inverse input is still bounded. An example of the inverse input  $u_{inv}$  is shown in Fig. 3.7 (right,

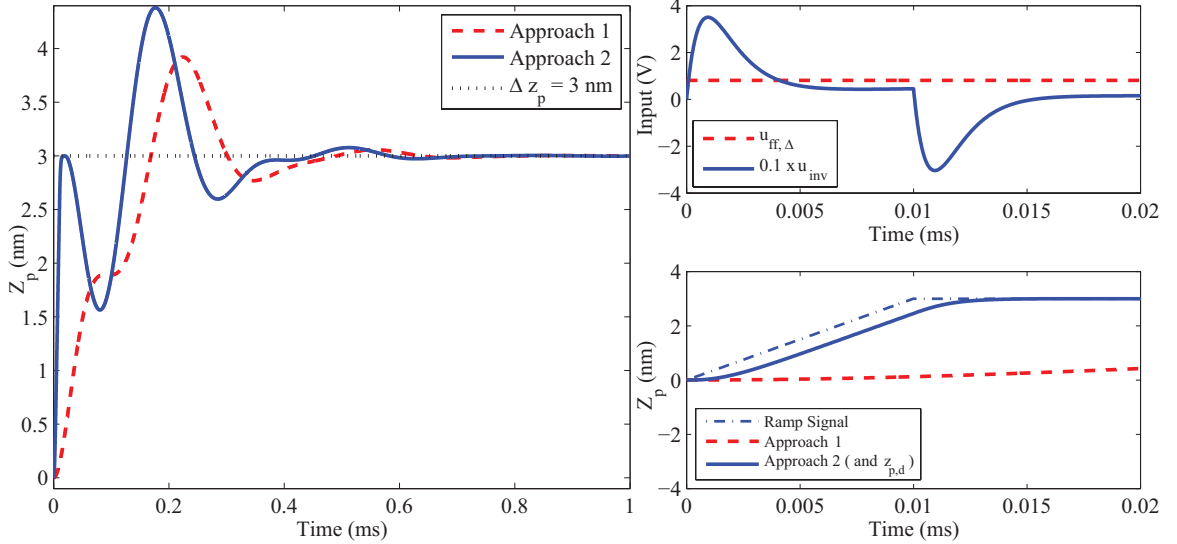


Figure 3.7: Piezo transition results. Left: output  $z_p$  for 1 ms: (dashed line) approach 1 (step input); (solid line) approach 2 (exact inverse). Right: input and output during the transition time of 0.02 ms. Top right: inputs: (dashed line)  $u_{ff,\Delta}$  in Eq. (3.27); (solid line)  $0.1 \times u_{inv}$  in Eq. (3.29). Bottom right: output  $z_p$ : (dash-dotted line) ramp signal; (dashed line) output  $z_p$  with  $u_{ff,\Delta}$ ; (solid line) desired piezo position  $z_{p,d}$  and output  $z_p$  with  $u_{inv}$ .

top). During the transition time  $t_{tran}$  exact tracking is achieved as shown in Fig. 3.7 (right, bottom), which also shows that the transition achieved is much faster than that achieved with the step input. However, after the transition time  $t_{tran}$ , the feedforward input becomes the steady state input (see Eq. (3.28)) needed to maintain the piezo position at the desired value  $z_{p,d}(t_k)$ . This will lead to a loss in tracking as shown in Fig. 3.7 (left), but the ensuing vibrations are away from the boundary and within the acceptable region. For the inversion-based boundary transition approach, the transition input  $u_{tran}$  (in Eq. 3.28) is chosen as, for the time interval  $t_k \leq t < t_{k+1}$ ,

$$\begin{aligned}
 u_{tran}(t) &= -u_{inv}(t - t_k) && \text{if } z_c(t_k) < \underline{z}_c, \\
 &= u_{inv}(t - t_k) && \text{if } z_c(t_k) > \bar{z}_c, \\
 &= 0 && \text{otherwise.}
 \end{aligned} \tag{3.36}$$

In the following simulations, the transition-check interval  $t_{ff}$  is chosen as the transition time  $t_{tran}$ , i.e.,  $t_{ff} = t_{tran} = 0.02$  ms.

### *Results with Boundary Regulation*

Comparative evaluation of boundary regulation is presented, below, for the two approaches: (i) using step inputs; and (ii) using exact inversion of the system dynamics. Example plots when scanning the sample surface (in Fig. 3.4) are shown in Fig. 3.8 (left plots) for approach 1 at a scan frequency of 31 Hz, and in Fig. 3.8 (right plots) for approach 2 at a scan frequency of 56 Hz. As already seen in Fig. 3.7, the output  $z_p$  with exact-inversion based feedforward (approach 2) can be rapidly transitioned away from the boundary, which leads to the ability to maintain the cantilever deflection  $z_c$  inside the desired region during high scan frequencies. To illustrate the use of boundary regulation with the exact inverse, example transitions from Fig. 3.8 are shown in Fig. 3.9.

#### *3.2.4 Discussion of Results*

The maximum and minimum cantilever deflections at each scan frequency for the case without boundary regulation (feedback and reference DC gain feedforward) are shown in Figure 3.10. The same figure also shows the maximum and minimum cantilever deflections with the boundary regulation for two cases: approach 1 with step input as feedforward; and approach 2 with exact inverse as feedforward. With the use of boundary regulation, the cantilever deflection  $z_c$  can be maintained to be inside the desired region [0 10] nm, which implies small sample surface forces can be maintained during high frequency scanning. As seen in Fig. 3.10, the maximum scan frequency can be increased from 8 Hz without boundary regulation (Fig. 3.10 (top)) to 31 Hz (Fig. 3.10 (middle)) by using the step input as feedforward with boundary regulation and further increased to 56 Hz (Fig. 3.10 (bottom)) by using the exact inverse as feedforward during boundary regulation. Thus, the simulation results show that the output-boundary regulation approach can be used to substantially increase the maximum scan frequency, from 8 Hz to 56 Hz, when compared to the case without the boundary-regulation approach.

Output-boundary regulation for high-speed AFM imaging of soft samples is discussed in this chapter. The proposed feedforward-based control technique overcomes both: (i) lack of preview information of the disturbances; and (ii) performance limitations of feedback-based

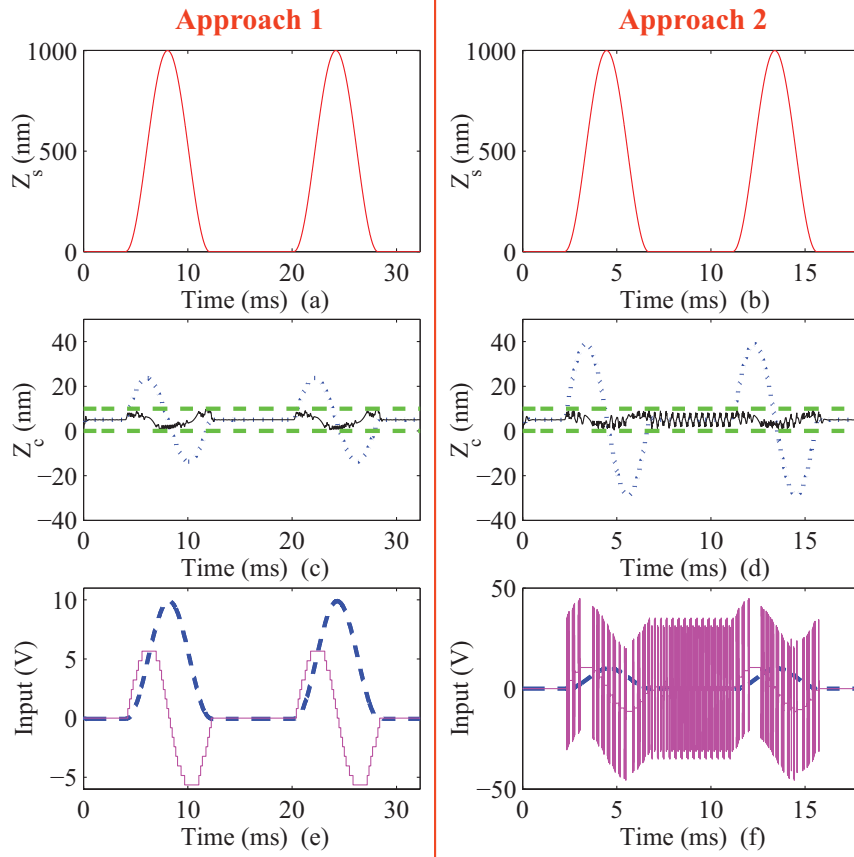


Figure 3.8: Results. Left: approach 1 (step input) at 31 Hz with the transition-check interval  $t_{ff} = 0.17$  ms. Right: approach 2 (exact inverse) at 56 Hz with the transition-check interval  $t_{ff} =$  the transition time  $t_{tran} = 0.02$  ms. Top: sample profile  $z_s$ . Middle: cantilever deflection  $z_c$ : (solid line) cantilever deflection with boundary regulation; (dotted line) cantilever deflection without boundary regulation; (dashed line) the region's boundary  $[0 \ 10]$  nm. Bottom: Inputs: (solid line) additional feedforward input  $u_{ff}$ ; (dashed line) feedback and reference DC gain feedforward input  $0.1u$ , with  $u$  from Eq. (3.19).

control methods for nonminimum phase systems. Simulation results for an example AFM show that the output-boundary regulation approach can be used to substantially increase the maximum scan frequency.

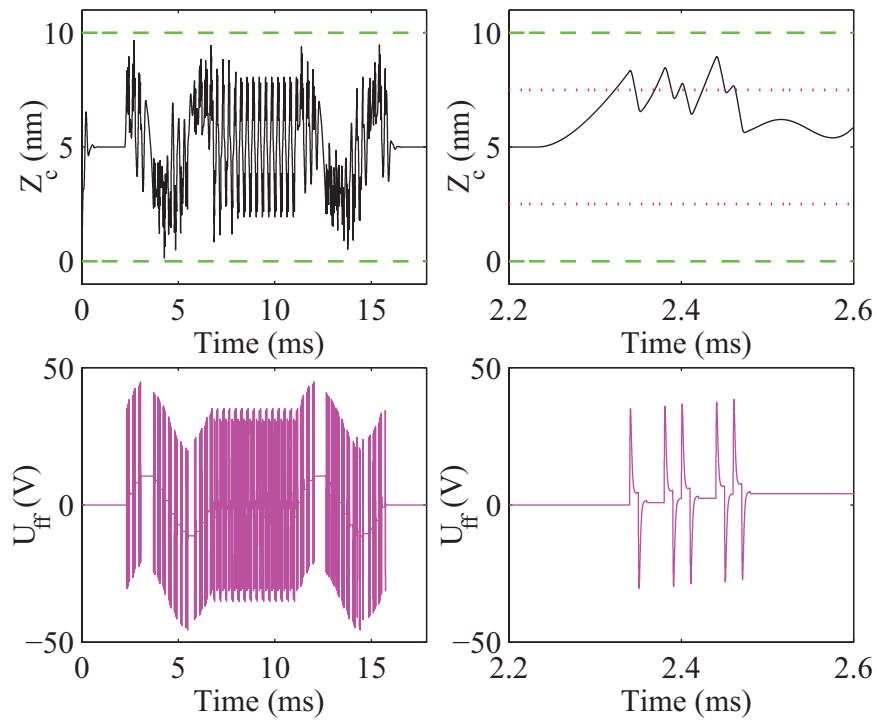


Figure 3.9: Magnification of Fig. 3.8 approach 2 (exact inverse). Left: from Fig. 3.8(d) and (f). Right: magnification of the left plots between 2.2 ms and 2.6 ms. Top: cantilever deflection: (solid line) cantilever deflection  $z_c$ ; (dashed line) region's boundary [0 10] nm; (dotted line) transition trigger values [2.5 7.5] nm. Bottom: additional feedforward input  $u_{ff}$ .

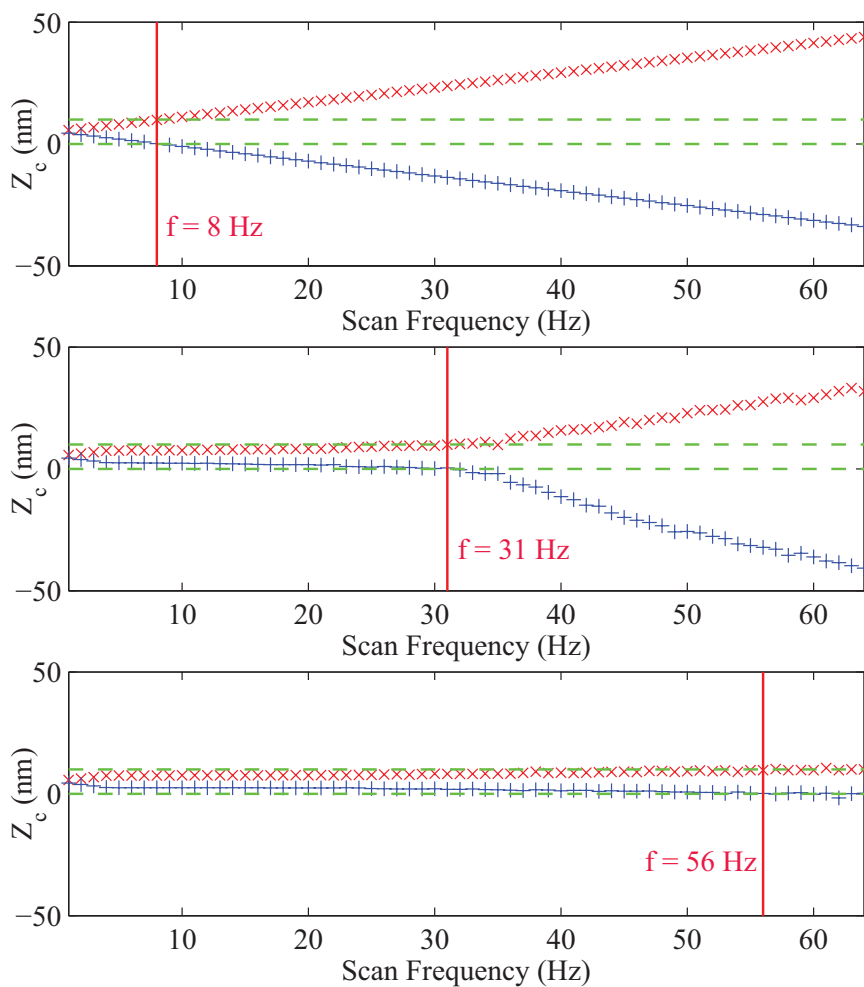


Figure 3.10: Maximum and minimum cantilever deflection  $z_c$  at different scan frequencies: (cross) maximum  $z_c$ ; (plus sign) minimum  $z_c$ ; (dashed line) the region's boundary. Top: without boundary regulation. Middle: with boundary regulation, approach 1 – step input (the transition-check interval  $t_{ff} = 0.17$  ms). Bottom: with boundary regulation, approach 2 – exact inverse (the transition-check interval  $t_{ff} = 0.02$  ms).

## Chapter 4

## OUTPUT-BOUNDARY REGULATION USING EVENT-BASED FEEDFORWARD FOR LINEAR SYSTEMS

This chapter<sup>1</sup> presents the proposed event-based feedforward which expands on the concept of event-based feedback proposed, e.g., in [74, 75], by allowing more general input (rather than, say, an input proportional to the state) at each event trigger. The proposed output-boundary regulation approach is to rapidly transition the output back to the nominal value when the output approaches a specified output-boundary to maintain the output inside the given output-boundary as shown in Fig. 4.1. While the use of feedforward to control

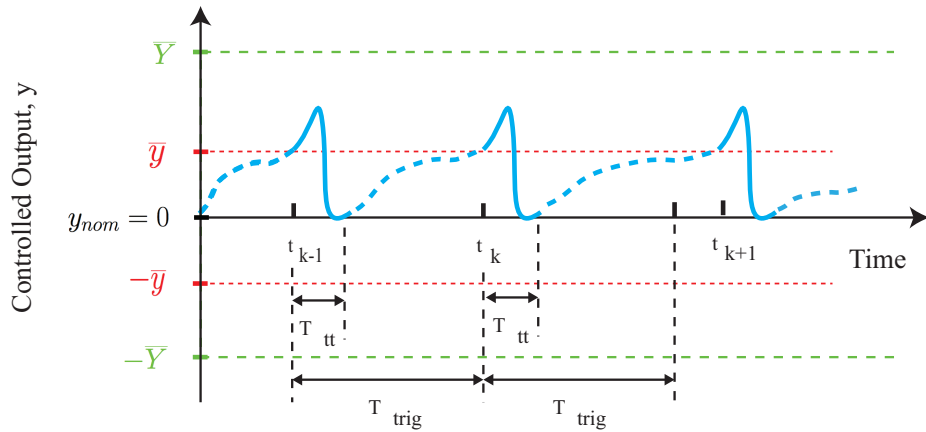


Figure 4.1: Schematic of output-boundary regulation. The output  $y$  is maintained within given bounds  $[-\bar{Y}, \bar{Y}]$ . If the output-error is outside a transition-trigger region  $(-\bar{y}, \bar{y}) \subset (-\bar{Y}, \bar{Y})$ , then the proposed OBR approach triggers a transition of the output  $y$  back towards the nominal value  $y_{nom} = 0$ .

a single output transition is well understood, several research problems arise when using multiple such transitions with the proposed boundary regulation approach. For example,

---

<sup>1</sup>Material in this chapter was presented in 2015 American Control Conference (ACC) [12] and was published in IEEE Transactions on Control Systems Technology [13].

multiple transitions can lead to potential loss of performance (boundary regulation) and loss of input boundedness. These theoretical problems, of ensuring performance and input boundedness with event-based feedforward, are addressed here.

#### 4.1 Problem Formulation and Solution Approach

##### 4.1.1 Output-Boundary-Regulation (OBR) Problem

Let  $G$  be a linear time-invariant nonminimum-phase plant described by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ z(t) &= Cx(t)\end{aligned}\tag{4.1}$$

where  $A$  is Hurwitz with eigenvalues on the open-left-half of the complex plane (i.e.,  $G$  is stable), and a disturbance  $d$  that adds to the scalar output  $z$  of the plant  $G$  as shown in Fig. 2.4 to yield the output  $y$

$$y(t) = z(t) + d(t).\tag{4.2}$$

Then, the OBR problem is to find a bounded input  $u$  that maintains the output  $y$  within pre-specified bounds

$$y(t) \in [\underline{Y}, \bar{Y}] = [-\bar{Y}, \bar{Y}], \quad \bar{Y} > 0.\tag{4.3}$$

Although the lower bound  $\underline{Y}$  and upper bound  $\bar{Y}$  are kept the same, in the following, for simplicity, they can be chosen to be different, without substantial change in the following arguments.

##### 4.1.2 OBR Approach

If the output  $y$  nears the output-error bounds  $[-\bar{Y}, \bar{Y}]$ , and is outside a transition-trigger region

$$(-\bar{y}, \bar{y}) \subset (-\bar{Y}, \bar{Y}),\tag{4.4}$$

then the proposed OBR approach triggers a transition of the output  $y$  back towards the nominal value  $y_{nom} = 0$ . This is achieved using a precomputed feedforward input  $u_{ff}$  that transitions the output by a transition-trigger bound of  $\bar{y}$  (or  $-\bar{y}$  by using  $-u_{ff}$ ) within a

transition time  $T_{tt}$ . In particular, a transition is triggered at time

$$t_k \geq t_{k-1} + T_{trig}, \quad k \geq 1 \quad (4.5)$$

if

$$|y(t_k)| \geq \bar{y}, \quad (4.6)$$

where the initial time is defined as  $t_0 = -T_{trig}$  and the minimum trigger time  $T_{trig}$  between transitions is greater than the transition time  $T_{tt}$ , i.e.,

$$T_{trig} \geq T_{tt}. \quad (4.7)$$

Thus, the output transitions are enabled at discrete time instants (trigger times)  $t_k > 0$  ( $k = 1, 2, 3, \dots$ ), which are separated by at-least the minimum trigger time  $T_{trig}$ .

**Remark 7.** *Although the emphasis of this article is on nonminimum-phase systems, the proposed OBR approach is applicable to minimum-phase systems as well.*

#### 4.1.3 Main Issues

The main issues in the proposed OBR approach (addressed in the following sections) are to

1. find the output-transition control to rapidly move the output  $y$  away from the boundary,
2. quantify performance in terms of a relationship between the acceptable output-error bound  $[-\bar{Y}, \bar{Y}]$  and the type of disturbance  $d$  that can be handled by the OBR approach, and
3. ensure that the OBR input  $u$  is bounded when the disturbance  $d$  is bounded.

## 4.2 Output-Transition Control

This section begins by developing the input law for a single output transition. The output  $y$  can be changed by changing the plant output  $z$ . Therefore, the input law for a single plant-output transition is studied first. Multiple output-transitions are then composed of time-delayed versions of these single output-transitions to develop the OBR solution.

#### 4.2.1 Single Plant-Output Transition

Consider the general problem of transitioning the plant output  $z$  from (a steady-state value) 0 to the final (steady state) value of  $\bar{y}$ , i.e.,

$$z(0) = 0, \quad z(T_{tt}) = \bar{z} = \bar{y} \quad (4.8)$$

during the transition time  $T_{tt}$ . The plant output should be constant before and after the transition as shown in Fig. 4.2, i.e.,

$$z(t) = 0 \quad \forall t \leq 0, \quad z(t) = \bar{z} \quad \forall t \geq T_{tt}. \quad (4.9)$$

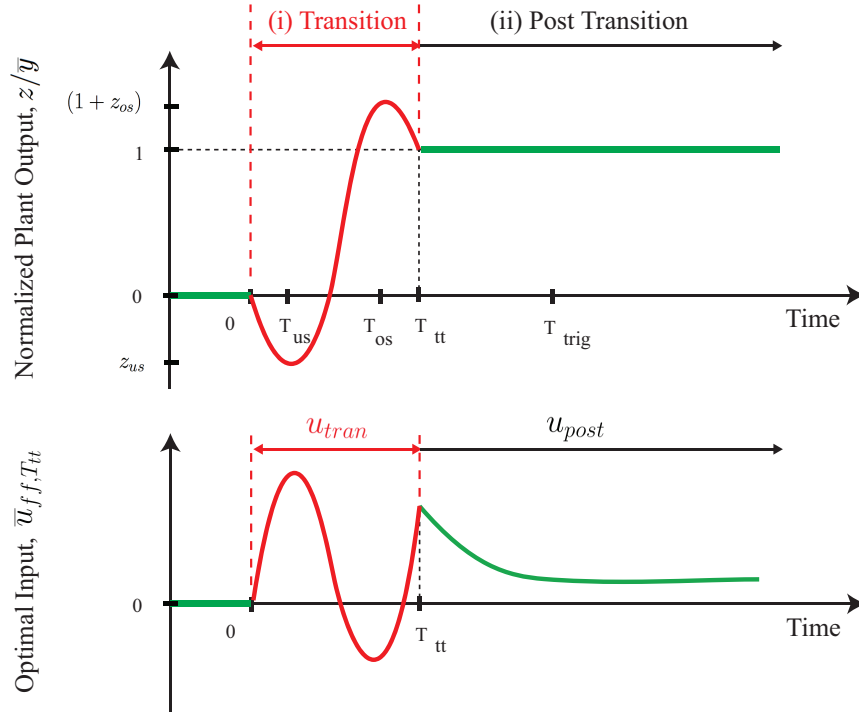


Figure 4.2: Normalized optimal single plant output transition and the optimal input. The optimal plant output  $z = \bar{z}_{ff}$  is (i) transitioned from (steady-state value) 0 to the nominal (steady state) value  $\bar{z} = \bar{y}$  during the transition time  $T_{tt}$  and (ii) maintained constant afterward. The normalized undershoot  $z_{us}$  and overshoot  $z_{os}$  are shown to occur at time instants  $t = T_{us}$  and  $t = T_{os}$ .

**Remark 8** (Non-zero steady-state post-transition input). *In general the input can be non-zero outside of the time interval  $t \in (0, T_{tt})$  over which the plant output  $z$  is transitioned. In this article, we assume that preview information about the desired transition is not available, and hence pre-transition input (before the transition, for time  $t \leq 0$ ) cannot be used. However, post-transition input (after the transition, for time  $t \geq T_{tt}$ ) is possible. This post-transition input tends to a non-zero steady-state value after the transition since the final plant output  $\bar{z}$ , at the end of the transition, is not necessarily an equilibrium point of the system in Eq. (4.1) as in standard OOT approaches, e.g., as in [44].*

There are many input-and-plant-output pairs that achieve the single plant-output  $z$  transition specified in Eqs. (4.8) and (4.9). However, the non-zero steady-state value for the input (after the transition) implies that standard cost functions, e.g., that minimize the input energy to optimally select an input-and-plant-output pair as in [44], cannot be used because the post-transition energy-cost would be infinite. In the following, we use a cost function that optimizes the plant output, or more specifically, the time-derivative of the plant output  $\frac{d^r}{dt^r}z(t)$  where  $r$  is the relative degree of the system to select the plant-output trajectory  $z$  to achieve the desired transition. This approach ensures that the resulting plant-output trajectory  $z$  is invertible and an exact-tracking input  $u$  can be found to achieve the transition.

Note that there are two issues in a single transition: (i) transitioning the plant output by  $\bar{z}$  in the transition time interval i.e., time  $t \in \mathbb{I} = [0, T_{tt}]$ ; and (ii) maintaining the state output constant afterwards during post transition for time  $t > T_{tt}$ . The second problem is discussed first, since it informs on the type of state constraints needed during the transition to ensure that (a) pre-actuation is not needed, and (b) the plant output can be maintained constant during post actuation with bounded inputs as in the standard OOT approach, e.g., see [44].

### Inverse Input

A relationship between the plant output  $z$  and the input  $u$  can be found using the  $r^{\text{th}}$  time derivative of the plant output as

$$\begin{aligned}\frac{d^r}{dt^r}z(t) &= (CA^r)x(t) + (CA^{r-1}B)u(t) \\ &= A_yx(t) + B_yu(t)\end{aligned}\tag{4.10}$$

where  $r$  is the relative degree of the system, which corresponds to the difference between the number of poles and zeros of a linear time-invariant system, and  $B_y \neq 0$ . The inverse input  $u_{inv}$  needed to track any sufficiently-smooth plant output trajectory  $z$  (including the specific case of constant plant output where the time derivatives are zero) can be found from Eq. (4.10) as

$$u_{inv}(t) = B_y^{-1} \left( \frac{d^r}{dt^r}z(t) - A_yx(t) \right).\tag{4.11}$$

The system state  $x$  can be decomposed into known components  $\xi$  (in terms of the desired plant output  $z$  and its time derivatives) and unknown components  $\eta$ , i.e.,

$$\begin{bmatrix} \xi(t) \\ \eta(t) \end{bmatrix} = \begin{bmatrix} T_\xi \\ T_\eta \end{bmatrix} x(t) = Tx(t)\tag{4.12}$$

where

$$\xi(t) = \begin{bmatrix} z(t) \\ \dot{z}(t) \\ \vdots \\ \frac{d^{(r-1)}}{dt^{(r-1)}}z(t) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} x(t),\tag{4.13}$$

the matrix  $T_\eta$  is chosen such that the transformation matrix  $T$  is invertible resulting in the following inverse transformation

$$x = T^{-1} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \Phi \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \Phi_\xi \xi + \Phi_\eta \eta.\tag{4.14}$$

The expression for the inverse input in Eq. (4.11) can be rewritten as the output of the following inverse system, by differentiating  $\eta$  in Eq. (4.12) and using the system description in Eq. (4.1) for the time derivative of state  $x$  to obtain

$$\dot{\eta}(t) = T_\eta \dot{x}(t) = T_\eta (Ax(t) + Bu(t)).\tag{4.15}$$

Then, replacing the state  $x$  using Eq. (4.14) results in

$$\dot{\eta}(t) = A_{inv}\eta(t) + B_{inv}\mathcal{Z}(t) \quad (4.16)$$

$$u_{inv}(t) = C_{inv}\eta(t) + D_{inv}\mathcal{Z}(t) \quad (4.17)$$

where  $\eta$  is referred to as the internal state,

$$\begin{aligned} A_{inv} &= T_\eta (A - BB_y^{-1}A_y) \Phi_\eta \\ B_{inv} &= T_\eta \left[ (A - BB_y^{-1}A_y) \Phi_\xi \ ; \ BB_y^{-1} \right] \\ C_{inv} &= -B_y^{-1}A_y\Phi_\eta \\ D_{inv} &= [-B_y^{-1}A_y\Phi_\xi \ ; \ B_y^{-1}] \\ \mathcal{Z} &= \begin{bmatrix} \xi \\ \frac{d^r}{dt^r}z \end{bmatrix}. \end{aligned} \quad (4.18)$$

For hyperbolic systems (without zeros on the imaginary axis of the complex plane), it can be assumed, without loss of generality, that the matrix  $T_\eta$  is chosen such that the internal dynamics in Eq. (4.16) is decoupled into stable  $\eta_s$  and unstable  $\eta_u$  sub-systems, i.e., [44]

$$\dot{\eta}(t) = \begin{bmatrix} A_s & 0 \\ 0 & A_u \end{bmatrix} \eta(t) + \begin{bmatrix} B_{s,\xi} & B_s \\ B_{u,\xi} & B_u \end{bmatrix} \mathcal{Z}(t) \quad (4.19)$$

with

$$\eta = \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix}, \quad (4.20)$$

where  $A_s$  and  $-A_u$  are Hurwitz matrices whose eigenvalues are on the open left-half of the complex plane.

### *Post Transition*

When the plant output is constant  $z(t) = \bar{z}$  for  $t > T_{tt}$ , its derivatives are zero and the known part of the state  $\xi$  in Eq. (4.13) becomes

$$\xi(t) = \bar{\xi} = \begin{bmatrix} \bar{z} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \forall t > T_{tt} \quad (4.21)$$

and the input  $\mathcal{Z}$  in Eq. (4.19) to the internal dynamics becomes

$$\mathcal{Z}(t) = \bar{\mathcal{Z}} = \begin{bmatrix} \bar{\xi} \\ 0 \end{bmatrix}, \quad \forall t > T_{tt}. \quad (4.22)$$

Note that the internal dynamics in Eq. (4.19) becomes autonomous, and the unstable component  $\eta_u$  of the internal dynamics (for  $t > T_{tt}$ ) is given by

$$\eta_u(t) = \bar{\eta}_u + e^{A_u(t-T_{tt})} (\eta_u(T_{tt}) - \bar{\eta}_u) \quad (4.23)$$

for time  $t > T_{tt}$  with a potential steady-state (equilibrium) value

$$\bar{\eta}_u = -A_u^{-1} \begin{bmatrix} B_{u,\xi} & B_u \end{bmatrix} \bar{\mathcal{Z}}. \quad (4.24)$$

To prevent the unstable component  $\eta_u$  of the internal dynamics from increasing without bound, it is required to be at the steady-state value  $\bar{\eta}_u$  at time  $t = T_{tt}$ , i.e.,

$$\eta_u(T_{tt}) = \bar{\eta}_u. \quad (4.25)$$

On the other hand, irrespective of the value of the stable component  $\eta_s$  at the end of the transition at time  $t = T_{tt}$ , it reaches the steady state with a solution given by

$$\eta_s(t) = \bar{\eta}_s + e^{A_s(t-T_{tt})} (\eta_s(T_{tt}) - \bar{\eta}_s) \quad (4.26)$$

for time  $t > T_{tt}$  with the steady-state value

$$\bar{\eta}_s = -A_s^{-1} \begin{bmatrix} B_{s,\xi} & B_s \end{bmatrix} \bar{\mathcal{Z}}. \quad (4.27)$$

Then, the post-transition input is given, from Eqs. (4.17), (4.22), (4.25), (4.26), as (for  $t > T_{tt}$ )

$$u_{post}(t) = C_{inv} \begin{bmatrix} \eta_s(t) \\ \bar{\eta}_u \end{bmatrix} + D_{inv} \bar{\mathcal{Z}}. \quad (4.28)$$

#### *During The Transition Interval*

As discussed before, there is no constraint on the stable, internal-state component  $\eta_s(T_{tt})$  at the end of the transition interval  $\mathbb{I}$ , i.e., at time  $t = T_{tt}$ . However, the plant output  $z$

should be chosen such that, both, (i) the plant output and its derivatives  $\mathcal{Z}$  as well as (ii) the unstable component of the internal dynamics  $\eta_u$  transition to their steady state values during the transition interval  $\mathbb{I}$ . The dynamics of the associated reduced-dimensional state  $\tilde{x}$

$$\tilde{x}(t) = \begin{bmatrix} \mathcal{Z}(t) \\ \eta_u(t) \end{bmatrix} \quad (4.29)$$

that need to be transitioned, can be found by differentiating Eq. (4.13) and using Eq. (4.19) as

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}\tilde{u}(t) \quad (4.30)$$

where  $\mathcal{Z}$  is defined in Eq. (4.18),

$$\tilde{A} = \begin{bmatrix} A_\xi & B_\xi & 0 \\ 0 & A_f & 0 \\ B_{u,\xi} & B_u & A_u \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ B_f \\ 0 \end{bmatrix}, \quad (4.31)$$

$$A_\xi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B_\xi = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (4.32)$$

and the input  $\tilde{u}$  to the system is a pre-filtered version of the time-derivative of the plant output  $\frac{d^r}{dt^r}z(t)$ , i.e.,

$$\begin{aligned} \dot{\hat{u}}(t) &= A_f\hat{u}(t) + B_f\tilde{u}(t) \\ \frac{d^r}{dt^r}z(t) &= \hat{u}(t). \end{aligned} \quad (4.33)$$

The reduced-dimensional state  $\tilde{x}$  should transition from

$$\tilde{x}(0) = \begin{bmatrix} \mathcal{Z}(0) \\ \eta_u(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.34)$$

to, from Eqs. (4.21), (4.22), (4.24), (4.25),

$$\begin{aligned}
\tilde{x}(T_{tt}) &= \begin{bmatrix} \mathcal{Z}(T_{tt}) \\ \eta_u(T_{tt}) \end{bmatrix} \\
&= \begin{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\ -A_u^{-1} \begin{bmatrix} B_{u,\xi} & B_u \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \bar{z} \\
&= \Psi \bar{z}.
\end{aligned} \tag{4.35}$$

### *Optimal Output-Transition Trajectory*

For a given transition time  $T_{tt}$ , a standard (minimum-energy) state transition (SST) problem is solved to select the plant output  $z$  during the transition interval  $\mathbb{I}$ , by minimizing the cost functional  $J$  given by

$$\min_{\tilde{u}} \left[ J_{T_{tt}} = \int_0^{T_{tt}} [\tilde{u}(\tau)]^2 d\tau \right] \tag{4.36}$$

subject to the state boundary constraints in Eqs. (4.34) and (4.35), where the subscript  $T_{tt}$  is used to emphasize the dependence on the specific choice of the final time  $T_{tt}$ .

**Remark 9** (Frequency Weighting). *The pre-filter in Eq. (4.33) allows frequency weighting in the cost functional  $J_{T_{tt}}$  as discussed in [76]. The pre-filter can be designed to reduce the high frequency components in the input  $\tilde{u}$ , and therefore, in the plant output  $z$  during the transition. It also ensures continuity of the plant-output's time derivative  $\frac{dr}{dt^r} z(T_{tt})$ .*

**Remark 10.** *Other approaches can be used to select the plant output  $z$  during the transition interval  $\mathbb{I}$ , e.g., by optimally selecting from a prespecified set of polynomial trajectories, e.g., as studied in [43] or choosing other forms of the cost functional  $J_{T_{tt}}$ .*

The optimal, minimal-energy input  $\tilde{u}_{T_{tt}}$  is given by

$$\tilde{u}_{T_{tt}}(t) = \tilde{B}' \left[ e^{\tilde{A}'(T_{tt}-t)} \right] \left[ G_{tran}(T_{tt}) \right]^{-1} \tilde{d}(T_{tt}) \tag{4.37}$$

where  $'$  indicates transpose, the state difference  $\tilde{d}(T_{tt})$  is, from Eqs. (4.34) and (4.35),

$$\tilde{d}(T_{tt}) = \tilde{x}(T_{tt}) - \left[ e^{\tilde{A}T_{tt}} \right] \tilde{x}(0) = \Psi \bar{z}. \tag{4.38}$$

The invertible controllability gramian is

$$G_{tran}(T_{tt}) = \int_0^{T_{tt}} e^{\tilde{A}(T_{tt}-\tau)} \tilde{B} \tilde{B}' e^{\tilde{A}'(T_{tt}-\tau)} d\tau. \quad (4.39)$$

### *Transition Input*

Corresponding to the optimal input  $\tilde{u}_{T_{tt}}$ , the optimal reduced state  $\tilde{x}_{T_{tt}}$  can be found by solving Eq. (4.30) using the zero initial condition from Eq. (4.34) — the corresponding plant output and its time derivatives are represented by  $\mathcal{Z}_{T_{tt}}$  and the unstable internal dynamics by  $\eta_{u,T_{tt}}$ . The resulting stable internal dynamics can be found from Eq. (4.19) as

$$\eta_{s,T_{tt}}(t) = \int_0^t e^{A_s(t-\tau)} [B_{s,\xi} \ B_s] \mathcal{Z}_{T_{tt}}(\tau) d\tau \quad (4.40)$$

for all time  $t$  in the transition interval  $\mathbb{I}$ , where the initial condition is  $\eta_s(0) = 0$  corresponding to the initial equilibrium state  $x(0) = 0$ . The input  $u$  during the transition interval  $\mathbb{I}$  can then be found from Eq. (4.17) as

$$u_{tran}(t) = C_{inv} \begin{bmatrix} \eta_{s,T_{tt}}(t) \\ \eta_{u,T_{tt}}(t) \end{bmatrix} + D_{inv} \mathcal{Z}_{T_{tt}}(t). \quad (4.41)$$

### *Optimal Input and Output*

The time-energy optimal input  $\bar{u}_{ff,T_{tt}}$  for single plant-output transition from  $z = 0$  to  $z = \bar{z} = \bar{y}$  is given by Eqs. (4.28) and (4.41) as

$$\begin{aligned} \bar{u}_{ff,T_{tt}}(t) &= u_{tran}(t) \text{ for } 0 \leq t \leq T_{tt} \\ &= u_{post}(t) \text{ for } t > T_{tt}. \end{aligned} \quad (4.42)$$

The plant output  $z$  corresponding to applying the optimal input  $\bar{u}_{ff,T_{tt}}$  in Eq. (4.42) to the system in Eq. (4.1) with zero initial conditions  $x(0) = 0$  is referred to as the optimal plant output  $\bar{z}_{ff,T_{tt}}$ . The corresponding output  $y$  without disturbance  $d$  is referred to as the optimal output

$$\bar{y}_{ff,T_{tt}}(t) = \bar{z}_{ff,T_{tt}}(t) \quad \forall t > 0. \quad (4.43)$$

Note that the optimal output  $\bar{y}_{ff,T_{tt}}$  reaches a constant after the transition time  $T_{tt}$ , i.e.,  $\bar{y}_{ff,T_{tt}}(t) = \bar{y} \quad \forall t > T_{tt}$ .

### Undershoot and Overshoot

The normalized undershoot  $y_{us}$  and overshoot  $y_{os}$  of the optimal output  $\bar{y}_{ff,T_{tt}}$ , during the single output-transition from  $z = 0$  to  $z = \bar{y}$ , are defined as

$$y_{us} = z_{us} = \inf_{t>0} \left\{ \frac{\bar{z}_{ff,T_{tt}}(t)}{\bar{y}}, 0 \right\} \quad (4.44)$$

and

$$y_{os} = z_{os} = \sup_{t>0} \left\{ \left( \frac{\bar{z}_{ff,T_{tt}}(t)}{\bar{y}} - 1 \right), 0 \right\} \quad (4.45)$$

since the undershoot and overshoot of the optimal output  $\bar{y}_{ff,T_{tt}}$  also corresponds to the normalized undershoot  $z_{us}$  and overshoot  $z_{os}$  of the optimal plant output  $\bar{z}_{ff,T_{tt}}$  (from Eq. (4.43)) and illustrated in Fig. 4.2.

#### 4.2.2 OBR Input with Multiple Output Transitions

The single-transition input  $\bar{u}_{ff,T_{tt}}$  in Eq. (4.42) is used to develop the OBR input  $u_{ff,k}$  for multiple transitions. In particular, for time  $t \geq t_k$ , the OBR input  $u_{ff,k}$  is defined as

$$u_{ff,k}(t) = u_{ff,k-1}(t) - \text{sgn}\{y(t_k)\} [\bar{u}_{ff,T_{tt}}(t - t_k)] \quad (4.46)$$

and the initial input is zero

$$u_{ff,0}(t) = 0, \quad \forall t > 0. \quad (4.47)$$

This input  $u = u_{ff,k}$  is applied as a feedforward to the system from output transition time  $t = t_k$  till the next transition at time  $t_{k+1}$ , i.e., during the time interval  $\mathbb{I}_k = [t_k, t_{k+1})$ . Thus, the OBR input in Eq. (4.46) aims to transition the output  $y$  by  $\pm\bar{y}$  at every transition-trigger time  $t_k$ . The issue is to quantify performance with such multiple-output transitions.

**Remark 11** (Computational Effort). *An advantage of adding the same feedforward input  $\pm\bar{u}_{ff,T_{tt}}$  at each output-transition time  $t_k$ ,  $k > 1$  in Eq. (4.46), is the ease of implementation in terms of the computational effort.*

### 4.3 Performance with Multiple Transitions

#### 4.3.1 OBR Performance Quantification

Towards establishing conditions for maintaining the output  $y$  inside the bounds  $y(t) \in [-\bar{Y}, \bar{Y}]$ , this section begins by showing that the OBR input in Eq. (4.46) aims to adjust the plant output  $z$  to cancel the disturbance  $d$ .

**Lemma 1** (OBR Disturbance Cancellation). *If the initial output  $y(0)$  is within the transition-trigger bound  $\bar{y}$  and the initial condition of the plant  $x(0)$  is zero, i.e.,*

$$|y(0)| \leq \bar{y}, \quad x(0) = 0, \quad (4.48)$$

*then the OBR input  $u_{ff,k}$  in Eq. (4.46) changes the plant output  $z$  to cancel the disturbance  $d(t_k)$  at trigger time  $t_k$ , i.e., for all  $k \geq 1$ ,*

$$\begin{aligned} z(t_{k+1}) &= -d(t_k), \\ y(t_{k+1}) &= d(t_{k+1}) - d(t_k) = \pm\bar{y}. \end{aligned} \quad (4.49)$$

*Proof.* At the first trigger time  $t_1$ , the output  $y$  and disturbance  $d$  are related by, from Eq. (4.2),

$$y(t_1) = z(t_1) + d(t_1) = d(t_1) = \pm\bar{y} \quad (4.50)$$

since the initial plant output  $z(0) = Cx(0)$  is zero, and the input is zero from Eq. (4.47), leading to

$$z(t_1) = z(0) = 0. \quad (4.51)$$

Let the results of the Lemma in Eq. (4.49) be satisfied for some  $k > 1$ . Then, the next input  $u_{ff,k+1}$  in Eq. (4.46) changes the plant output by the optimal plant output  $\bar{z}_{ff,T_{tt}}$ , i.e., for time  $t \in [t_{k+1}, t_{k+2})$ ,

$$z(t) = z(t_{k+1}) - \text{sgn}\{y(t_{k+1})\} [\bar{z}_{ff,T_{tt}}(t - t_{k+1})], \quad (4.52)$$

to change the plant output  $z$  by  $\pm\bar{y}$  to cancel the output at  $y(t_{k+1})$  within the transition time  $T_{tt}$ , i.e., for time  $t \in [t_{k+1} + T_{tt}, t_{k+2})$ ,

$$z(t) = z(t_{k+1}) - y(t_{k+1}). \quad (4.53)$$

By substituting expressions for  $z(t_{k+1})$  and  $y(t_{k+1})$  from Eq. (4.49) into Eq. (4.53) yields, for time  $t \in [t_{k+1} + T_{tt}, t_{k+2})$ ,

$$z(t) = -d(t_k) - [d(t_{k+1}) - d(t_k)] = -d(t_{k+1}), \quad (4.54)$$

which results in

$$\begin{aligned} z(t_{k+2}) &= -d(t_{k+1}), \\ y(t_{k+2}) &= z(t_{k+2}) + d(t_{k+2}) = d(t_{k+2}) - d(t_{k+1}). \end{aligned} \quad (4.55)$$

Moreover, the magnitude of the output  $|y(t_{k+2})| = \bar{y}$  by definition of the trigger time in Eq. (4.6). Thus, the results of the Lemma in Eq. (4.49) are satisfied for some  $k+1$ , and the Lemma's claims follow for any  $k \geq 1$ , by induction.  $\square$

The following Lemma establishes conditions for maintaining the output  $y$  inside the bounds  $y(t) \in [-\bar{Y}, \bar{Y}]$  by reducing the effects of disturbance  $d$  for all time  $t > 0$ .

**Lemma 2** (OBR Performance). *The output  $y$  remains between the desired bounds  $y \in [-\bar{Y}, \bar{Y}]$ , by using the input  $u_{ff,k}$  in Eq. (4.46), provided the change in disturbance  $d$  is bounded as*

$$|d(t_a) - d(t_b)| < \bar{y}, \quad \text{if } |t_a - t_b| < T_{trig} \quad (4.56)$$

for all time instants  $t_a$  and  $t_b$ , the minimum trigger time  $T_{trig}$  satisfies an inequality in Eq. (4.7), the transition-trigger bound  $\bar{y}$  is chosen to be sufficiently small, i.e.,

$$\bar{y} \leq \min \left\{ \frac{\bar{Y}}{2 + |z_{us}|}, \frac{\bar{Y}}{1 + |z_{os}|} \right\} \quad (4.57)$$

and, as in Eq. (4.48), the initial output  $y(0)$  is within the transition-trigger bound  $\bar{y}$  and the initial condition of the plant  $x(0)$  is zero.

*Proof.* Subtracting the output  $y(t_k)$  from the output  $y(t)$  at any time  $t \in [t_k, t_{k+1})$  with  $k \geq 1$  results in, from Eq. (4.2),

$$y(t) - y(t_k) = [z(t) + d(t)] - [z(t_k) + d(t_k)] \quad (4.58)$$

where the plant output  $z$  is changed with the input  $\bar{u}_{ff,k}$  in Eq. (4.46) as

$$z(t) = z(t_k) - \text{sgn}\{y(t_k)\} [\bar{z}_{ff, T_{tt}}(t - t_k)], \quad (4.59)$$

resulting in

$$\begin{aligned} y(t) &= y(t_k) - \text{sgn}\{y(t_k)\} \bar{z}_{ff, T_{tt}}(t - t_k) + \\ &\quad \{d(t) - d(t_k)\}, \quad \forall t \in [t_k, t_k + T_{trig}], \end{aligned} \quad (4.60)$$

$\forall k \geq 1$ . From Eq. (4.49), the magnitude of the output  $y(t_k)$  is  $\bar{y}$ . The maximum  $y_{sup}$  and minimum  $y_{inf}$  of the output  $y$  are found below using Eqs. (4.56) and (4.60) — these should remain within the desired output bounds  $[-\bar{Y}, \bar{Y}]$ . Assuming that the output  $y(t_k)$  at the trigger time  $t_k$  is positive,  $y(t_k) = +\bar{y} > 0$ , the supremum of the output  $y$  for time  $t \in [t_k, t_k + T_{trig}]$  is

$$\begin{aligned} y_{sup} &= \sup_{t \in [t_k, t_k + T_{trig}]} \{y(t)\} \\ &\leq \bar{y} + \sup_t \{-\bar{z}_{ff, T_{tt}}(t - t_k)\} + \sup_t \{d(t) - d(t_k)\} \\ &\leq \bar{y} + \bar{y}|z_{us}| + \bar{y} = \bar{y}(2 + |z_{us}|) \leq \bar{Y} \end{aligned} \quad (4.61)$$

where the normalized undershoot  $z_{us}$  is defined in Eq. (4.44), and the infimum  $y_{inf}$  of the output  $y$  for time  $t \in [t_k, t_k + T_{trig}]$  is (similarly)

$$y_{inf} \geq -\bar{y}(1 + |z_{os}|) \geq -\bar{Y} \quad (4.62)$$

where the normalized overshoot  $z_{os}$  is defined in Eq. (4.45). On the other hand, if the output  $y(t_k)$  at the trigger time  $t_k$  is negative,  $y(t_k) = -\bar{y} < 0$ , then the supremum  $y_{sup}$  and infimum  $y_{inf}$  of the output  $y$  for time  $t \in [t_k, t_k + T_{trig}]$  is (using arguments similar to the case when  $y(t_k) = +\bar{y}$ )

$$y_{sup} \leq \bar{y}(1 + |z_{os}|) \leq \bar{Y} \quad (4.63)$$

$$y_{inf} \geq -\bar{y}(2 + |z_{us}|) \geq -\bar{Y}. \quad (4.64)$$

Requiring that the minimum  $y_{inf}$  and maximum  $y_{sup}$  values of the output  $y$  in Eqs. (4.61)-(4.64) to be within the desired bounds  $[-\bar{Y}, \bar{Y}]$  results in the condition, in Eq. (4.57), of the Lemma.  $\square$

**Remark 12** (Rate of Change of Disturbance). *The condition, in Eq. (4.56) on the maximum change in the disturbance  $d$ , in the Lemma 2 can be met, for example, if the derivative of the disturbance  $\dot{d}$  is bounded as*

$$\dot{d}(t) \leq \frac{\bar{y}}{T_{trig}} \quad \forall t > 0. \quad (4.65)$$

**Remark 13** (Nonzero Initial Conditions). *Lemma 2 assumes that the initial plant state  $x(0)$  is zero as in Eq. (4.48). Nonzero initial conditions can be accommodated, by reducing the effective bound on the output from  $\bar{Y}$  to  $\bar{Y}_{new}$*

$$\bar{Y}_{new} = \bar{Y} - \bar{z}_0 \quad (4.66)$$

during the OBR design to ensure  $|y| \leq \bar{Y}$ , where  $\bar{z}_0$  is a bound on the magnitude of the plant output  $z$  over a set of possible initial plant states  $x(0)$ .

#### 4.3.2 OBR Input Boundedness

Bounds on the optimal single-transition input  $\bar{u}_{ff,T_{tt}}$  in Eq. (4.42) is established in the Lemma below, and used to establish bounds on the OBR input  $u_{ff,k}$  in Eq. (4.46) for the multiple-transition case.

**Lemma 3** (OBR Single-transition Input Boundedness). *The bound  $u^*(T_{tt})$  on the single transition input  $\bar{u}_{ff,T_{tt}}$  in Eq. (4.42) is finite, i.e.,*

$$\sup_{t>0}\{|\bar{u}_{ff,T_{tt}}|\} \leq u^*(T_{tt}) < \infty \quad (4.67)$$

for any finite transition time  $T_{tt} (< \infty)$ .

*Proof.* The single transition input  $\bar{u}_{ff,T_{tt}}$  in Eq. (4.42) consists of the transition input  $u_{tran}$  in Eq. (4.41) and the post-transition input  $u_{post}$  in Eq. (4.28). Given a finite transition time  $T_{tt} < \infty$ , the input  $u_{tran}$  in Eq. (4.41) is bounded since a linear system cannot have a finite escape time, i.e., the stable internal state  $\eta_{s,T_{tt}}$  in Eq. (4.40), the reduced-dimensional state  $\tilde{x}$  in Eq. (4.30), the optimal minimal-energy input  $\tilde{u}$  in Eq. (4.37), and the controllability gramian  $G_{tran}$  in Eq. (4.39) are bounded provided time  $0 \leq t \leq T_{tt} < \infty$ . The post-transition input  $u_{post}$  given in Eq. (4.28) is bounded for all time  $t \geq 0$  because the stable component  $\eta_s$  is bounded in Eq. (4.26).  $\square$

**Lemma 4** (OBR Input Boundedness). *The OBR input  $u_{ff,k}$  in Eq. (4.46) is bounded by*

$$\sup_{t \geq t_k} |u_{ff,k}(t)| \leq u^*(T_{tt}) \frac{D^*}{\bar{y}} < \infty \quad (4.68)$$

where the bound  $u^*(T_{tt})$  is in Eq. (4.67) provided the disturbance is bounded by, for all  $t \geq t_k$ , and  $k \geq 1$ ,

$$|d(t)| \leq D^* < \infty. \quad (4.69)$$

*Proof.* A bound  $U_k^*$  on the OBR input  $u_{ff,k}$   $k \geq 1$  in Eq. (4.46) can be established as

$$\begin{aligned} u^*(T_{tt})_k &= \sup_{t \geq t_k} \{|u_{ff,k}(t)|\} \\ &= \sup_{t \geq t_k} \left\{ \sum_{j=1}^k (\text{sgn} \{y(t_j)\}) |\bar{u}_{ff,T_{tt}}(t - t_j)| \right\} \\ &\leq \left| \sum_{j=1}^k \{\text{sgn} \{y(t_j)\} \sup_{t \geq t_k} \{|\bar{u}_{ff,T_{tt}}(t - t_j)|\}\} \right| \\ &\leq u^*(T_{tt}) \left| \sum_{j=1}^k \text{sgn} \{y(t_j)\} \right| \end{aligned} \quad (4.70)$$

where the bound  $u^*(T_{tt})$  is in Eq. (4.67). Then, from Eqs. (4.49) and (4.50)

$$U_k^* \leq u^*(T_{tt}) \left| \text{sgn} \{d(t_1)\} + \sum_{j=2}^k \{\text{sgn} \{d(t_j) - d(t_{j-1})\}\} \right|. \quad (4.71)$$

The disturbance  $d$  at any time  $t_k$  can be written as, from Eqs. (4.49) and (4.50),

$$\begin{aligned} d(t_k) &= d(t_1) + \sum_{j=2}^k [d(t_j) - d(t_{j-1})] \\ &= \bar{y} \left[ \text{sgn} \{d(t_1)\} + \sum_{j=2}^k \text{sgn} \{d(t_j) - d(t_{j-1})\} \right]. \end{aligned} \quad (4.72)$$

Then, from Eqs. (4.69) and (4.72),

$$\left| \text{sgn} \{d(t_1)\} + \sum_{j=2}^k \text{sgn} \{d(t_j) - d(t_{j-1})\} \right| \leq \frac{D^*}{\bar{y}}. \quad (4.73)$$

Substituting Eq. (4.73) into Eq. (4.71), the OBR input in Eq. (4.46) is bounded by  $U_k^* \leq (u^*(T_{tt})D^*/\bar{y})$ , and the Lemma follows since this is true for all  $k \geq 1$ .  $\square$

**Remark 14.** *Undershoot and overshoot in the plant output can occur during the single transition. Similarly the input  $u^*(T_{tt})$  needed for a single transition can become large due to peaking-type phenomena. In general, one would expect a tradeoff between the speed of transition (i.e., the minimum transition time  $T_{tt}$ ) and the bound on the input  $u^*(T_{tt})$  as well as undershoot and overshoot in the output. The OBR approach can be used with other single-transition inputs, e.g., found using approaches that can place bounds on the input-size for the single transition [77]. The analysis in the article can be used to develop a relationship between the overall bounds on the input (for multiple transitions) and the acceptable types of disturbances.*

#### 4.4 Numerical Example

The OBR feedforward approach is illustrated using a benchmark flexible structure example. The proposed approach is comparatively evaluated against other feedforward methods.

##### 4.4.1 System Model

Consider a flexible structure with two masses connected by a flexible rod and one side is connected to the fixed wall by another flexible rod as shown in Fig. 4.3. The input  $u$  is

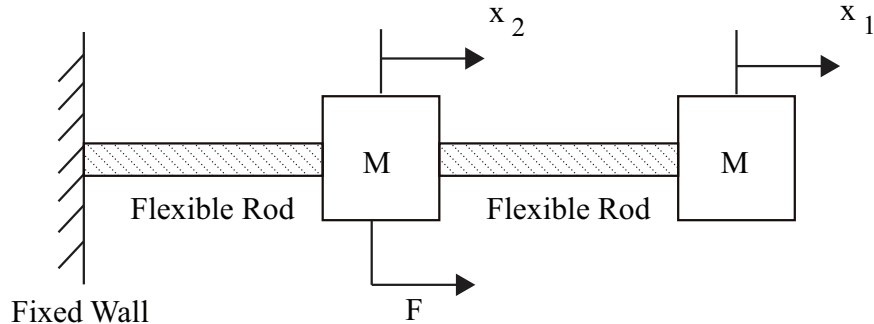


Figure 4.3: Flexible structure example.

force  $F$  applied to the left mass (displacement  $x_2$ ) and output  $z$  is the displacement  $x_1$  of the right mass. The example system is nonminimum-phase due to non-collocation of the input and output of the system.

The output  $z$  needs to cancel the disturbance  $d$  as shown in Fig. 2.4. Such a problem of positioning the end-point of a flexible structure arises in applications such as atomic force microscopes (AFMs) [11]. For example, the tip of an AFM cantilever needs to track an unknown sample surface (time trajectory) with bounded error to ensure that the tip-sample forces are bounded while the flexible structure (AFM cantilever) is scanned over a sample. Note that the problem of tracking an unknown sample surface with the end-point is equivalent to a disturbance-rejection problem.

The equation of motion, modeled using finite element method (FEM), e.g., see [78], is

given by

$$[M^L + M^r] \ddot{U} + [C^r] \dot{U} + [K^r] U = \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \quad (4.74)$$

where  $U = [x_1 \ x_2]'$ , the lumped mass  $M^L$  and effective mass  $M^r$  of the rods are

$$M^L = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}, \quad M^r = \frac{\rho_r A_r L_r}{6} \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}, \quad (4.75)$$

and the stiffness matrix  $K^r$  for the rods is given by

$$K^r = \frac{A_r E_r}{L_r} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \quad (4.76)$$

where the density, the cross-sectional area, length, and elastic modulus of each rod are  $\rho_r$ ,  $A_r$ ,  $L_r$ , and  $E_r$  respectively, and the damping matrix is  $[C^r] = \alpha^r [K^r]$ , where  $\alpha^r$  is a constant. In the simulation presented below, the system parameters were chosen as  $M = 10$ ,  $(\rho_r A_r L_r)/6 = 1$  (in Eq. (4.75)),  $(A_r E_r)/L_r = 1$  (in Eq. (4.76)), and  $\alpha^r = 0.1$  (in appropriate units). The system matrices  $A, B, C$  for the state space form of the system in Eq. (4.1) are given by

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.090 & 0.096 & -0.009 & 0.010 \\ 0.078 & -0.150 & 0.008 & -0.015 \end{bmatrix},$$

the matrix  $B = [0 \ 0 \ -0.006 \ 0.0719]'$ , the matrix  $C = [1 \ 0 \ 0 \ 0]$ , and the state  $x = [x_1 \ x_2 \ \dot{x}_1 \ \dot{x}_2]'$ , the input  $u$  is the force  $F$  in Fig. 4.3, i.e.,  $u = F$ , and the output  $z$  is the displacement of the right mass  $x_1$ , i.e.,  $z = x_1$ . The plant  $G$ , as shown in Fig. 2.4, is given by

$$G(s) = \frac{(-0.0060)(s + 0.951)(s - 1.051)}{(s^2 + 0.0028s + 0.028)(s^2 + 0.0211s + 0.211)}, \quad (4.77)$$

which is stable, with a nonminimum-phase zero at 1.051, and has relative degree  $r = 2$ .

#### 4.4.2 OBR Approach

The OBR performance is evaluated by comparing the disturbance rate-of-change that can be accommodated, i.e., without exceeding the desired output bound  $\bar{Y}$ . The disturbance  $d$  in Eq. (4.2) is chosen as a ramp with ramp-rate  $\dot{d}(t) = D^*/T_D = \dot{D}^*$  for all time  $0 \leq t \leq T_D$ . The disturbance is kept constant after time  $T_D$ , i.e., the disturbance is bounded by  $d(t) \leq d(T_D) = D^*$  for all time  $t > 0$ . A higher disturbance ramp-rate  $\dot{D}^*$  implies better performance, e.g., see Fig. 4.4.

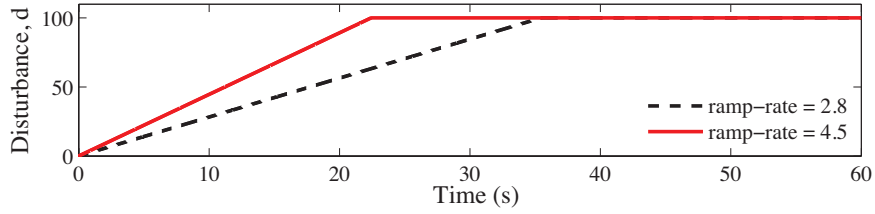


Figure 4.4: Disturbance  $d$  with a disturbance ramp-rate of  $\dot{D}^*$ .

#### Comparative Evaluation of OBR

Tracking results using the proposed OBR approach are compared with results using DC-gain feedforward without OBR and results using OBR with step input as feedforward.

#### DC-gain Feedforward (Without OBR)

The current value of the disturbance (if available) is applied to the plant  $G$  as a feedforward to cancel the disturbance  $d$  by using

$$u(t) = u_{ref}(t) = \frac{-d(t)}{(-CA^{-1}B)}, \quad \forall t > 0. \quad (4.78)$$

#### OBR with Step Input as Feedforward

The single-transition input  $\bar{u}_{ff,T_{tt}}$  used in the OBR input  $u_{ff,k}$  in Eq. (4.46) is replaced by the steady-state value of the step input  $u_{st,k}$  for  $t \geq t_k$  as

$$u_{st,k}(t) = u_{st,k-1}(t) - \text{sgn}\{y(t_k)\} \left[ \frac{\bar{y}}{(-CA^{-1}B)} \right] \quad (4.79)$$

where the transition-trigger bound  $\bar{y}$  can still be obtained using Eq. (4.57) (with information of the normalized undershoot  $z_{us}$  and overshoot  $z_{os}$  of the step response) and the initial input is zero  $u_{st,0}(t) = 0$  for all  $t > 0$  using a settling time  $T_{set}$  (e.g., a typical 5% settling time was used in the simulation below) as the minimum trigger time  $T_{trig}$ .

### *OBR Single Plant-Output Transition*

The optimal input  $\bar{u}_{ff,T_{tt}}$  for a single transition and the transition-trigger bound  $\bar{y}$  are interdependent. To find optimal input  $\bar{u}_{ff,T_{tt}}$ , it is necessary to know the final value of the plant output  $z$ , which is, according to Eq. (4.8), the transition-trigger bound  $\bar{y}$ . However, to find the transition-trigger bound  $\bar{y}$  from Eq. (4.57), it is necessary to know the normalized undershoot  $z_{us}$  and overshoot  $z_{os}$  associated with the optimal input  $\bar{u}_{ff,T_{tt}}$ .

To resolve this interdependence, the minimal-energy input  $\tilde{u}_{T_{tt},nom}$  for a set of specified transition times  $T_{tt}$  (chosen to be in the range  $T_{tt} \in [0.01, 10]$ s for the following simulations) was obtained from Eq. (4.37) when the final value of the plant output  $z(T_{tt})$  was chosen as a nominal value of one, i.e.,

$$z(T_{tt}) = \bar{z}_{nom} = 1. \quad (4.80)$$

Note that the minimal-energy input  $\tilde{u}_{T_{tt},nom}$  is linear in the final plant output  $z(T_{tt}) = \bar{z}_{nom}$ , from Eq. (4.37) and (4.38). Therefore, the corresponding plant output  $\bar{z}_{T_{tt},nom}$  was used to find the the normalized undershoot  $z_{us}$  and overshoot  $z_{os}$ , for all final values  $z(T_{tt})$  of the plant output  $z$ , by replacing  $\bar{z}_{ff,T_{tt}} = \bar{z}_{T_{tt},nom}$  in Eqs. (4.44) and (4.45). Next, the transition-trigger bound  $\bar{y}_{T_{tt}}$  was computed from Eq. (4.57), and the single-transition input  $\bar{u}_{ff,T_{tt}}$  for a given transition time  $T_{tt}$  was found using

$$\bar{u}_{ff,T_{tt}}(t) = \bar{y}_{T_{tt}} \bar{u}_{T_{tt},nom}, \quad \forall t > 0. \quad (4.81)$$

Using the nominal final value of the plant output  $\bar{z}_{nom} = 1$  in Eq. (4.80), the equilibrium internal states  $\bar{\eta}$  was obtained using Eqs. (4.24) and (4.27), where the transformation matrix

$T$  was

$$T = \begin{bmatrix} T_\xi \\ T_\eta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -0.725 & 0 & 0.689 \\ 0 & 0.689 & 0 & 0.725 \end{bmatrix}, \quad (4.82)$$

with the matrices in Eq. (4.19) given by  $A_s = -0.951$ ,  $A_u = 1.051$ ,  $B_{s,\xi} = \begin{bmatrix} -0.689 & -0.0689 \end{bmatrix}$ ,  $B_s = -8.271$ ,  $B_{u,\xi} = \begin{bmatrix} -0.725 & -0.0725 \end{bmatrix}$ , and  $B_u = -8.695$ , as the nominal stable internal state  $\bar{\eta}_{s,nom} = -0.725$  and the nominal unstable internal state  $\bar{\eta}_{u,nom} = 0.689$  with the nominal final value of the plant output  $\bar{z}_{nom} = 1$ . Solving Eq. (4.40) yielded the nominal stable internal state  $\eta_{s,nom}(T_{tt}) = 9.5$  for a transition time of  $T_{tt} = 1$ s with a first order prefilter with a break frequency of  $1/T_{tt}$  Hz (matrices  $A_f = -2\pi/T_{tt}$  and  $B_f = 2\pi/T_{tt}$ ) as required in Eq. (4.33). The normalized undershoot of  $z_{us} = -1.6$  and normalized overshoot of  $z_{os} = 0$  of the plant output  $z$  were as shown in Fig. 4.5 (Top right plot).

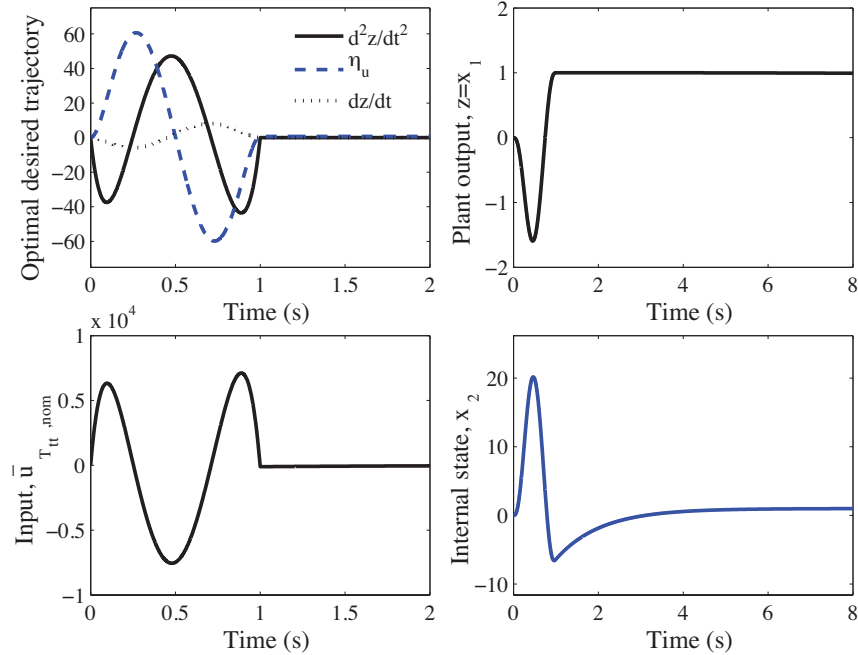


Figure 4.5: The nominal transition of plant output from  $z(0) = 0$  to  $z(T_{tt}) = 1$  with a transition time of  $T_{tt} = 1$ s. (Left) Optimal desired trajectory and nominal single-transition input  $\bar{u}_{T_{tt},nom}$ . (Right) States  $x_{1,nom}$  and  $x_{2,nom}$ .

### *Transition-Trigger Bound*

The constant output-error bound  $\bar{Y}$  should be less than the disturbance bound  $D^*$ , i.e.,  $\bar{Y} < D^*$  — otherwise the OBR method would not be necessary. In the following simulations, the output-error bound was chosen to be  $\bar{Y} = 10$ , which is 10% of the disturbance bound  $D^* = 100$ . The transition-trigger bound  $\bar{y}_{T_{tt}}$  was obtained using Eq. (4.57) as  $\bar{y}_{T_{tt}} = 2.78$ . The nominal input  $\bar{u}_{ff,nom}$  shown in Fig. 4.5 was scaled by the transition-trigger bound  $\bar{y}_{T_{tt}}$  as in Eq. (4.81) to find the single-transition input  $\bar{u}_{ff,T_{tt}}$  that changed the plant output  $z$  by the transition-trigger bound  $\bar{y}_{T_{tt}}$ .

### *4.4.3 Results and Discussions*

#### *Effect of Minimum Trigger-Time Choice*

The acceptable limit of the disturbance ramp-rate  $\dot{D}^*$  is inversely proportional to the minimum trigger time  $T_{trig}$  from Eq. (4.65). The minimum trigger time  $T_{trig}$ , in turn, needs to be at-least as large as the transition time  $T_{tt}$  from Eq. (4.7). Therefore better OBR performance (rejection of disturbances with larger ramp-rates  $\dot{D}^*$ ) can be achieved if the transition time  $T_{tt}$  is smaller. The results show this trend. For example, with a transition time of  $T_{tt} = 1$ s, the acceptable disturbance ramp-rate  $\dot{D}^*$  is 2.8 (Fig. 4.4) as shown in Fig. 4.6 (Left plots). The acceptable disturbance ramp-rate  $\dot{D}^*$  can be increased by 60% to  $\dot{D}^* = 4.5$  by decreasing the transition time to  $T_{tt} = 0.1$ s, as seen in Fig. 4.6 (Right plots). Thus, a smaller minimum trigger time  $T_{trig}$  provides better OBR performance.

A smaller minimum trigger time  $T_{trig}$  requires larger inputs. For example, the input magnitude of  $u^*(T_{tt}) = 3.5 \times 10^6$  required with a minimum trigger time of  $T_{trig} = 0.1$ s is more than two-orders-of-magnitude larger than the input magnitude of  $u^*(T_{tt}) = 2.1 \times 10^4$  needed with a minimum trigger time of  $T_{trig} = 1$ s as seen in Figs. 4.6 (Top plot) and 4.7 (Top left plot). Thus, a larger single-transition input  $\bar{u}_{ff,T_{tt}}$  is required for canceling larger ramp-rate  $\dot{D}^*$  disturbances. Additionally, a smaller minimum trigger time  $T_{trig}$  leads to a higher trigger frequency, and therefore, requires an input with a higher frequency content. This implies that the actuator bandwidth should be high and the system should accommodate potentially high-frequency vibrations (although the amplitude might be low) to reject large

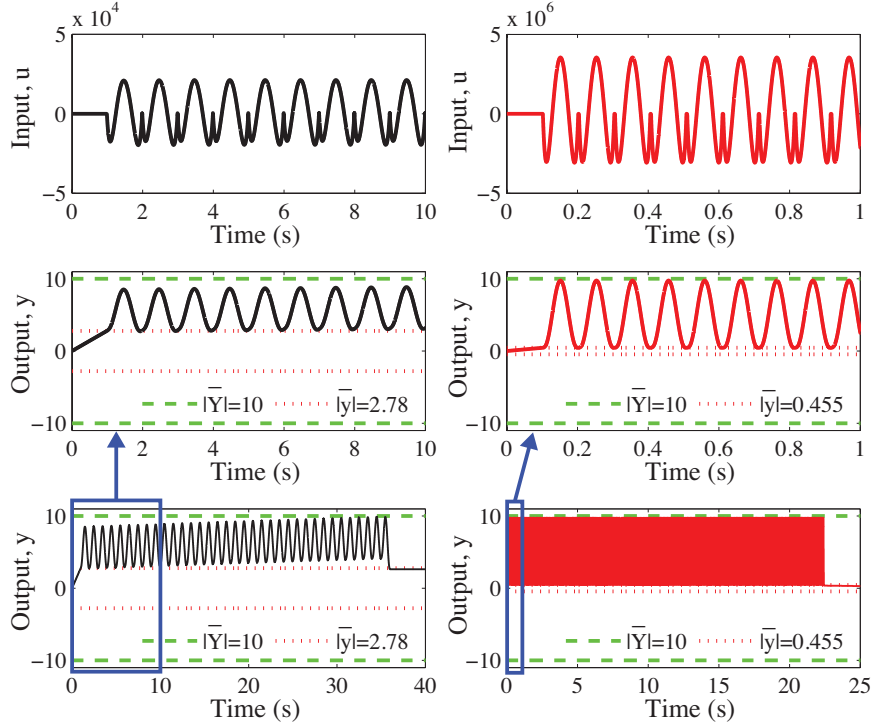


Figure 4.6: OBR performance with higher trigger frequency. OBR with a minimum trigger time of (Left):  $T_{trig} = 1\text{s}$  and (Right):  $T_{trig} = 0.1\text{s}$ . (Top): OBR input  $u_{ff,k}$ . (Middle and Bottom) Output  $y$ .

ramp-rate  $\dot{D}^*$  disturbances.

Reducing the minimum trigger time  $T_{trig}$ , however, also tends to increase the normalized undershoot  $z_{us}$  for nonminimum-phase systems, which reduces the transition trigger bound  $\bar{y}_{T_{tt}}$  from Eq. (4.57). This can, in turn, limit the maximum value of  $\bar{y}_{T_{tt}}/T_{trig}$  and thereby limit the maximum acceptable disturbance ramp-rate  $\dot{D}^*$  from Eq. (4.65). For example, as seen in Table 4.1 and Fig. 4.7 (Bottom left plot), the normalized undershoot of  $z_{us} = 20$  (with a minimum trigger time of  $T_{trig} = 0.1\text{s}$ ) is 12.5 times larger than the normalized undershoot of  $z_{us} = 1.6$  (with a minimum trigger time of  $T_{trig} = 1\text{s}$ ). The increase in the normalized undershoot  $z_{us}$  with a decrease in the minimum trigger time  $T_{trig}$  limits the OBR performance — the maximum disturbance ramp-rate  $\dot{D}^*$  is bounded to be approximately  $\dot{D}^* \approx 5$  as the minimum trigger time  $T_{trig}$  is decreased as seen in Fig. 4.7 (Bottom right

plot).

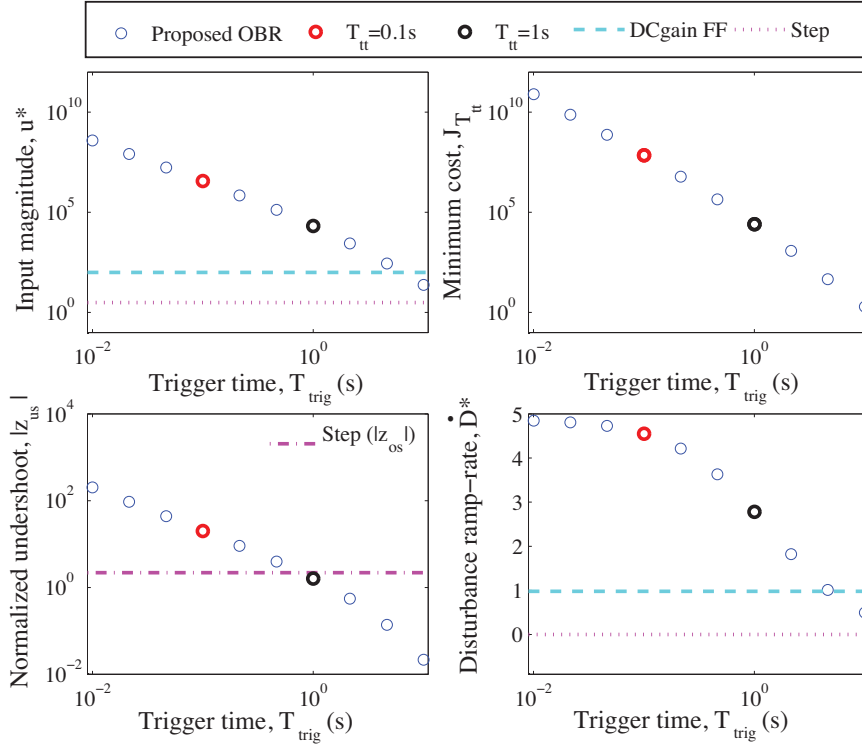


Figure 4.7: Simulation results with different minimum trigger time  $T_{trig}$ .

### Comparative Evaluation

The proposed OBR approach can improve performance when compared to the DC-gain feedforward, the cost is larger input and increased input bandwidth. For example, the acceptable disturbance ramp-rate of  $\dot{D}^* = 4.5$  with OBR with a minimum trigger time of  $T_{trig} = 0.1s$  is 4.6 times larger when compared to the acceptable disturbance ramp-rate of  $\dot{D}^* = 0.98$  with the DC-gain feedforward as seen in Table 4.1. However, the corresponding input magnitude of  $u^*(T_{tt}) = 3.5 \times 10^6$  with the OBR approach is substantially larger than input magnitude of  $u^*(T_{tt}) = 100$  with the DC-gain feedforward as seen in Table 4.1. Note that results using OBR with step input as feedforward is substantially worse than both the proposed optimal OBR and the DC-gain feedforward approaches due to typically low

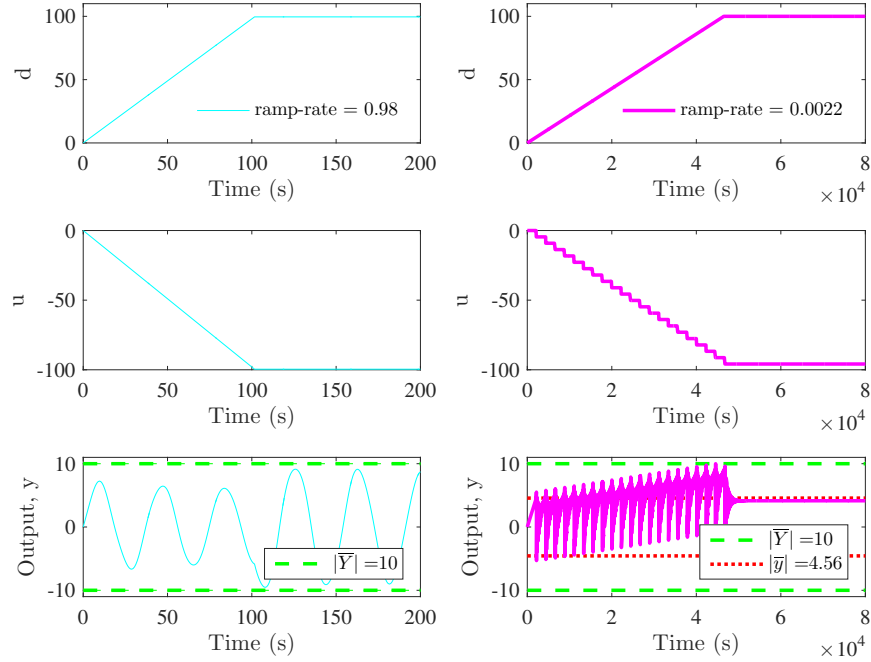


Figure 4.8: Example results. (Left): DC-gain feedforward (without OBR). (Right): Step-input OBR. (Top): Disturbance  $d$ . (Middle) Input  $u$ . (Left) Input  $u_{ref}$ , Eq. (4.78). (Right) Input  $u_{st,k}$ , Eq. (4.79). (Bottom): Output  $y$ .

structural damping of flexible structures that leads to a very large settling time of the system  $T_{set} = 2.2 \times 10^3$  s. The acceptable disturbance ramp-rate  $\dot{D}^*$  for the step-input based OBR is  $\dot{D}^* = 2.2 \times 10^{-3}$ , which is a couple of orders smaller than the DC-gain feedforward and the optimal OBR approach.

These results show that the proposed optimal OBR approach provides better disturbance cancellation rate when compared to the DC-gain-based method and significantly enhances the OBR performance when compared to the step-input-based approach.

### *Sensitivity to Parametric Uncertainties*

The effects of parametric uncertainties on the proposed OBR approach are numerically investigated here. Uncertainties in the system  $G(s)$  in Eq. (4.77) can be modeled as a

Table 4.1: Comparative evaluation. Transition-trigger bound  $\bar{y}$ , normalized undershoot  $|z_{us}|$ , input magnitude  $u^*(T_{tt})$ , and allowable disturbance ramp-rate  $\dot{D}^*$ .

	$\bar{y}$	$ z_{us} $	$u^*(T_{tt})$	$\dot{D}^*$
(i) DC-gain feedforward	-	-	100	0.98
(ii) Step input: $T_{set} = 2.2 \times 10^3 \text{s}$	4.56	$ z_{us}  = 0.188$  $( z_{os}  = 1.191)$	4.56	$2.2 \times 10^{-3}$
(iii) OBR $T_{trig} = 1 \text{s}$	2.78	$ z_{us}  = 1.6$	$2.1 \times 10^4$	2.8
(iii) OBR $T_{trig} = 0.1 \text{s}$	0.455	$ z_{us}  = 20.0$	$3.5 \times 10^6$	4.5

perturbed system  $G_p(s)$

$$G_p(s) = \frac{k_m(s + \omega_{z,1})(s + \omega_{z,2})}{(s^2 + 2\zeta_{p,1}\omega_{p,1}s + \omega_{p,1}^2)(s^2 + 2\zeta_{p,2}\omega_{p,2}s + \omega_{p,2}^2)}, \quad (4.83)$$

where  $k_m = -0.006$  is the gain,  $\zeta$  is the damping ratio,  $\omega$  is the frequency, and subscripts  $z$  and  $p$  represent zeros and poles of the system, respectively. Simulations were performed by varying the damping and frequency parameters — all variations were the same, i.e.,  $\Delta\zeta_{p,1} = \Delta\zeta_{p,2} = \Delta\omega_{z,1} = \Delta\omega_{z,2} = \Delta\omega_{p,1} = \Delta\omega_{p,2}$ , where the variation  $\Delta\beta$  of a parameter  $\beta$  from the nominal value  $\beta_{nom}$  is defined as

$$\Delta\beta = \frac{\beta - \beta_{nom}}{\beta_{nom}} \times 100 \text{ and } \Delta\dot{D}^* = \frac{\dot{D}^* - \dot{D}^*_{nom}}{\dot{D}^*_{nom}} \times 100 \quad (4.84)$$

and  $\Delta\dot{D}^*$  is the resulting variation in the acceptable ramp rate  $\dot{D}^*$  without recomputing the single-transition input  $\bar{u}_{ff,T_{tt}}$  and a fixed trigger time  $T_{trig}$  of 1s. The effects of  $\pm 5\%$  variations in the damping ratio  $\zeta$  and the natural frequency  $\omega$  of the system are shown in

Fig. 4.9. As can be seen in Fig. 4.9, parametric uncertainties in the damping ratio  $\zeta$  and

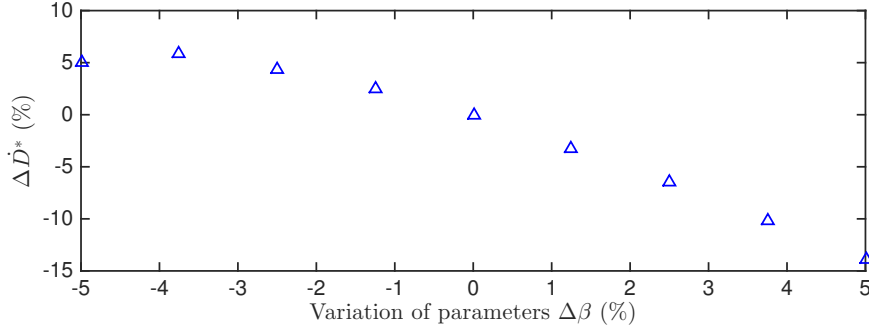


Figure 4.9: Simulation results. Variation  $\Delta\dot{D}^*$  in the acceptable ramp rate  $\dot{D}^*$  with parametric uncertainties  $\Delta\beta = \Delta\zeta_{p,1} = \Delta\zeta_{p,2} = \Delta\omega_{z,1} = \Delta\omega_{z,2} = \Delta\omega_{p,1} = \Delta\omega_{p,2}$ .

the natural frequency  $\omega$  results in a gradual degradation of performance of the proposed OBR approach. Therefore, a desired performance can be maintained at the expected level provided the modeling uncertainties are sufficiently small.

This chapter presents output-boundary regulation (OBR) using event-based feedforward for nonminimum-phase systems. The proposed OBR approach can reduce effects of unknown disturbances to be within given output-error bounds and the acceptable level of disturbance rate is quantified. Simulation results for a simplified flexible structure example shows substantial performance improvement with the proposed optimal OBR approach (more than 4 times increase in the rate-of-change of the disturbance that can be accommodated) when compared to DC-gain-based or step-input-based approaches. Additional results using SST when compared with OOT as well as perturbation analysis for a linear system example are provided in Appendix A and MATLAB code for this chapter is provided in Appendix C.

## Chapter 5

## EXTENSION TO NONLINEAR NONMINIMUM-PHASE SYSTEMS

This chapter<sup>1</sup> proposes an output-boundary regulation (OBR) approach that maintains the output-tracking error  $e$  within prescribed bounds  $|e| \leq \bar{E}$  for nonlinear nonminimum-phase systems for situations when preview information of the desired output is not available. The proposed OBR approach rapidly transitions the output error to zero at time events  $t_k$ ,  $k = 1, 2, \dots$  (separated by at-least time  $T_{trig}$ , i.e.,  $t_{k+1} - t_k \geq T_{trig}$ ) when the output-tracking error  $e$  exceeds a prescribed value  $\bar{e}$ . In particular, as illustrated in Fig. 5.1, whenever the magnitude of the output-tracking error  $|e(t_k)|$  exceeds a set (transition-trigger) value  $\bar{e}$ , which, in turn is less than the error-boundary value  $\bar{E}$ , i.e.,  $|e(t_k)| = \bar{e} < \bar{E}$ , the output-tracking error  $e$  is reset to zero, i.e.,  $e = 0$ . Note that the stable-inversion approach in [22]

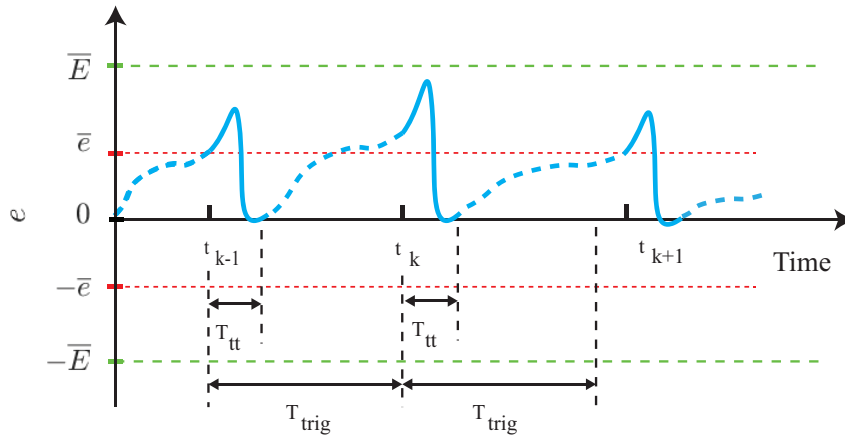


Figure 5.1: Schematic of proposed OBR. The output-tracking error  $e$  is reset to zero, i.e.,  $e = 0$  at time instants  $t_k$ , with  $k = 1, 2, \dots$ ), and separated by at-least time  $T_{trig}$ , i.e.,  $t_{k+1} - t_k \geq T_{trig}$ , when the output-tracking error  $|e(t_k)|$  is greater than a set (transition-trigger) value  $\bar{e}$ , which is less than the error-boundary value  $\bar{E}$ , i.e.,  $|e(t_k)| = \bar{e} < \bar{E}$ .

<sup>1</sup>Material in this chapter was submitted to ASME 2016 Dynamic Systems and Control Conference (DSCC) [14].

maintains exact-output tracking, i.e.,  $e = 0$ , but requires preview information. In contrast, the proposed OBR approach only achieves bounded tracking-error, i.e.,  $|e| \leq \bar{E}$ , but does not require preview information.

## 5.1 Problem Formulation and Solution Approach

### 5.1.1 Problem formulation

Let the plant  $G$  be a nonlinear time-invariant nonminimum-phase system described by

$$\begin{aligned}\dot{x}(t) &= f[x(t)] + g[x(t)]u(t), \\ y(t) &= h[x(t)]\end{aligned}\tag{5.1}$$

where the output  $y$  needs to track the desired output-trajectory  $y_d$  and the output-tracking error is given by

$$e(t) = y(t) - y_d(t)\tag{5.2}$$

as shown in Fig. 5.2. Formally, the goal of the OBR approach is to maintain the output-

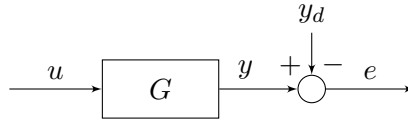


Figure 5.2: Schematic of output tracking for the nonlinear system  $G$ .

tracking error  $e$  within pre-specified output-error bounds

$$e(t) \in [-\bar{E}, \bar{E}], \quad \bar{E} > 0.\tag{5.3}$$

The control problem is to find the input  $u$  that maintains output-tracking errors within pre-specified values.

### 5.1.2 Solution approach

An output transition is triggered at time  $t_k$  from  $y(t_k)$  to  $y(t_k + T_{tt}) = y_f$  given by

$$y_f = y(t_k) - e(t_k)\tag{5.4}$$

where  $T_{tt}$  is the output transition time, if the output-tracking error  $e(t_k)$  is outside a transition trigger region  $(-\bar{e}, \bar{e})$  but within the tracking error bounds  $[-\bar{E}, \bar{E}]$ , i.e.,

$$e(t_k) \notin (-\bar{e}, \bar{e}) \subset [-\bar{E}, \bar{E}]. \quad (5.5)$$

Note that if the desired output  $y_d$  has not changed during this time interval  $[t_k, t_k + T_{tt}]$ , then the above output transition from  $y(t_k)$  to  $y(t_k + T_{tt})$  will bring the output-tracking error  $e$  back to zero during the transition time  $T_{tt}$ . Another transition is avoided while a previous transition is being completed, i.e., during the output transition interval  $[t_k, t_k + T_{tt}]$ , by specifying that the output transition time instants  $t_k$  be separated by at-least the transition time  $T_{tt}$ , i.e.,

$$t_k \geq t_{k-1} + T_{trig}, \quad k \geq 1 \quad (5.6)$$

where  $t_0 = -T_{trig}$  and the minimum transition-trigger time  $T_{trig}$  between transitions is greater than the transition time  $T_{tt}$ , i.e.,

$$T_{trig} \geq T_{tt}. \quad (5.7)$$

### 5.1.3 Polynomial based S-OBR and O-OBR

The single output transition in Eq. (5.4), e.g., from the output  $y_i = y(t_k)$  at initial time  $T_i = t_k$  to the output  $y(t_k + T_{tt}) = y_f$  at final time  $T_f = t_k + T_{tt}$ ,

$$y(T_i) = y_i \rightarrow y(T_f) = y_f \quad (5.8)$$

as in Eq. (5.4), is a critical component of the OBR approach. Three approaches to achieve the output transition are described and discussed below.

#### *Stable inversion approach*

Stable inversion can be used to exactly track a specified output such as a pre-specified smooth polynomial  $\hat{y}_{tran}$  in Fig. 5.3 (left plot), without undershoot and overshoot, to achieve the desired output transition from  $y_i$  to  $y_f$  in Eq. (5.8), as illustrated in Fig. 5.4 (left top plot). However, exact tracking of a general polynomial output  $\hat{y}_{tran}$  requires both non-zero pre-actuation ( $t \in (-\infty, T_i]$ ) input and post-actuation ( $t \in [T_f, \infty)$ ) input, e.g., [22]. This

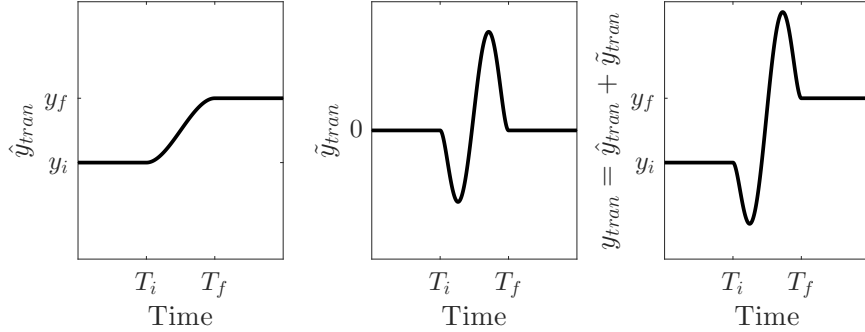


Figure 5.3: Transition polynomial: (left) main polynomial  $\hat{y}_{tran}$ , (middle) flexible polynomial  $\tilde{y}_{tran}$ , (right) transition polynomial  $y_{tran} = \hat{y}_{tran} + \tilde{y}_{tran}$ .

approach to output transition cannot be used for the OBR problem since pre-actuation cannot be applied when preview information of the desired trajectory  $y_d$  is not available.

#### *State-transition-based approach*

State-transition can be used to achieve the output transitions needed for OBR. This S-OBR approach modifies the pre-specified output polynomial  $\hat{y}_{tran}$  by adding a flexible polynomial  $\tilde{y}_{tran}$ , illustrated in Fig. 5.3 (middle plot)), to satisfy two conditions. The first condition is that the resulting total output polynomial  $y_{tran}$  (Fig. 5.3 (right plot)) achieves the desired output transition from  $y_i$  to  $y_f$  in Eq. (5.8) as in Fig. 5.4 (top middle plot). The second condition is that the flexible polynomial  $\tilde{y}_{tran}$  is chosen (e.g., see [15]) such that the exact tracking of the total output polynomial  $y_{tran}$  results in transitioning of the entire system state  $x$  from the initial state  $x_i = x(T_i)$  to a final stationary  $x(T_f) = x_f$

$$x(T_i) \rightarrow x(T_f) = x_f, \quad (5.9)$$

where  $x_f$  corresponds to the stationary equilibrium point  $(x_f, u_f, y_f)$ , i.e., from Eq. (5.1),

$$\begin{aligned} \mathbf{0} &= f[x_f] + g[x_f]u_f, \\ y_f &= h[x_f]. \end{aligned} \quad (5.10)$$

Assuming the system starts out at a stationary equilibrium point  $(x(0), u(0), y(0))$  at time  $t = 0$ , the final state  $x_f$  being a stationary equilibrium point implies that all initial state

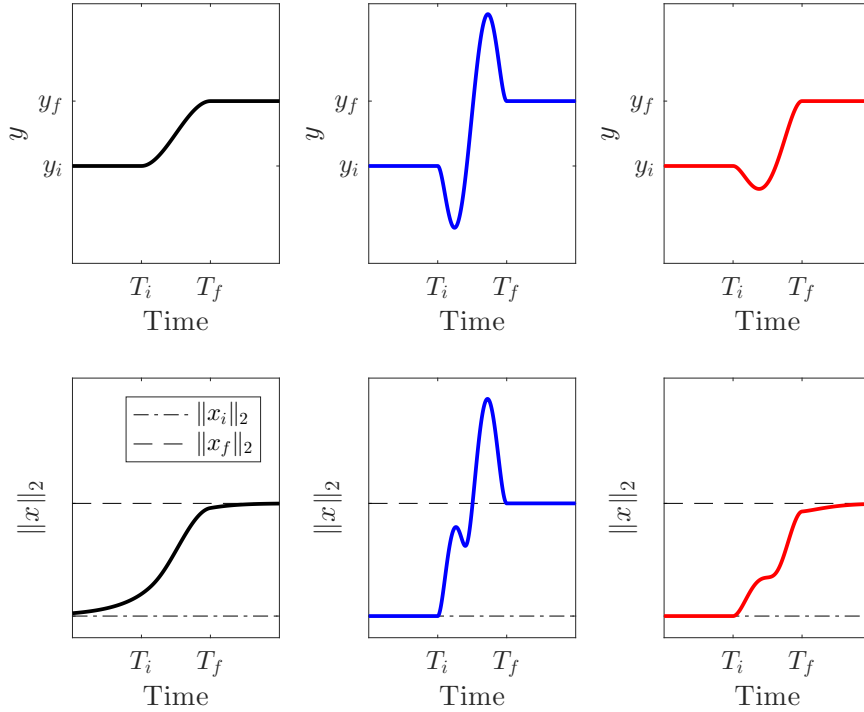


Figure 5.4: Single transition from the initial output  $y(T_i)$  to the final output  $y(T_f)$  within the same transition time  $T_{tt} = T_f - T_i$  for all cases: (left) stable-inversion approach (with pre- and post-actuation inputs), (middle) state-transition-based approach (without post-actuation input), (right) output-transition-based approach (with post-actuation input), (top) output  $y$ , and (bottom) Euclidean norm of the state  $\|x\|_2$ .

$x_i$  (which is final state for the previous transition) is also a stationary equilibrium point  $(x_i, u_i, y_i)$  satisfying

$$\begin{aligned} \mathbf{0} &= f[x_i] + g[x_i]u_i, \\ y_i &= h[x_i]. \end{aligned} \tag{5.11}$$

**Remark 15.** (*Stability of stationary equilibrium points*) In the following it is assumed that all the stationary equilibrium points  $x$  corresponding to output  $y$  are uniquely determined and are stable or have been stabilized using feedback. In this case, potential errors due to non-equilibrium initial condition  $x(0)$  at time  $t = 0$  will decrease over time.

Although the modification of the original polynomial output trajectory  $\hat{y}_{tran}$  by the flexible polynomial  $\tilde{y}_{tran}$  achieves the transition of the state (and therefore the output transition)

inside the time interval  $[T_i, T_f]$ , the modification also leads to undershoot and overshoot in the total output polynomial  $y_{tran}$  as illustrated in Fig. 5.3 (right plot). Large undershoot or overshoot can result large output-tracking error  $e$ , which in turn, can lead to loss of output boundary regulation.

#### *Output-transition-based approach*

The size of the overshoot or undershoot can be potentially reduced if the additional flexible polynomial  $\tilde{y}_{tran}$  does not have to achieve transition of the entire state as in Eq. (5.9) within a finite time interval  $[T_i, T_f]$ . Rather, the output-transition-based OBR (O-OBR) only achieves the output transition in the time interval  $[T_i, T_f]$  from  $y_i$  to  $y_f$  in Eq. (5.8). The state transition to  $x_f$  is then completed outside this time interval, i.e.,

$$x(t) \rightarrow x_f, \quad \text{as, } t \rightarrow \infty. \quad (5.12)$$

**Remark 16.** *(Improved performance with O-OBR) Performance of the two approaches to OBR (previous state-based transition and the proposed output-based transition) will be comparatively evaluated in terms of the acceptable rate of change of the output-trajectory. Since the state transition in Eq. (5.9) is also an acceptable output transition, i.e., transitions with the S-OBR approach are also acceptable transitions with the O-OBR problem, the output transition approach has more flexibility in choosing a transition. Therefore, in general, the performance of the O-OBR approach is anticipated to be at-least as good as (if not better than) the S-OBR approach.*

## **5.2 Output-Transition Input**

We start this section with the procedure to find the input that tracks a given output-transition polynomial for general nonlinear systems. Then, this inversion-based procedure is applied to S-OBR and O-OBR.

### 5.2.1 Inverse input:

Let the system  $G$  have a well-defined vector relative degree  $r$ . Then, the state  $x \in \mathbb{R}^{n \times 1}$  can be decomposed into known  $r$  components  $\xi$ , i.e., the output  $y$  and its time-derivatives

$$\xi = \left[ y \quad \frac{dy}{dt} \quad \frac{d^2y}{dt^2} \quad \cdots \quad \frac{d^{r-1}y}{dt^{r-1}} \right]^T \quad (5.13)$$

and remaining unknown components  $\eta$  representing the internal (or zero) dynamics given by

$$\dot{\eta}(t) = s_1[\xi(t), \eta(t)] + s_2[\xi(t), \eta(t)]u(t). \quad (5.14)$$

Furthermore, there exists an invertible transformation  $\Phi$  from the state  $x$  to the known components  $\xi$  and internal states  $\eta$ , i.e.,

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \Phi x = \begin{bmatrix} \Phi_\xi \\ \Phi_\eta \end{bmatrix} x \quad (5.15)$$

e.g., see [22] for more details. Thus, a well-defined vector relative degree  $r$  implies that a relationship between input  $u$  and output  $y$  can be found by differentiating the output as

$$\frac{d^r y}{dt^r}(t) = \alpha[\xi(t), \eta(t)] + \beta[\xi(t), \eta(t)]u(t) \quad (5.16)$$

with a locally invertible function  $\beta$ . From the above relation, the inverse input  $u_{inv}$  can be found to track a desired output polynomial  $y_{tran}$  as

$$u_{inv}(t) = \beta^{-1}[\xi_d(t), \eta(t)](y_{tran}^{(r)}(t) - \alpha[\xi_d(t), \eta(t)]), \quad (5.17)$$

where

$$y_{tran}^{(r)}(t) = \frac{d^r y_{tran}}{dt^r}(t) \quad (5.18)$$

and the known components  $\xi_d$  is found by evaluating Eq. (5.13) with  $y = y_{tran}$ . Substituting for the known components  $\xi_d$  and inverse input from Eq. (5.17) into the internal dynamics in Eq. (5.14) results in

$$\begin{aligned} \dot{\eta}(t) &= s_1[\xi_d(t), \eta(t)] + s_2[\xi_d(t), \eta(t)]u_{inv}(t) \\ &= s[\xi_d(t), \eta(t), y_{tran}^{(r)}(t)]. \end{aligned} \quad (5.19)$$

**Remark 17.** (*“Unstable” internal dynamics*) *The origin of the internal dynamics (with a constant output  $y_{tran}$ ) is unstable for nonminimum-phase systems. Therefore a challenge is to find a bounded reference internal state trajectory  $\eta$  for a given desired output-trajectory  $y_{tran}$  to ensure that the inverse input  $u_{inv}$  computed using this internal state  $\eta$  in Eq. (5.17) remains bounded.*

In the following, when the output  $y$  is a constant, it is assumed that the origin of the internal dynamics in Eq. (5.19) is an hyperbolic equilibrium point [79].

**Remark 18.** (*Hyperbolic equilibrium at the origin*) *Hyperbolicity allows the decoupling of the internal dynamics (with constant output) locally, into stable and unstable subsystems [79].*

### 5.2.2 Single transition using S-OBR

This section reviews the use of the polynomial-based approach in [15] for the single output transition in Eq. (5.8) for OBR. A sufficiently smooth output-transition polynomial  $y_{tran}$  is chosen as

$$y_{tran}(t) = \hat{y}_{tran}(t) + \tilde{y}_{tran}(p, t), \quad t \in [T_i, T_f], \quad (5.20)$$

where main output polynomial  $\hat{y}_{tran}$  is selected to satisfy the known boundary conditions at the beginning and end of the output transition, i.e., in terms of the output and its time derivatives  $\xi(T_i)$  and  $\xi(T_f)$  from Eq. (5.13) with  $y = \hat{y}_{tran}$ , i.e., the  $2r$  constraints given by

$$\begin{aligned} \xi(T_i) &= [y_i \quad \mathbf{0}_{1 \times (r-1)}]^T \\ \xi(T_f) &= [y_f \quad \mathbf{0}_{1 \times (r-1)}]^T. \end{aligned} \quad (5.21)$$

Additionally, for continuity of the inverse input in Eq. (5.17), the  $r^{th}$  derivative of the output  $y^{(r)}$  is also chosen as zero at the initial time  $T_i$  and final time  $T_f$ , i.e.,

$$\hat{y}_{tran}^{(r)}(T_i) = \hat{y}_{tran}^{(r)}(T_f) = 0. \quad (5.22)$$

The main polynomial can be given by

$$\hat{y}_{tran} = \sum_{i=0}^{2r+1} a_i \left( \frac{t - T_i}{T_{tt}} \right)^i, \quad t \in [T_i, T_f] \quad (5.23)$$

where coefficient  $a_i$  can be obtained by satisfying the total of  $2r + 2$  boundary conditions given in Eqs. (5.21) and (5.22) to obtain [15,80]

$$\hat{y}_{tran}(t) = y_i + (y_f - y_i) \sum_{i=r+1}^{2r+1} a_i \left( \frac{t - T_i}{T_{tt}} \right)^i \quad (5.24)$$

for  $t \in [T_i, T_f]$  where  $r$  is the relative degree and the coefficients  $a_i$  are given by

$$a_i = \frac{(-1)^{i-r-1} (2r+1)!}{i \times r! (i-r-1)! (2r+1-i)!}. \quad (5.25)$$

The flexible polynomial is given by

$$\tilde{y}_{tran}(p, t) = \sum_{i=0}^{2r+1} \tilde{a}_i(p) \left( \frac{t - T_i}{T_{tt}} \right)^i + \sum_{i=1}^{n-r} p_i \left( \frac{t - T_i}{T_{tt}} \right)^{i+2r+1} \quad (5.26)$$

for  $t \in [T_i, T_f]$  where the coefficient  $\tilde{a}_i(p)$  ( $\tilde{a}$  is a function of  $p$ ) can be obtained by satisfying the total of  $2r + 2$  zero boundary conditions, i.e.,  $\tilde{\xi}(T_i)$  and  $\tilde{\xi}(T_f)$  from Eq. (5.13) with  $y = \tilde{y}_{tran}$  are zero and the  $r^{th}$  time derivative  $\tilde{y}_{tran}^{(r)}$  is zero at the initial time  $T_i$  and final time  $T_f$ , i.e.,

$$\begin{bmatrix} \tilde{y}_{tran} \\ \tilde{y}_{tran}^{(1)} \\ \vdots \\ \tilde{y}_{tran}^{(r)} \end{bmatrix} (T_i) = \begin{bmatrix} \tilde{y}_{tran} \\ \tilde{y}_{tran}^{(1)} \\ \vdots \\ \tilde{y}_{tran}^{(r)} \end{bmatrix} (T_f) = \mathbf{0}_{(r+1) \times 1}, \quad (5.27)$$

where

$$\tilde{y}_{tran}^{(k)}(t) = \frac{d^k \tilde{y}_{tran}}{dt^k}(t). \quad (5.28)$$

In particular, the  $r + 1$  boundary conditions at initial time  $t = T_i$  in Eq. (5.27) can be satisfied with

$$\tilde{a}_0(p) = \tilde{a}_1(p) = \dots = \tilde{a}_r(p) = 0. \quad (5.29)$$

The remaining  $r + 1$  boundary conditions at the final time  $t = T_f$  in Eq. (5.27) can be used to find the remaining coefficients  $\tilde{a}$  as function of  $p$  as

$$\begin{bmatrix} \tilde{a}_{r+1}(p) \\ \vdots \\ \tilde{a}_{2r+1}(p) \end{bmatrix} = -C^{-1}P \quad (5.30)$$

where the  $i^{th}$  row and  $j^{th}$  component  $C_{i,j}$  of the  $(r+1) \times (r+1)$  invertible matrix  $C$  is given by

$$C_{i,j} = \frac{(j+r)!}{(j+r-i+1)! T_{tt}^{i-1}}, \quad (5.31)$$

and the  $i^{th}$  row and  $k^{th}$  component  $P_{i,k}$  of the  $(r+1) \times (n-r)$  matrix  $P$  is given by

$$P_{i,k} = p_k \frac{(k+2r+1)!}{(k+2r-i+2)! T_{tt}^{i-1}}. \quad (5.32)$$

The remaining parameters  $p_i$  in Eq. (5.26) are then chosen to satisfy the boundary conditions on the stationary internal states  $\eta$ , i.e., from Eq. (5.15),

$$\eta(T_i) = \eta_i = \Phi_\eta x_i \rightarrow \eta(T_f) = \eta_f = \Phi_\eta x_f. \quad (5.33)$$

The parameters  $p_i$  and the reference internal state trajectory  $\eta$  can be computed, e.g., by using the MATLAB `bvp4c` function that solves the boundary value problem in Eq. (5.33) for the internal dynamics  $\eta$  in Eq. (5.19), and results in a desired output-transition trajectory  $\bar{y}_d$

$$\bar{y}_d(t) = \begin{cases} y_{tran}(t) & , t \in [T_i, T_f], \\ y_f & , t > T_f. \end{cases} \quad (5.34)$$

The inverse input  $u_{inv}$  can be computed by substituting the desired output-transition trajectory  $\bar{y}_d^{(r)}$  from Eq. (5.34) and the corresponding reference internal state trajectory  $\eta$  into Eq. (5.17) as

$$u_{inv}(t) = \begin{cases} u_{tran}(t) & , t \in [T_i, T_f], \\ u_f & , t > T_f. \end{cases} \quad (5.35)$$

The desired output-transition trajectory  $\bar{y}_d$  in Eq. (5.34) can be used for both S-OBR and O-OBR by replacing the output-transition polynomial  $y_{tran}$ .

**Remark 19.** (*Internal state transitions to equilibrium*) In addition of the output transition from  $y_i$  to  $y_f$ , the state-transition-based approach transitions the entire internal state  $\eta$  to its equilibrium position at the end of transition, i.e.,  $\eta(T_f) = \eta_f$  as illustrated in Fig. 5.5 (top plot).

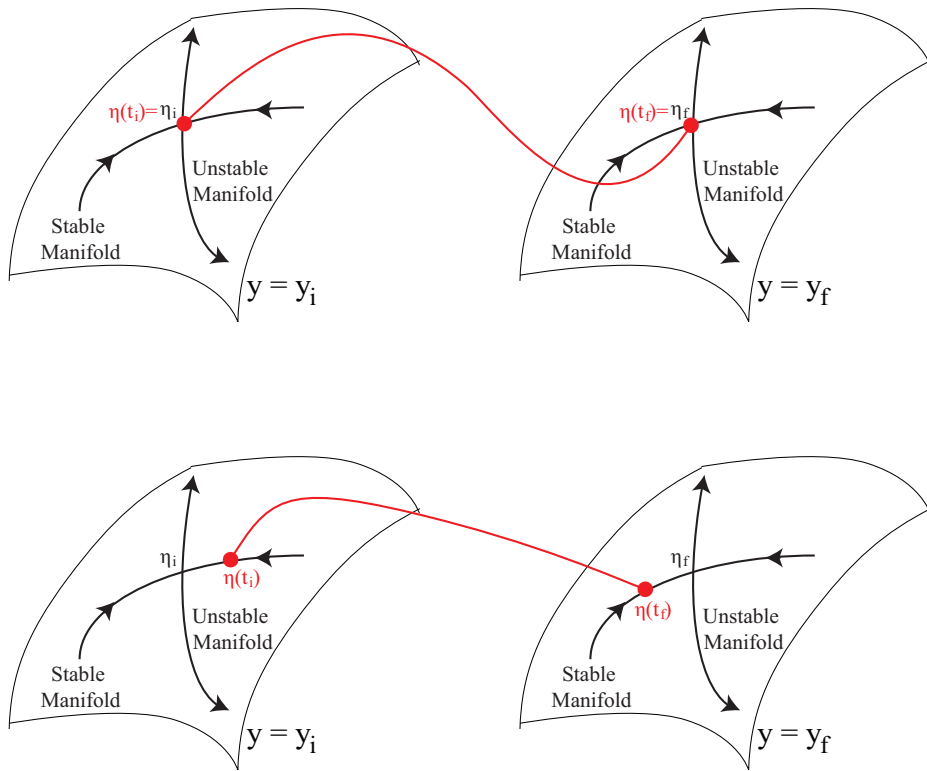


Figure 5.5: Internal state  $\eta$  transition to  $\eta_f$  is completed: by the output transition time  $t = T_f$  with S-OBR (top) and after the output transition at time  $t = T_f$  with O-OBR (bottom). In both cases, the output remains at  $y_f$  after time  $t = T_f$  till another transition is required.

### 5.2.3 Single transition using O-OBR

Rather than transition the entire internal state  $\eta$  to its final value  $\eta_f$  at time  $t = T_f$ , the output transition only requires that the output be transitioned by time  $T_f$ . The internal state  $\eta$  can evolve in a bounded manner on the stable manifold of the internal dynamics during post-actuation as illustrated in Fig. 5.5 (bottom plot).

#### *Decoupled internal dynamics*

To illustrate the approach, consider the case when the internal dynamics in Eq. (5.19) is decoupled into “stable”  $\eta_s$  and “unstable”  $\eta_u$  sub-systems with stable and unstable origins

( $\eta_s = 0, \eta_u = 0$ ) respectively, i.e.,

$$\dot{\eta}_s(t) = s[\xi_d(t), \eta_s(t), y_{tran}^{(r)}(t)], \quad (5.36)$$

$$\dot{\eta}_u(t) = s[\xi_d(t), \eta_u(t), y_{tran}^{(r)}(t)], \quad (5.37)$$

with

$$\eta = \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix}. \quad (5.38)$$

Let the stable and unstable components of the initial internal state  $\eta_i$  and final internal state  $\eta_f$  be

$$\eta_i = \begin{bmatrix} \eta_{s,i} \\ \eta_{u,i} \end{bmatrix}, \quad \eta_f = \begin{bmatrix} \eta_{s,f} \\ \eta_{u,f} \end{bmatrix}. \quad (5.39)$$

Then, the unstable internal state  $\eta_u(T_f)$  at the end of the output transition at time  $t = T_f$  must reach and remain at its desired final value  $\eta_{u,f}$ , i.e.,

$$\eta_u(T_f) = \eta_{u,f}, \quad (5.40)$$

otherwise the unstable internal dynamics  $\dot{\eta}_u$  in Eq. (5.37) can lead to an unbounded solution and an unbounded inverse input  $u_{inv}$ . In contrast, the stable internal state at the end of transition  $\eta_s(T_f)$  can be arbitrarily selected as it will eventually converge the final value  $\eta_{s,f}$  as illustrated in Fig. 5.5 (bottom plot), i.e.,

$$\eta_s(T_f + t) \rightarrow \eta_{s,f} \quad \text{as } t \rightarrow \infty. \quad (5.41)$$

Since any final stable internal state  $\eta_s(T_f)$  is acceptable, the output-transition approach only requires that the final  $n_u$  components of the unstable internal state  $\eta_u(T_f)$  be at its equilibrium value starting from the initial boundary value  $\eta_{u,i}$  defined in Eq. (5.39), i.e.,

$$\eta_u(T_i) = \eta_{u,i} \rightarrow \eta_u(T_f) = \eta_{u,f}. \quad (5.42)$$

Therefore, the flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.26) is modified to have only  $n_u$  parameters  $p_i$ , i.e.,

$$\tilde{y}_{tran}(p, t) = \sum_{i=0}^{2r+1} \tilde{a}_i(p) \left( \frac{t - T_i}{T_{tt}} \right)^i + \sum_{i=1}^{n_u} p_i \left( \frac{t - T_i}{T_{tt}} \right)^{i+2r+1}, \quad (5.43)$$

for  $t \in [T_i, T_f]$  to satisfy the  $n_u$  boundary conditions in Eq. (5.42). As in Eq. (5.30), the parameters  $\tilde{a}_i(p)$  are selected to satisfy the boundary conditions in Eq. (5.27). The parameters  $p_i$  and the reference unstable internal state trajectory  $\eta_u$  can be computed, e.g., by using the MATLAB `bvp4c` function that solves the boundary value problem in Eq. (5.42) for the unstable internal dynamics in Eq. (5.37). The resulting flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.43) and the main polynomial  $\hat{y}_{tran}$  in Eq. (5.24) results in the transition polynomial  $y_{tran}$  from Eq. (5.20). Then, the stable internal state  $\eta_s$  during the transition, i.e.,  $t \in [T_i, T_f]$  can be obtained by integrating Eq. (5.36). After completion of the output transition at time  $t = T_f$ , the stable internal state  $\eta_s$  can be obtained by integrating Eq. (5.36) with a constant output  $y = y_f$ , i.e.,

$$\dot{\eta}_s(t) = s[\xi_d(t), \eta_s(t), \mathbf{0}] \quad (5.44)$$

for  $t \in [T_i, \infty)$ . Once the internal states  $\eta$  are computed from the stable  $\eta_s$  and unstable  $\eta_u$  components using Eq. (5.38), the inverse input  $u_{inv}$  for the output transition can be found from Eq. (5.17).

### *Coupled internal dynamics*

Often, the internal dynamics in Eq. (5.19) cannot be decoupled into “stable”  $\eta_s$  and “unstable”  $\eta_u$  sub-systems. Therefore, an iterative approach is proposed below that exploits the ability to decouple the linearized internal dynamics. In particular, consider the following linearization of the internal dynamics in Eq. (5.19) as

$$A_\eta = \left. \frac{\partial}{\partial \eta} \left[ s[\xi_d(T_f), \eta(t), 0] \right] \right|_{\eta=\eta_f}, \quad (5.45)$$

and the linearized system is given by

$$\dot{\eta}(t) = A_\eta \eta(t) + \mathcal{P}(t) \quad (5.46)$$

where the nonlinear perturbation term  $\mathcal{P}$  is given by

$$\mathcal{P}(t) = s[\xi_d(t), \eta(t), \bar{y}_d^{(r)}(t)] - A_\eta \eta(t) \quad (5.47)$$

for  $t \in [T_i, \infty)$  where the desired output-transition trajectory  $\bar{y}_d$  can be obtained by using Eq. (5.34). Without the nonlinear perturbation terms, the origin of the internal dynamics

$$\dot{\eta}(t) = A_\eta \eta(t) \quad (5.48)$$

is hyperbolic (as in Remark 18), and can be decoupled by using a coordinate transformation

$$\eta = \Phi_{split} \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix} = \Phi_{split,u}^{-1} \eta = \begin{bmatrix} \Phi_{split,s}^{-1} \\ \Phi_{split,u}^{-1} \end{bmatrix} \eta \quad (5.49)$$

into stable  $\eta_s$  and unstable  $\eta_u$  sub-systems

$$\frac{d}{dt} \begin{bmatrix} \eta_s(t) \\ \eta_u(t) \end{bmatrix} = \begin{bmatrix} A_s & 0 \\ 0 & A_u \end{bmatrix} \begin{bmatrix} \eta_s(t) \\ \eta_u(t) \end{bmatrix}. \quad (5.50)$$

The flexible polynomial  $\tilde{y}_{tran}$  for output transition is selected in an iterative manner. At the initial iteration step  $m = 0$ , the initial internal dynamics  $\eta_0$  is considered to be zero, i.e.,

$$\eta_0(t) = 0, \quad \text{and}, \quad \forall t \in [T_i, \infty). \quad (5.51)$$

Then, at each iteration step  $m$  (where  $m = 1, 2, \dots$ ), the internal dynamics is considered to be the following linear approximation

$$\dot{\eta}_{s,m}(t) = A_s \eta_{s,m}(t) + \Phi_{split,s}^{-1} \mathcal{P}_m(t) \quad (5.52)$$

$$\dot{\eta}_{u,m}(t) = A_u \eta_{u,m}(t) + \Phi_{split,u}^{-1} \mathcal{P}_m(t) \quad (5.53)$$

for  $t \in [T_i, \infty)$  with stable linear subsystem  $\eta_{s,m}$  and unstable linear subsystem  $\eta_{u,m}$ . The perturbation term  $\mathcal{P}_m$  is given by, from Eq. (5.47),

$$\mathcal{P}_m(t) = s[\xi_{d,m}(t), \eta_{m-1}(t), \bar{y}_{d,m}^{(r)}(t)] - A_\eta \eta_{m-1}(t), \quad (5.54)$$

for  $t \in [T_i, \infty)$  where the desired output-transition trajectory  $\bar{y}_{d,m}$  can be obtained by using Eq. (5.34) with, from Eq. (5.20),

$$y_{tran,m}(t) = \hat{y}_{tran}(t) + \tilde{y}_{tran,m}(p, t), \quad t \in [T_i, T_f]. \quad (5.55)$$

The internal state  $\eta$  at the end of transition  $t = T_f$  must lie on the stable manifold and eventually converges to the desired final value, as illustrated in Fig. 5.5 (bottom plot),

$$\eta(t) \rightarrow \eta_f \quad \text{as } t \rightarrow \infty, \quad (5.56)$$

which implies that the unstable linear component  $\eta_{u,m}$  will tend to its desired final value, i.e.,

$$\eta_{u,m}(t) \rightarrow \eta_{u,f} \quad \text{as } t \rightarrow \infty. \quad (5.57)$$

where, from Eq. (5.49)

$$\eta_{u,f} = \Phi_{split,u}^{-1} \eta_f. \quad (5.58)$$

Thus, the transition of the unstable linear component  $\eta_{u,m}$  is from the initial state, obtained through Eq. (5.49),

$$\eta_{u,m}(T_i) = \Phi_{split,u}^{-1} \eta(T_i) \quad (5.59)$$

to its final equilibrium value  $\eta_{u,f}$  as in Eq. (5.57). Instead of using infinite post-transition time to solve Eq. (5.53) with boundary conditions in Eqs. (5.57) and (5.59), a sufficiently-large finite post-transition time  $T_{post}$  is used during computation, i.e., the boundary condition in Eq. (5.57) is modified to

$$\eta_u(T_f + T_{post}) = \eta_{u,f}. \quad (5.60)$$

Thus, the boundary value problem is to transition the  $n_u$  components of the unstable  $\eta_{u,m}$  internal dynamics at each iteration step  $m$  in Eq. (5.53) from the initial conditions given in Eq. (5.59) to the final condition in Eq. (5.60), i.e.,

$$\eta_{u,m}(T_i) = \Phi_{split,u}^{-1} \eta(T_i) \rightarrow \eta_{u,m}(T_f + T_{post}) = \Phi_{split,u}^{-1} \eta_f. \quad (5.61)$$

Therefore, the flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.26) is modified to have only  $n_u$  parameters  $p_{i,m}$ , i.e.,

$$\tilde{y}_{tran,m}(p_m, t) = \sum_{i=0}^{2r+1} \tilde{a}_{i,m}(p) \left( \frac{t - T_i}{T_{tt}} \right)^i + \sum_{i=1}^{n_u} p_{i,m} \left( \frac{t - T_i}{T_{tt}} \right)^{i+2r+1}, \quad (5.62)$$

for  $t \in [T_i, T_f]$  to satisfy the  $n_u$  boundary conditions in Eq. (5.61). As in Eq. (5.30), the parameters  $\tilde{a}_{i,m}(p)$  are selected to satisfy the boundary conditions in Eq. (5.27), i.e., using

Eqs. (5.29)-(5.32). The only difference is that  $P$  matrix in Eq. (5.30) is a smaller  $(r+1) \times n_u$  matrix. The parameters  $p_{i,m}$  and the reference unstable internal state trajectory  $\eta_{u,m}$  can be computed, as in the S-OBR case, by using the MATLAB `bvp4c` function that solves the boundary value problem in Eq. (5.61) for the unstable internal dynamics in Eq. (5.53). The resulting flexible polynomial  $\tilde{y}_{tran,m}$  in Eq. (5.62) and the main polynomial  $\hat{y}_{tran}$  in Eq. (5.24) results in the transition polynomial  $y_{tran,m}$  from Eq. (5.20). Then, the stable internal state  $\eta_{s,m}$  can be obtained by integrating Eq. (5.52) for  $t \in [T_i, T_f + T_{post}]$ . Provided the iterations converge (i.e., the difference between  $\eta_m$  and  $\eta_{m-1}$  is sufficiently small), the internal state  $\eta$  is estimated as  $\eta_m$ , i.e.,

$$\eta \approx \eta_m = \Phi_{split} \begin{bmatrix} \eta_{s,m} \\ \eta_{u,m} \end{bmatrix}. \quad (5.63)$$

Then, the inverse input  $u_{inv}$  for the output transition can be obtained from Eq. (5.17).

### 5.3 Simulation Results and Discussion

Simulations are provided below to illustrate the proposed S-OBR and O-OBR approaches.

#### 5.3.1 Example system and inverse input

Consider the following nonlinear nonminimum-phase system, which is an extension of the example system in [15]. The current system is modified to include both stable and unstable internal dynamics components rather than only a fully stable or a fully unstable internal dynamics as in [15]

$$\dot{x}_1(t) = -x_2^3(t) - x_3^3(t) + u(t) \quad (5.64)$$

$$\dot{x}_2(t) = x_1(t) - x_2(t) - x_2^3(t) + \beta_u x_3^3(t) \quad (5.65)$$

$$\dot{x}_3(t) = x_1(t) + x_3(t) + x_3^3(t) - \beta_s x_2^3(t) \quad (5.66)$$

where constants  $\beta_u = \beta_s = 1$  are used to represent a coupled internal dynamics system in the following examples.

**Remark 20.** (*Decoupled internal dynamics*) If the  $\beta$  constants are zero, i.e.,  $\beta_u = \beta_s = 0$ , then the internal dynamics is decoupled into stable  $x_2$  and unstable  $x_3$  subsystems when the output  $y = x_1$ .

### *Inverse input*

Let the output be  $y = x_1$ . Then, the example system has relative degree  $r = 1$ , the known component  $\xi$  in Eq. (5.13) is given by

$$\xi = y \quad (5.67)$$

and the internal state  $\eta$  in Eq. (5.15) is given by

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \quad (5.68)$$

where the transformation matrix  $\Phi$  in Eq. (5.15) is an identity matrix  $\mathbf{I}$ . The inverse input  $u_{inv}$  in Eq. (5.17) can be found from Eq. (5.64) as

$$u_{inv}(t) = \dot{y}_d(t) + \eta_1^3(t) + \eta_2^3(t). \quad (5.69)$$

Therefore, the inverse input  $u_{inv}$  in Eq. (5.69) is used to track a desired output transition trajectory  $\bar{y}_d$  in Eq. (5.34) as

$$u_{inv}(t) = \dot{\bar{y}}_d(t) + \eta_1^3(t) + \eta_2^3(t) \quad (5.70)$$

using S-OBR and O-OBR. Moreover, the internal  $\eta$  dynamics in Eq. (5.19) is given by

$$\dot{\eta}(t) = \begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \end{bmatrix} = \begin{bmatrix} \bar{y}_d(t) - \eta_1(t) - \eta_1^3(t) + \beta_u \eta_2^3(t) \\ \bar{y}_d(t) + \eta_2(t) + \eta_2^3(t) - \beta_s \eta_1^3(t) \end{bmatrix}. \quad (5.71)$$

### *5.3.2 Equilibrium state and use of feedback with feedforward*

Given desired equilibrium outputs  $y_i$  and  $y_f$  in Eq. (5.8), the equilibrium states  $x_i$  and  $x_f$  in Eq. (5.9) where

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y \\ \eta_1 \\ \eta_2 \end{bmatrix} \quad (5.72)$$

can be found by using Eqs. (5.10) and (5.11) to solve nonlinear equations in Eqs. (5.65) and (5.66) for equilibrium internal states  $\eta_i$  and  $\eta_f$ . For systems with decoupled internal dynamics with constants  $\beta_u = \beta_s = 0$ , nonlinear equations in Eqs. (5.65) and (5.66) can be solved for the exact solution of equilibrium internal states  $\eta_i$  and  $\eta_f$

$$0 = y_i - \eta_{1,i} - \eta_{1,i}^3, \quad 0 = y_f - \eta_{1,f} - \eta_{1,f}^3 \quad (5.73)$$

and

$$0 = y_i + \eta_{2,i} + \eta_{2,i}^3, \quad 0 = y_f + \eta_{2,f} + \eta_{2,f}^3. \quad (5.74)$$

Real roots of cubic equations in Eqs. (5.73) and (5.74) are equilibrium internal states  $\eta_i$  and  $\eta_f$ . However, nonlinear equations in Eqs. (5.65) and (5.66) need to be solved numerically to obtain equilibrium internal states  $\eta_i$  and  $\eta_f$  by combining Eqs. (5.65) and (5.66) into

$$\begin{aligned} 0 &= 2y_i - \eta_{1,i} - \eta_{1,i}^3 + \beta_u \eta_{2,i}^3 + \eta_{2,i} + \eta_{2,i}^3 - \beta_s \eta_{1,i}^3 \text{ and} \\ 0 &= 2y_f - \eta_{1,f} - \eta_{1,f}^3 + \beta_u \eta_{2,f}^3 + \eta_{2,f} + \eta_{2,f}^3 - \beta_s \eta_{1,f}^3. \end{aligned} \quad (5.75)$$

A MATLAB function `fsolve` can be used, for example, to solve for equilibrium internal states  $\eta_i$  and  $\eta_f$  in Eq. (5.75). For example, given a final equilibrium output  $y_f = 1$  and constants  $\beta_u = \beta_s = 1$ , solving Eq. (5.75) provides the final equilibrium internal states  $\eta_{1,f} = 0.590$  and  $\eta_{2,f} = -0.590$ . Note that equilibrium internal states  $\eta_i$  and  $\eta_f$  which are numerically obtained from Eq. (5.75) will be similar to those (real) values which are analytically obtained from Eqs. (5.73) and (5.74) if constants  $\beta_u = \beta_s = 0$  in Eq. (5.75).

#### *Use of feedback with feedforward*

Stability of the example system is determined by considering the linearized dynamics of the system in Eqs. (5.64)-(5.66)

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (5.76)$$

where

$$A = \begin{bmatrix} 0 & -3(x_{2,f})^2 & -3(x_{3,f})^2 \\ 1 & -1 - 3(x_{2,f})^2 & 3\beta_u(x_{3,f})^2 \\ 1 & -3\beta_s(x_{2,f})^2 & 1 + 3(x_{3,f})^2 \end{bmatrix} \quad (5.77)$$

and  $B = [1 \ 0 \ 0]^T$ . Eigenvalues of the linearized matrix  $A$  in Eq. (5.77) depend on the final equilibrium internal states  $x_{2,f}$ , and  $x_{3,f}$  which are dependent on the final equilibrium output  $y_f$  in Eqs. (5.73)- (5.75). As a result, stability of the example system depends on the final equilibrium output  $y_f$ , e.g., eigenvalues of the linearized matrix  $A$  are 0 and  $\pm 1$  at both origin and final equilibrium output  $y_f = 1$ . Since the equilibrium points of the example system are not stable, feedback needs to be added to stabilize the output tracking with the inverse input  $u_{inv}$  applied as a feedforward input, i.e.,

$$u(t) = u_{inv}(t) - K_{fb}\{x(t) - x_d(t)\} \quad (5.78)$$

where a feedback gain  $K_{fb}$  is used to stabilize the system along the desired state  $x_d$ . Gain-scheduling technique, e.g., in [81], could be used to design a linear feedback controller  $K_{fb}$ . However, for simplicity, a constant feedback gain  $K_{fb} = [3 \ 0 \ 18]$  found by verifying non-positive real parts of the eigenvalues of the linearized dynamics  $A$  in Eq. (5.77) at different desired final equilibrium output  $y_f \in [0, 1]$  is used throughout the example. Other choices of the constant feedback gain  $K_{fb}$  are also possible without major difference in simulation results.

### 5.3.3 Example output transition

The S-OBR and O-OBR approaches are described below for a single output transition in Eq. (5.8) from

$$y(T_i) = y_i = 0 \rightarrow y(T_f) = y_f = 1 \quad (5.79)$$

where the initial time  $T_i = 0$  and the final time  $T_f = T_i + T_{tt} = 1$ s. The initial and final stationary states in Eq. (5.9) are given by

$$x_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow x_f = \begin{bmatrix} 1 \\ 0.682 \\ -0.682 \end{bmatrix} \quad (5.80)$$

for the decoupled internal dynamics system example obtained from Eqs. (5.73) and (5.74) and

$$x_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow x_f = \begin{bmatrix} 1 \\ 0.590 \\ -0.590 \end{bmatrix} \quad (5.81)$$

for the coupled internal dynamics system example obtained from Eq. (5.75) with constants  $\beta_u = \beta_s = 1$ . A desired output transition trajectory  $\bar{y}_d$  in Eq. (5.34)

$$\bar{y}_d(t) = \begin{cases} y_{tran}(t) & , t \in [0, 1], \\ 1 & , t > 1. \end{cases} \quad (5.82)$$

is explained for both S-OBR and O-OBR cases in the following examples.

#### *Single transition example using S-OBR*

To find a transition polynomial  $y_{tran}$  in Eq. (5.20)

$$y_{tran}(t) = \hat{y}_{tran}(t) + \tilde{y}_{tran}(t) \quad (5.83)$$

for  $t \in [0, 1]$ s, the main polynomial  $\hat{y}_{tran}$  in Eq. (5.24) and the flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.26) are evaluated in this section. The main polynomial  $\hat{y}_{tran}$  in Eq. (5.24) is given by

$$\hat{y}_{tran}(t) = 3t^2 - 2t^3 \quad (5.84)$$

for  $t \in [0, 1]$ s where  $y_i = 0$  and  $y_f = 1$  from Eq. (5.79) and coefficients  $a_2 = 3$  and  $a_3 = -2$  obtained from Eq. (5.25) with relative degree  $r = 1$ . The flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.26) is given by

$$\tilde{y}_{tran}(p, t) = (p_1 + 2p_2)t^2 + (-2p_1 - 3p_2)t^3 + p_1t^4 + p_2t^5 \quad (5.85)$$

for  $t \in [0, 1]$ s with number of internal states  $n - r = 2$  in Eq. (5.26) where coefficients  $\tilde{a}_2(p_1) = 1$ ,  $\tilde{a}_3(p_1) = -2$ ,  $\tilde{a}_2(p_2) = 2$  and  $\tilde{a}_3(p_2) = -3$  are computed by using Eq. (5.30)

$$-C^{-1}P = - \begin{bmatrix} 3 & -1 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ -2 & -3 \end{bmatrix} \quad (5.86)$$

with matrices  $C$  and  $P$  obtained by using Eqs. (5.31) and (5.32) as

$$C = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \text{ and } P = \begin{bmatrix} 1 & 1 \\ 4 & 5 \end{bmatrix}. \quad (5.87)$$

Parameters  $p_1$  and  $p_2$  in Eq. (5.85) together with the reference internal state  $\eta$  can be obtained by using the MATLAB `bvp4c` function (see [15] for more details) for both decoupled and coupled internal dynamics system examples which are provided in the following sections.

#### *S-OBR for decoupled internal dynamics example*

By using the MATLAB `bvp4c` function to solve the following internal dynamics

$$\begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \end{bmatrix} = \begin{bmatrix} y_{tran}(t) - \eta_1(t) - \eta_1^3(t) \\ y_{tran}(t) + \eta_2(t) + \eta_2^3(t) \end{bmatrix} \quad (5.88)$$

for  $t \in [0, 1]$ s which is obtained from Eqs. (5.65) and (5.66) with

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \quad (5.89)$$

$\beta_u = \beta_s = 0$ , and the output transition polynomial  $y_{tran}$  from Eq. (5.20) as well as boundary conditions in Eq. (5.33)

$$\eta(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \eta(1) = \begin{bmatrix} 0.682 \\ -0.682 \end{bmatrix} \quad (5.90)$$

which come from Eq. (5.80), parameters  $p_1$  and  $p_2$  in Eq. (5.85) can be found as

$$p_1 = -843 \text{ and } p_2 = 339 \quad (5.91)$$

with the reference internal state  $\eta$  shown by using dashed lines in Fig. 5.6 (right plots) where the stable internal state  $\eta_s = \eta_1$  (top right plot) and the unstable internal state  $\eta_u = \eta_2$  (bottom right plot). The output  $y$  which exactly tracks the desired output transition trajectory  $\bar{y}_d$  in Eq. (5.34) is shown in Fig. 5.6 (bottom left plot) and the inverse input  $u_{inv}$  in Eq. (5.35) obtained from Eq. (5.69) is shown in Fig. 5.6 (top left plot).

*S-OBR for coupled internal dynamics example*

By using the MATLAB `bvp4c` function with the following internal dynamics

$$\begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \end{bmatrix} = \begin{bmatrix} y_{tran}(t) - \eta_1(t) - \eta_1^3(t) + \beta_u \eta_2^3(t) \\ y_{tran}(t) + \eta_2(t) + \eta_2^3(t) - \beta_s \eta_1^3(t) \end{bmatrix} \quad (5.92)$$

for  $t \in [0, 1]$ s which is obtained from Eqs. (5.65) and (5.66) with

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \quad (5.93)$$

$\beta_u = \beta_s = 1$ , and the output transition polynomial  $y_{tran}$  from Eq. (5.20) together with boundary conditions in (5.33)

$$\eta(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \eta(1) = \begin{bmatrix} 0.590 \\ -0.590 \end{bmatrix}, \quad (5.94)$$

which come from Eq. (5.81), parameters  $p_1$  and  $p_2$  in Eq. (5.85) can be obtained as

$$p_1 = -1.4 \times 10^3 \text{ and } p_2 = 569 \quad (5.95)$$

as well as the reference internal state  $\eta$ . To be comparable with O-OBR, internal states  $\eta_1$  and  $\eta_2$  are not shown in this S-OBR for the coupled internal dynamics system example but are transformed into stable  $\eta_s$  and unstable  $\eta_u$  internal states using the coordinate transformation matrix  $\Phi_{split}$  in Eq. (5.49) instead (see example of O-OBR for the value of the transformation matrix  $\Phi_{split}$ ). The reference stable  $\eta_s$  and unstable  $\eta_u$  internal states are shown with dashed lines in Fig. 5.8 (right plots). The output  $y$  which exactly tracks the desired output transition trajectory  $\bar{y}_d$  in Eq. (5.34) is shown in Fig. 5.8 (bottom left plot) and the inverse input  $u_{inv}$  in Eq. (5.35) obtained from Eq. (5.69) is shown in Fig. 5.8 (top left plot). By using S-OBR, the entire state  $x$  is transitioned to its final equilibrium value  $x_f$  at the end of transition time  $t = 1$ s.

*Single transition example using O-OBR*

Since the main polynomial  $\hat{y}_{tran}$  in Eq. (5.24) is valid for both S-OBR and O-OBR, i.e., from Eq. (5.84)

$$\hat{y}_{tran}(t) = 3t^2 - 2t^3 \quad (5.96)$$

for  $t \in [0, 1]$ s, only the flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.43) for the decoupled internal dynamics system example and  $\tilde{y}_{tran,m}$  in Eq. (5.62) for the coupled internal dynamics system example are evaluated in the following sections to obtain a transition polynomial  $y_{tran}$  in Eq. (5.20). The flexible polynomial  $\tilde{y}_{tran}$  in Eq. (5.43) for the decoupled internal dynamics system example with number of unstable internal states  $n_u = 1$  (the unstable internal state  $\eta_u = x_3$  for the decoupled internal dynamics system example) can be given by

$$\tilde{y}_{tran}(p, t) = p_1 t^2 - 2p_1 t^3 + p_1 t^4 \quad (5.97)$$

for  $t \in [0, 1]$ s where coefficients  $\tilde{a}_2(p_1) = 1$  and  $\tilde{a}_3(p_1) = -2$  are computed by using Eq. (5.30)

$$-C^{-1}P = - \begin{bmatrix} 3 & -1 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad (5.98)$$

with matrices  $C$  and  $P$  obtained by using Eqs. (5.31) and (5.32) as

$$C = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \text{ and } P = \begin{bmatrix} 1 \\ 4 \end{bmatrix}. \quad (5.99)$$

Similarly, the flexible polynomial  $\tilde{y}_{tran,m}$  in Eq. (5.62) for the coupled internal dynamics system example can be given by

$$\tilde{y}_{tran,m}(p_m, t) = p_{1,m} t^2 - 2p_{1,m} t^3 + p_{1,m} t^4 \quad (5.100)$$

for  $t \in [0, 1]$ s. Parameters  $p_1$  in Eq. (5.97) and  $p_{1,m}$  in Eq. (5.100) together with the reference unstable internal state  $\eta_u$  (and  $\eta_{u,m}$ ) can be obtained by using the MATLAB `bvp4c` function for both decoupled and coupled internal dynamics system examples which are provided in the following sections.

#### *O-OBR for decoupled internal dynamics example*

If the internal dynamics is decoupled, i.e., constants  $\beta_u = \beta_s = 0$  in Eq. (5.92), the stable  $\eta_s$  and unstable  $\eta_u$  internal dynamics in Eqs. (5.36) and (5.37) are given by

$$\begin{bmatrix} \dot{\eta}_s(t) \\ \dot{\eta}_u(t) \end{bmatrix} = \begin{bmatrix} \bar{y}_d(t) - \eta_s(t) - \eta_s^3(t) \\ \bar{y}_d(t) + \eta_u(t) + \eta_u^3(t) \end{bmatrix} \quad (5.101)$$

for  $t \in [0, \infty)$  from Eqs. (5.65) and (5.66) where the desired output transition trajectory  $\bar{y}_d$  is given in Eq. (5.34) with the stable  $\eta_s$  and unstable  $\eta_u$  internal states in Eq. (5.38) given by

$$\begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}. \quad (5.102)$$

The initial and final stationary states in Eq. (5.39) are given in Eq. (5.80)

$$x_i = \begin{bmatrix} y_i \\ \eta_{s,i} \\ \eta_{u,i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow x_f = \begin{bmatrix} y_f \\ \eta_{s,f} \\ \eta_{u,f} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.682 \\ -0.682 \end{bmatrix}. \quad (5.103)$$

By using the MATLAB `bvp4c` function to solve the boundary value problem of the unstable  $\eta_u$  internal dynamics given in Eq. (5.101)

$$\dot{\eta}_u(t) = y_{tran}(t) + \eta_u(t) + \eta_u^3(t) \quad (5.104)$$

for  $t \in [0, 1]$ s using a transition polynomial  $y_{tran}$  in Eq. (5.20) with the main polynomial  $\hat{y}_{tran}$  and the flexible polynomial  $\tilde{y}_{tran}$  given in Eqs. (5.96) and (5.97) and boundary conditions in Eq. (5.42)

$$\eta_u(0) = 0 \rightarrow \eta_u(1) = -0.682, \quad (5.105)$$

the parameter  $p_1$  in Eq. (5.97) can be obtained as

$$p_1 = -22.3 \quad (5.106)$$

together with the reference unstable internal state  $\eta_u$  which is shown by using a dotted line in Fig. 5.6 (bottom right plot). The reference stable internal state  $\eta_s$  can be found by integrating Eq. (5.44) which is given in Eq. (5.101)

$$\dot{\eta}_s(t) = \bar{y}_d(t) - \eta_s(t) - \eta_s^3(t) \quad (5.107)$$

for  $t \in [0, \infty)$  with initial stable internal state  $\eta_s(0) = 0$  and is shown by using a dotted line in Fig. 5.6 (top right plot) where the desired output transition trajectory  $\bar{y}_d$  is given in Eq. (5.34). The output  $y$  which exactly tracks the desired output transition trajectory  $\bar{y}_d$  is shown in Fig. 5.6 (bottom left plot) and the inverse input  $u_{inv}$  in Eqs. (5.17) and (5.69)

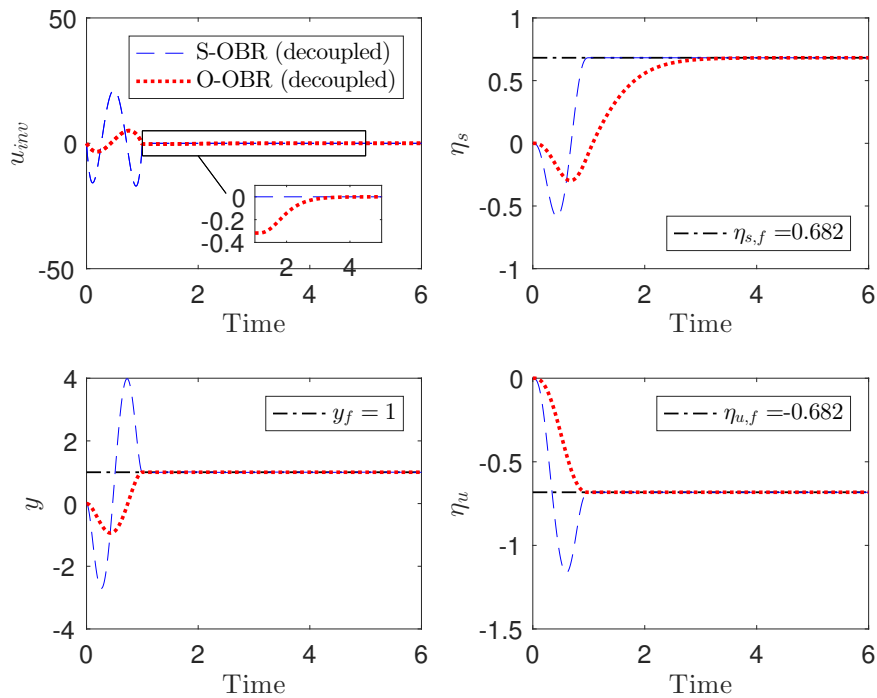


Figure 5.6: Single output transition using S-OBR and O-OBR for the decoupled internal dynamics system example.

is shown in Fig. 5.6 (top left plot). As shown in Fig. 5.6 (right bottom plot), the unstable internal state  $\eta_u$  reaches and remains at its final equilibrium value  $\eta_{u,f}$  at the end of transition  $t = 1$ s for both S-OBR and O-OBR while the stable internal state  $\eta_s$  eventually reaches its final equilibrium value  $\eta_{s,f}$  during post-transition in the O-OBR case shown by using a dotted line in Fig. 5.6 (top right plot). Results using O-OBR (dotted lines) for the decoupled internal dynamics system example are compared to results using S-OBR (dashed lines) for the same decoupled internal dynamics system example in Fig. 5.6 and O-OBR shows improvements with smaller magnitude of both input  $u_{inv}$  and output  $y$  than S-OBR

*O-OBR for coupled internal dynamics example*

The internal  $\eta$  dynamics

$$\dot{\eta}(t) = \begin{bmatrix} \dot{\eta}_1(t) \\ \dot{\eta}_2(t) \end{bmatrix} = \begin{bmatrix} \bar{y}_d(t) - \eta_1(t) - \eta_1^3(t) + \beta_u \eta_2^3 \\ \bar{y}_d(t) + \eta_2(t) + \eta_2^3(t) - \beta_s \eta_1^3 \end{bmatrix} \quad (5.108)$$

for  $t \in [0, \infty)$  from Eqs. (5.65) and (5.66) where the desired output transition trajectory  $\bar{y}_d$  is given in Eq. (5.34) cannot be directly decoupled into stable  $\eta_s$  and unstable  $\eta_u$  internal states when both constants  $\beta_u$  and  $\beta_s$  are not zero. An iterative approach is used in this section to find the flexible polynomial  $\tilde{y}_{tran,m}$  in Eq. (5.62) at iteration steps  $m$  for the system with coupled internal dynamics. The linearized internal dynamics in Eq. (5.46) is given by using Eq. (5.45) as

$$A_\eta = \left[ \begin{array}{c|c} -1 - 3(\eta_{1,f})^2 & 3\beta_u(\eta_{2,f})^2 \\ \hline -3\beta_s(\eta_{1,f})^2 & 1 + 3(\eta_{2,f})^2 \end{array} \right] \left| \begin{array}{l} \eta_{1,f} = 0.590 \\ \eta_{2,f} = -0.590 \end{array} \right. \quad (5.109)$$

and the nonlinear perturbation term  $\mathcal{P}$  in Eq. (5.47) as

$$\mathcal{P}(t) = \left[ \begin{array}{c|c} \bar{y}_d(t) + 3(\eta_{1,f})^2 \eta_1(t) - \eta_1^3(t) - & \\ \hline 3\beta_u(\eta_{2,f})^2 \eta_2(t) + \beta_u \eta_2^3(t) & \\ \bar{y}_d(t) - 3(\eta_{2,f})^2 \eta_2(t) + \eta_2^3(t) + & \\ \hline 3\beta_s(\eta_{1,f})^2 \eta_1(t) - \beta_s \eta_1^3(t) & \end{array} \right] \left| \begin{array}{l} \eta_{1,f} = 0.590 \\ \eta_{2,f} = -0.590 \end{array} \right. \quad (5.110)$$

for  $t \in [0, \infty)$ . The invertible transformation matrix  $\Phi_{split}$  in Eq. (5.49) which can be obtained from eigenvectors of matrix  $A_\eta$  in Eq. (5.109) is given by

$$\Phi_{split} = \begin{bmatrix} -0.964 & -0.265 \\ -0.265 & -0.964 \end{bmatrix} \quad (5.111)$$

and the stable  $\eta_s$  in Eqs. (5.50) and (5.52) and unstable  $\eta_u$  in Eqs. (5.50) and (5.53) linearized internal dynamics are given by

$$A_s = -1.757, \quad \Phi_{split,s}^{-1} = \left[ \begin{array}{c|c} -1.122 & 0.308 \end{array} \right] \quad (5.112)$$

and

$$A_u = 1.757, \Phi_{split,u}^{-1} = \begin{bmatrix} 0.308 & | & -1.122 \end{bmatrix} \quad (5.113)$$

where the nonlinear perturbation term  $\mathcal{P}_m$  at each iteration step  $m$  in Eq. (5.54) is given by using Eq. (5.110)

$$\mathcal{P}_m(t) = \left[ \begin{array}{c} \bar{y}_{d,m}(t) + 3(\eta_{1,f})^2 \eta_{1,m-1}(t) - \eta_{1,m-1}^3(t) - \\ 3\beta_u (\eta_{2,f})^2 \eta_{2,m-1}(t) + \beta_u \eta_{2,m-1}^3(t) \\ \hline \bar{y}_{d,m}(t) - 3(\eta_{2,f})^2 \eta_{2,m-1}(t) + \eta_{2,m-1}^3(t) + \\ 3\beta_s (\eta_{1,f})^2 \eta_{1,m-1}(t) - \beta_s \eta_{1,m-1}^3(t) \end{array} \right] \left| \begin{array}{l} \eta_{1,f} = 0.590 \\ \eta_{2,f} = -0.590 \end{array} \right. \quad (5.114)$$

for  $t \in [0, \infty)$  with the desired output transition trajectory  $\bar{y}_{d,m}$  at  $m^{th}$  iteration step given in Eq. (5.34) that contains a transition polynomial  $y_{tran,m}$  in Eq. (5.55) with the main polynomial  $\hat{y}_{tran}$  and the flexible polynomial  $\tilde{y}_{tran,m}$  given in Eqs. (5.96) and (5.100) and the zero initial internal state  $\eta_0$  is given in Eq. (5.51). By using the MATLAB `bvp4c` function to solve the boundary value problem of linearized unstable  $\eta_u$  internal dynamics in Eq. (5.53)

$$\dot{\eta}_{u,m}(t) = 1.757\eta_{u,m}(t) + \begin{bmatrix} 0.308 & | & -1.122 \end{bmatrix} \mathcal{P}_m(t) \quad (5.115)$$

for  $t \in [0, 6]$ s where the post-transition time  $T_{post} = 5$ s, the nonlinear perturbation term  $\mathcal{P}_m$  is given in Eq. (5.114), and boundary conditions in Eq. (5.61) as

$$\eta_{u,m}(0) = 0 \rightarrow \eta_{u,m}(6) = \Phi_{split,u}^{-1} \begin{bmatrix} 0.590 \\ -0.590 \end{bmatrix} = 0.843, \quad (5.116)$$

the parameter  $p_{1,m}$  in Eq. (5.100) can be obtained as

$$p_{1,m=4} = -22.5, \quad (5.117)$$

see Fig. 5.7 (right plot) together with the reference unstable internal state  $\eta_{u,m}$  which is shown by using a dotted line in Fig. 5.8 (bottom right plot) after  $4^{th}$  iteration step, i.e.,  $\eta_{u,m=4}$ . The reference stable internal state  $\eta_{s,m}$ , which is shown by using a dotted line in Fig. 5.8 (top right plot) after  $4^{th}$  iteration step, i.e.,  $\eta_{s,m=4}$ , can be found by integrating the stable  $\eta_s$  internal dynamics in Eq. (5.52)

$$\dot{\eta}_{s,m}(t) = -1.757\eta_{s,m}(t) + \begin{bmatrix} -1.122 & | & 0.308 \end{bmatrix} \mathcal{P}_m(t) \quad (5.118)$$

for  $t \in [0, 6]$ s with initial stable internal state  $\eta_{s,m}(0) = 0$  where the nonlinear perturbation term  $\mathcal{P}_m$  is given in Eq. (5.114). The reference internal state  $\eta$  can be obtained by using Eq. (5.63) as

$$\eta \approx \eta_{m=4} = \Phi_{split} \begin{bmatrix} \eta_{s,m=4} \\ \eta_{u,m=4} \end{bmatrix} \quad (5.119)$$

where the transformation matrix  $\Phi_{split}$  is given in (5.111). As number of iteration step  $m$  increases, difference between the  $m^{th}$  and  $(m-1)^{th}$  iteration steps for the stable  $\eta_s$  and unstable  $\eta_u$  internal states decreases as shown in Fig. 5.7 (left plot). Results using O-OBR

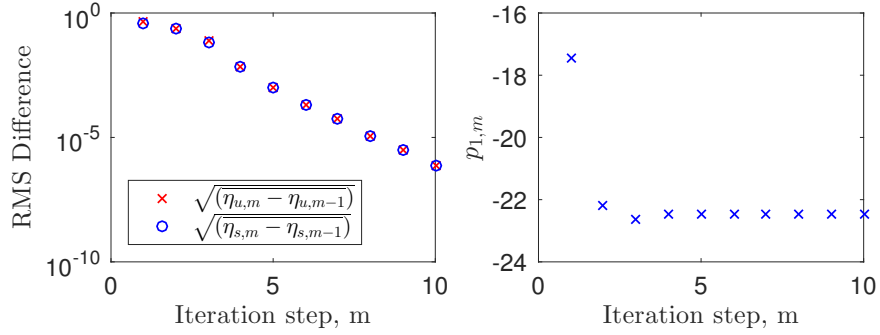


Figure 5.7: Results from O-OBR iteration process: (left plot) root mean squared (RMS) differences of stable  $\eta_s$  and  $\eta_u$  unstable internal states and (right plot) parameter  $p_{1,m}$  in Eq. (5.100).

(after 4<sup>th</sup> iteration step) are shown in Fig. 5.8 with dotted lines. The output  $y$  which exactly tracks the desired output transition trajectory  $\bar{y}_d$  in Eq. (5.34) is shown in Fig. 5.8 (bottom left plot) and the inverse input  $u_{inv}$  in Eqs. (5.17) and (5.69) is shown in Fig. 5.8 (top left plot). As can be seen in Fig. 5.8 (right plots), both unstable  $\eta_u$  and stable  $\eta_s$  internal states eventually reach their equilibrium final value  $\eta_{u,f}$  and  $\eta_{s,f}$  by flowing along the stable manifold during post-transition  $t \in [1, 6]$ s in O-OBR. Furthermore, smaller magnitude for both input  $u_{inv}$  and output  $y$  can be obtained from O-OBR when compared to S-OBR as shown in Fig. 5.8

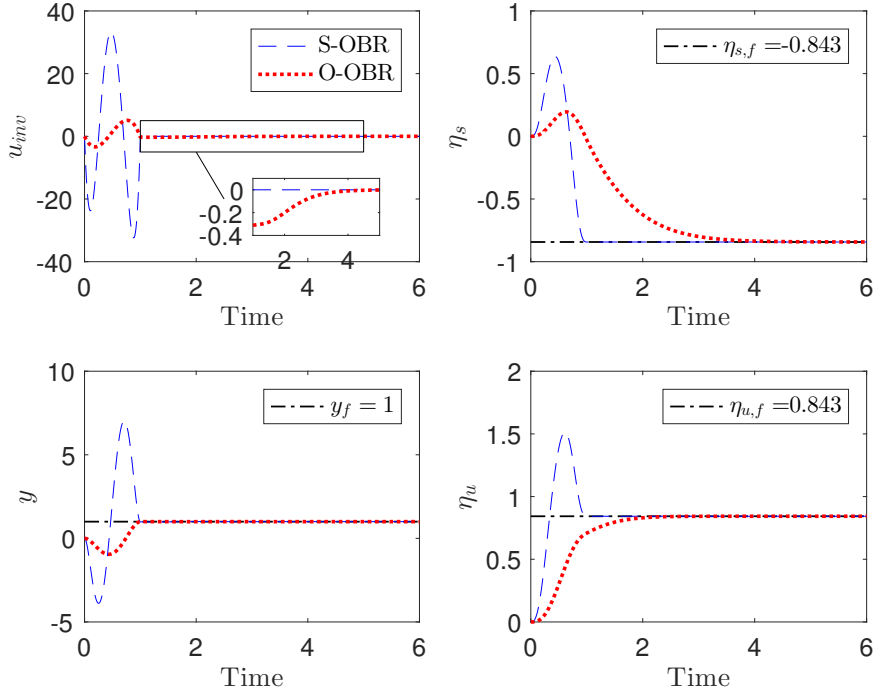


Figure 5.8: Single output transition using S-OBR and O-OBR for the coupled internal dynamics system example.

#### 5.3.4 Trajectory tracking using S-OBR and O-OBR

O-OBR can be used to track faster desired output trajectory when compared to S-OBR because relatively smaller undershoot of the output  $y_{us}$  occurs during rapid transition. The output tracking problem using S-OBR and O-OBR is compared with the desired output-trajectory  $y_d$  in Fig. 5.2 given by

$$y_d(t) = \begin{cases} \alpha_1 t + \alpha_2 & , 0 \leq t \leq (1 - \alpha_2)/\alpha_1, \\ 1 & , t > (1 - \alpha_2)/\alpha_1 \end{cases} \quad (5.120)$$

where slope of the desired trajectory is  $\alpha_1$  and a given constant  $\alpha_2$  provides the initial tracking error  $e(0)$ . For example, if

$$\alpha_2 = \bar{e} \quad (5.121)$$

where a transition trigger value  $\bar{e}$  is shown in Fig. 5.1, then

$$e(0) = y(0) - y_d(0) = 0 - \alpha_2 = -\bar{e} \quad (5.122)$$

from Eq. (5.2) and the initial output  $y(0)$  is assumed to be zero at equilibrium. The error boundary value  $\bar{E}$  in Fig. 5.1 is chosen as

$$\bar{E} = 0.1. \quad (5.123)$$

Higher maximum slope of the desired output trajectory  $\alpha_1$  in Eq. (5.120) that can be tracked provides better OBR performance.

*Effects between transition time and undershoot of the output*

Large trigger value  $\bar{e}$  and small transition time  $T_{tt}$  provide higher OBR performance by tracking higher slope of the desired output trajectory  $\alpha_1$  in Eq. (5.120). To allow large trigger value  $\bar{e}$ , undershoot of the output  $y_{us}$  needs to be small by choosing sufficiently large transition time  $T_{tt}$  as illustrated in Fig. 5.9 (left plots). Too small transition time  $T_{tt}$  causes too large undershoot of the output  $y_{us}$  and limits the change of the desired output trajectory to be smaller such that tracking error  $e$  is still inside the bounds  $e \in [-\bar{E}, \bar{E}]$ , see Fig. 5.9 (right plots). Given a trigger value  $\bar{e}$ , the transition time  $T_{tt}$  can be chosen by considering the undershoot  $y_{us}$  of the output

$$y_{us} = |\min_t\{y(t), 0\}|, \quad \forall t \in [0, T_{tt}] \quad (5.124)$$

during the first output transition from

$$y(0) = y_i = 0 \rightarrow y(T_{tt}) = y_f = \bar{e} \quad (5.125)$$

where the initial equilibrium output  $y_i$  is assumed to be zero in Eq. (5.122) and the final equilibrium output  $y_f$  is obtained from Eqs. (5.4) and (5.122)

$$y_f = y(0) - e(0) = 0 - (-\bar{e}) = \bar{e}. \quad (5.126)$$

By assuming maximum change of the desired output trajectory  $y_d$  (in a negative direction because an initial error  $e(0) = -\bar{e}$  in Eq. (5.122))

$$\inf_t\{y_d(t)\} = -\bar{e}, \quad t \in [0, T_{tt}], \quad (5.127)$$

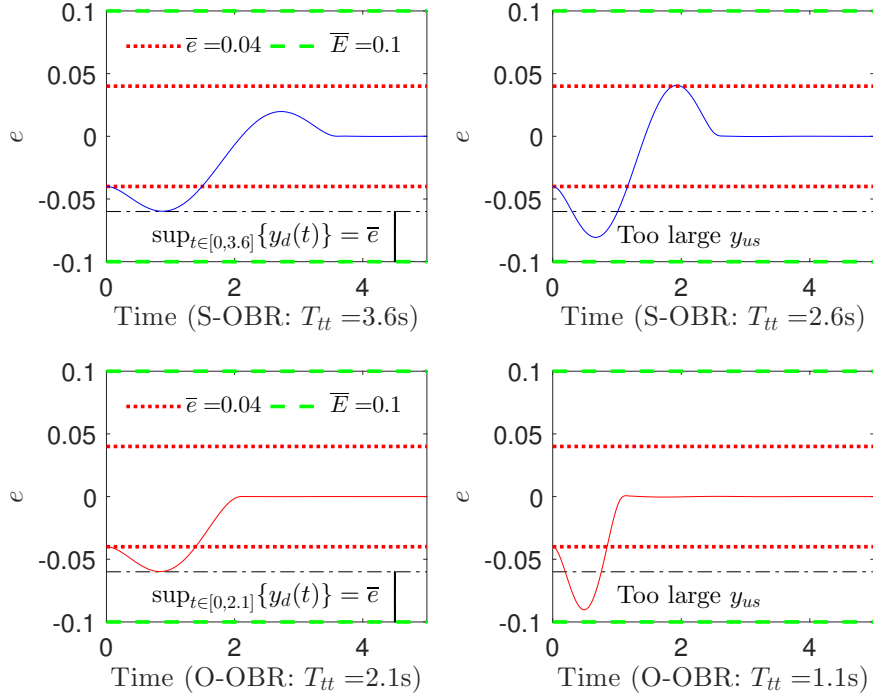


Figure 5.9: Undershoot of the output  $y_{us}$  using different transition time  $T_{tt}$  for a given trigger value  $\bar{e} = 0.04$ : (top plots) S-OBR, (bottom plots) O-OBR, (left plots) undershoot of the output  $y_{us}$  satisfies Eq. (5.129) with allowable change of the desired trajectory  $\sup_{t \in [T_i, T_f]} \{y_d(t)\} = \bar{e}$ , and (right plots) undershoot of the output  $y_{us}$  exceeds Eq. (5.129), therefore transition time  $T_{tt}$  is too small.

the allowable undershoot of the output  $\bar{y}_{us}$  ( $\geq 0$ ) can be given by

$$e(0) - \bar{y}_{us} + \inf\{y_d(t)\} = -\bar{E} \rightarrow \bar{y}_{us} = 0.1 - 2\bar{e}, \quad (5.128)$$

for  $t \in [0, T_{tt}]$  using Eqs. (5.122) and (5.127) as well as the error boundary value  $\bar{E} = 0.1$  from Eq. (5.123). Then, the transition time  $T_{tt}$  can be found by satisfying the allowable undershoot of the output  $\bar{y}_{us}$  in Eq. (5.128), i.e., the actual undershoot of the output  $y_{us}$  from Eq. (5.124) is approximately equal to the allowable undershoot of the output  $\bar{y}_{us}$  from Eq. (5.128)

$$y_{us} \approx \bar{y}_{us}. \quad (5.129)$$

For this example, the trigger value  $\bar{e}$  is chosen to be

$$\bar{e} \in [0.04, 0.4] \quad (5.130)$$

and the corresponding transition time  $T_{tt}$  that satisfies a condition in Eq. (5.129) is shown in Fig. 5.10 (top plot) where the actual undershoot of the output  $y_{us}$  for a given trigger value  $\bar{e}$  is shown in Fig. 5.10 (bottom plot) for both S-OBR and O-OBR approaches. In

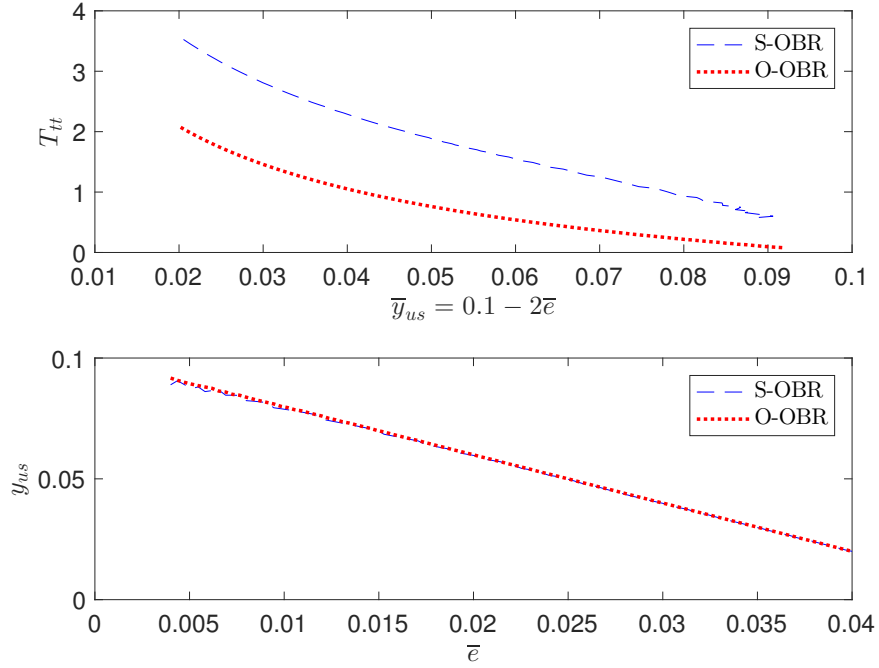


Figure 5.10: Transition time  $T_{tt}$  and undershoot of the output  $y_{us}$  for a given trigger value  $\bar{e}$ : (top) allowable undershoot  $y_{us}$  in Eq. (5.128) with the corresponding transition time  $T_{tt}$  and (bottom) actual undershoot of the output  $y_{us}$  for a given trigger value  $\bar{e}$ .

addition, the transition time  $T_{tt}$  for a given trigger value  $\bar{e}$  in Eq. (5.130) is shown in Fig. 5.11 (top plot) for both S-OBR and O-OBR approaches. As shown in Figs. 5.10 and 5.11 (top plot), small trigger time  $T_{tt}$  can be obtained from a given small trigger value  $\bar{e}$  and larger allowable undershoot of the output  $\bar{y}_{us}$  in Eq. (5.128). Thus, smaller trigger time  $T_{tt}$  is needed for faster output trajectory tracking by allowing larger undershoot of the output  $y_{us}$  with smaller trigger value  $\bar{e}$ .

#### *Comparative evaluation using S-OBR and O-OBR*

Faster output trajectory can be tracked with O-OBR when compared to S-OBR under the same bandwidth  $f_{max}$  ( $= 1/T_{tt}$ ). As shown in Fig. 5.11 (bottom plots), higher slope of the

desired trajectory  $\alpha_1$  in Eq. (5.120) can be tracked with O-OBR when compared to S-OBR at different bandwidth  $f_{max}$ . For example, by using a fixed transition time  $T_{tt} = 1$ s (bandwidth

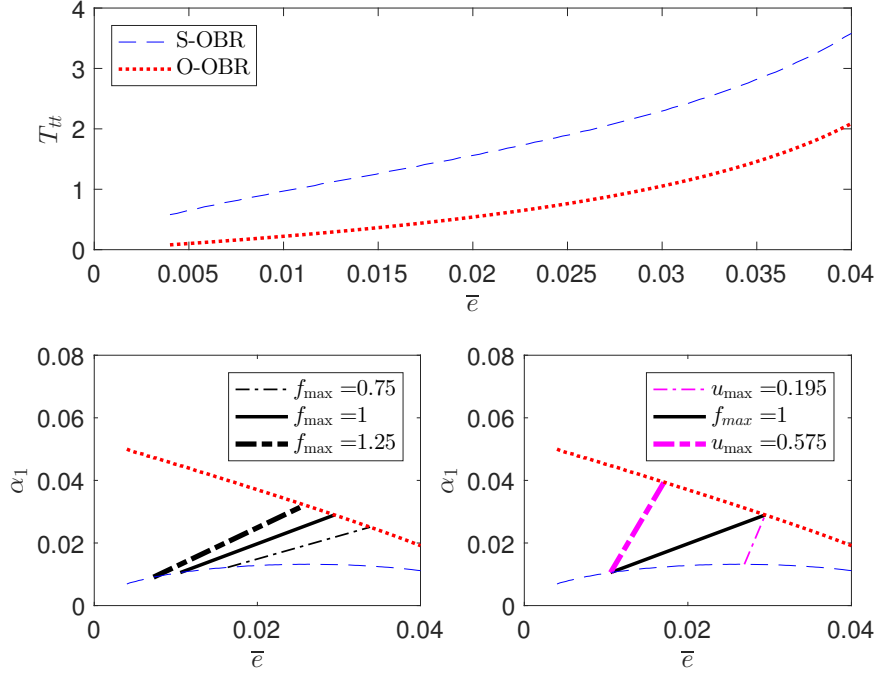


Figure 5.11: Comparative evaluation using S-OBR (dashed lines) and O-OBR (dotted lines): (top) transition time  $T_{tt}$  with a given trigger value  $\bar{e}$  and (bottom) maximum slope  $\alpha_1$  of the desired trajectory in Eq. (5.120) with varying a trigger value  $\bar{e}$ : (left) limits of bandwidth  $f_{max}$  and (right) limits of input  $u_{max}$ .

$f_{max} = 1$ Hz), O-OBR can track maximum slope of the desired trajectory  $\alpha_1 = 0.030$ , while S-OBR can track maximum slope of the desired trajectory  $\alpha_1 = 0.011$  which is 3 times lower than O-OBR. Tracking results with the fixed transition time  $T_{tt} = 1$ s are shown in Fig. 5.12. In addition, higher bandwidth  $f_{max}$  is required to track higher slope of the desired trajectory  $\alpha_1$  when using O-OBR, as shown in Fig. 5.11 (bottom left plot). For example, at bandwidth  $f_{max} = 1.25$ Hz shown in Fig. 5.11 (bottom left plot), maximum slope of the desired trajectory  $\alpha_1 = 0.032$  can be tracked with O-OBR. However, when using S-OBR, maximum slope of the desired trajectory  $\alpha_1$  drops from 0.011 with bandwidth  $f_{max} = 1$ Hz to 0.009 with bandwidth  $f_{max} = 1.25$ Hz as shown in Fig. 5.11 (bottom left plot). The lower S-OBR performance occurs because smaller trigger value  $\bar{e}$  requires too large transition

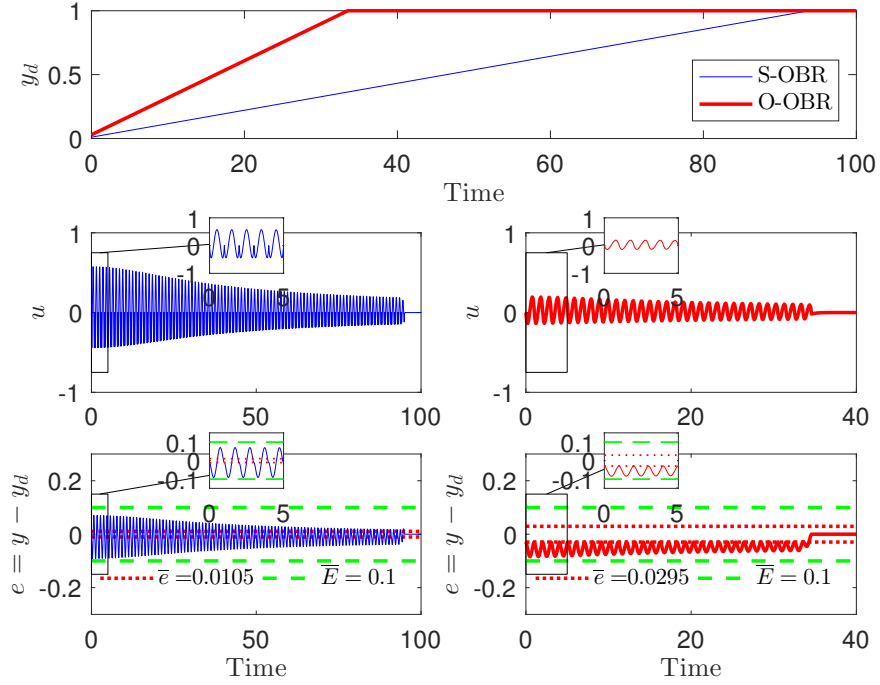


Figure 5.12: Example tracking results for the coupled internal dynamics system with a fixed transition time  $T_{tt} = 1\text{s}$  using: (left plots) S-OBR and (right plots) O-OBR.

time  $T_{tt}$  to limit the undershoot of the output  $y_{us}$  within the allowable value ( $\approx \bar{y}_{us}$  in Eq. (5.128)). In contrast, smaller trigger value  $\bar{e}$  does not limit O-OBR performance because the small transition time  $T_{tt}$  can be used with relatively smaller undershoot of the output  $y_{us}$  when compared to S-OBR. Therefore, O-OBR provides better tracking performance even using the lower bandwidth  $f_{max}$  when compared to S-OBR.

Maximum input  $u_{max}$  required when using O-OBR is also smaller than S-OBR. By using the same bandwidth  $f_{max} = 1\text{Hz}$  as shown in Fig. 5.11 (bottom right plot), S-OBR requires the maximum input  $u_{max} = 0.575$  (to track lower slope of the desired trajectory  $\alpha_1 = 0.011$ ), while O-OBR requires the maximum input  $u_{max} = 0.195$  (to track higher slope of the desired trajectory  $\alpha_1 = 0.030$ ) which is approximately 3 times smaller in input magnitude. Moreover, O-OBR can track a higher slope of the desired output trajectory  $\alpha_1 = 0.040$  which is 4 times faster than a slope of the desired output trajectory  $\alpha_1 = 0.011$  obtained from S-OBR by using the same maximum input  $u_{max} = 0.575$ , as shown in Fig. 5.11

(bottom right plot). O-OBR provides better performance by tracking faster desired output trajectory when compared to S-OBR, as expected.

This chapter presents output boundary regulation (OBR) for nonlinear nonminimum-phase systems. The proposed OBR approach can bound output-tracking errors within pre-specified values when the desired output-trajectory is not known ahead of time. Simulation results shows performance improvement by tracking 3 times higher slope of the desired output trajectory with the use of an output-transition-based OBR (O-OBR, with post-transition inputs) when compared to a state-transition-based OBR (S-OBR, without post-actuation inputs). In addition, perturbation analysis for a nonlinear system example is provided in Appendixes B and MATLAB code for this chapter is provided in Appendix C.

Chapter 6  
CONCLUSIONS

Output-boundary regulation (OBR) using event-based feedforward for nonminimum-phase systems to maintain the effects of disturbances within given boundaries when preview information of disturbances is not available was studied in this research. OBR triggers a rapid output transition when the effect of disturbances reaches pre-specified values.

A rapid output transition can be achieved by transitioning only the output  $y$  of the system rather than transitioning the entire state  $x$  of the system. The output transition can be done by transitioning the entire state from initial  $x(T_i) = x_i$  to final  $x(T_f) = x_f$  equilibrium points, i.e., a reference state  $x_{ref}$  for computing the inverse input  $u_{inv}$  can be obtained by using a standard state transition (SST) approach for linear systems as shown in Fig. 6.1 (left plot) and a state-transition-based OBR (S-OBR) approach for nonlinear

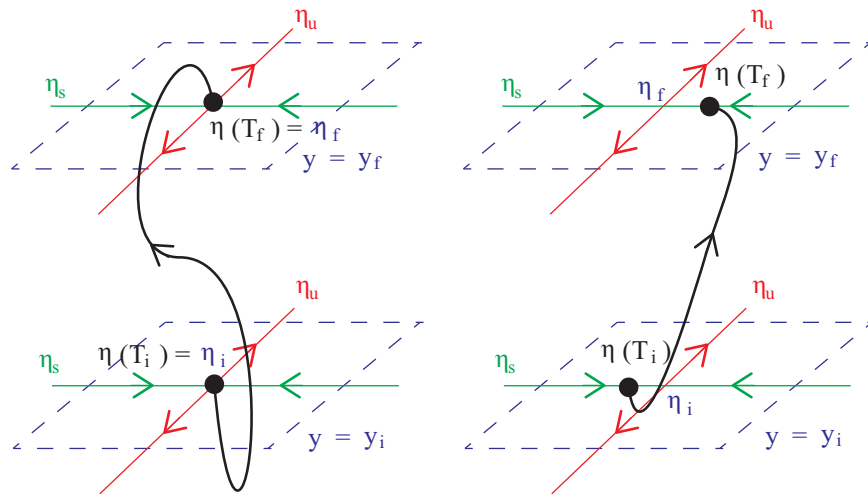


Figure 6.1: Output transition for linear systems: (left) SST and (right) OOT.

systems as shown in Fig. 6.2 (left plot). Alternatively, The output  $y$  can be completely transitioned from initial  $y(T_i) = y_i$  to final  $y(T_f) = y_f$  equilibrium positions, while the

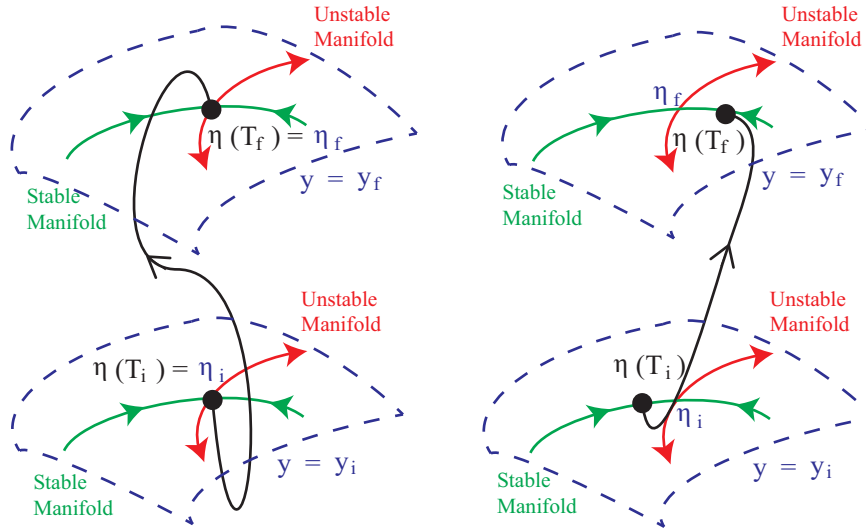


Figure 6.2: Output transition for nonlinear systems: (left) S-OBR and (right) O-OBR.

internal state  $\eta$  can be transitioned from initial  $\eta(T_f)$  (at the end of the output transition) to final  $\eta(\infty) = \eta_f$  equilibrium positions during post-actuation. This results in additional choices in an optimal output transition (OOT) for linear systems and an output-transition-based OBR (O-OBR) approach for nonlinear systems which also include the entire state transition  $x(T_f) = x_f$  in SST and S-OBR. Consequently, a smaller transition time can be achieved by allowing post-actuation inputs. Decoupled internal subsystems between stable and unstable internal dynamics of linear systems can be solved for the reference state  $x_{ref}$  from OOT by allowing boundary conditions of the internal states to be on the (decoupled) stable axis as illustrated in Fig. 6.1 (right plot). When the internal dynamics cannot be decoupled into stable and unstable subsystems for nonlinear systems, the reference state  $x_{ref}$  can be obtained from O-OBR by using the proposed polynomial-based iterative technique to obtain boundary conditions of the internal states that lie on the stable manifold as illustrated in Fig. 6.2 (right plot). As a result, faster output transition from OOT and O-OBR increases OBR performance by rejecting larger disturbance ramp-rate than SST and S-OBR.

## **6.1 Future Works**

Future works include proof of convergence for the O-OBR approach, extension of a nonlinear output transition approach, prediction of the desired output trajectory, and implementation of the proposed approach.

Convergence analysis for the O-OBR approach of nonlinear nonminimum-phase systems will be studied by providing convergence criteria for the proposed iterative-based technique in solving internal dynamics with the polynomial-based method.

A single output transition method using post-actuation inputs for nonlinear nonminimum-phase systems will be extended by comparing with a maneuver rest-to-rest problem using post-actuation inputs for linear nonminimum-phase systems studied in [45].

Extension of the proposed approach from a ramp trajectory to more general desired output trajectories will be studied for linear and nonlinear systems. This includes prediction of the desired output trajectory that may improve tracking performance by decreasing number of triggers and therefore the required bandwidth can be reduced.

Implementation of the proposed output transition in other applications such as trajectory tracking in robotics control and experimental validation are also part of ongoing efforts.

## BIBLIOGRAPHY

- [1] Garrett M Clayton, Szuchi Tien, Kam K Leang, Qingze Zou, and Santosh Devasia. A review of feedforward control approaches in nanopositioning for high speed SPM. *ASME Journal of Dynamic Systems, Measurement and Control*, 131(6):(061001) 1–19, November 2009.
- [2] Q. Zou, K. K. Leang, E. Sadoun, M. J. Reed, and S. Devasia. Control issues in high-speed AFM for biological applications: Collagen imaging example. *Special Issue on Advances in Nano-technology Control, Asian Journal Control*, 6(2):164–178, June 2004.
- [3] Toshio Ando, Takayuki Uchiashi, and Noriyuki Kodera. High-speed AFM and applications to biomolecular systems. *Annual Review of Biophysics*, 42(1), 2013.
- [4] V C Abraham, V Krishnamurthi, D L Taylor, and F Lanni. The actin-based nanomachine at the leading edge of migrating cells. *Biophysical Journal*, 77(3):1721–1732, Sept., 1999.
- [5] J. V. Small, T. Stradal, E. Vignal, and K. Rottner. The lamellipodium: where motility begins. *Trends in Cell Biology*, 12(3):112–120, March, 2002.
- [6] V.M. Laurent, S. Kasas, A. Yersin, T.E. Schaffer, S. Catsicas, G. Dietler, A.B. Verkhovsky, and J.J. Meister. Gradient of rigidity in the lamellipodia of migrating cells revealed by atomic force microscopy. *Biophysical Journal*, 89(1):667–675, July 2005.
- [7] Jason H Laks, Lucy Y Pao, and Alan D Wright. Control of wind turbines: Past, present, and future. In *American Control Conference*, St. Louis, MO, June 2009.
- [8] Lucy Y Pao and Kathryn E Johnson. Control of wind turbines. *IEEE Control Systems Magazine*, 31(2):44–62, April 2011.
- [9] John Hauser, Shankar Sastry, and George Meyer. Nonlinear control design for slightly non-minimum phase systems: application to V/STOL aircraft. *Automatica*, 28(4):665–679, July 1992.
- [10] Qingze Zou and Santosh Devasia. Precision preview-based stable-inversion for nonlinear nonminimum-phase systems: The VTOL example. *Automatica*, 43(1):117–127, January 2007.

- [11] Arom Boekfah and Santosh Devasia. Output-boundary regulation: High-speed AFM application. In *ASME Dynamic Systems and Control Conference*, Palo Alto, CA, October 2013.
- [12] Arom Boekfah and Santosh Devasia. Output-boundary regulation for nonminimum-phase systems. In *American Control Conference*, Chicago, IL, July 2015.
- [13] Arom Boekfah and Santosh Devasia. Output-boundary regulation using event-based feedforward for nonminimum-phase systems. *IEEE Transactions on Control Systems Technology*, 24(1):265 – 275, January 2016.
- [14] Arom Boekfah and Santosh Devasia. Output-boundary regulation for nonlinear nonminimum-phase systems. In *ASME Dynamic Systems and Control Conference (Submitted)*, Minneapolis, MN, October 2016.
- [15] Knut Graichen, Veit Hagenmeyer, and Michael Zeitz. A new approach to inversion-based feedforward control design for nonlinear systems. *Automatica*, 41(12):2033–2041, December 2005.
- [16] Bruce A Francis. The linear multivariable regulator problem. *SIAM Journal on Control and Optimization*, 15(3):486–505, May 1977.
- [17] Albert Isidori and Christopher Byrnes. Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(2):131–140, February 1990.
- [18] J. Freudenberg and D. Looze. Right half plane poles and zeros and design tradeoffs in feedback systems. *IEEE Transactions on Automatic Control*, 30(6):555–565, 1985.
- [19] L Qiu and E J Davison. Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica*, 29(2):337–349, March 1993.
- [20] J.B. Hoagg and D.S. Bernstein. Nonminimum-phase zeros. *IEEE Control Systems Magazine*, 27(3):45–57, 2007.
- [21] M. Hovd and R. R. Bitmead. Feedforward for stabilization. *IFAC Symposium AD-CHEM, Istanbul, Turkey*, July, 2009.
- [22] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 41(7):930–943, 1996.
- [23] H. Peng and M. Tomizuka. Preview control for vehicle lateral guidance in highway automation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 115(4):679–686, December 1993.

- [24] P.H. Meckl and R. Kinceler. Robust motion control of flexible systems using feedforward forcing functions. *IEEE Transactions on Control Systems Technology*, 2(3):245–254, September 1994.
- [25] Haiming Wang, Qingze Zou, and Hongbing Xu. Inversion-based optimal output tracking-transition switching with preview for nonminimum-phase linear systems. *Automatica*, 48(7):1364–1371, July 2012.
- [26] Karl J. Åström and Bo Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World congress*, 1999.
- [27] Anton Cervin and Karl J Åström. On limit cycles in event-based control systems. In *46th IEEE Conference on Decision and Control*, New Orleans, LA, December 2007.
- [28] Karl J. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and design of nonlinear control systems: in honor of Alberto Isidori*. Springer-Verlag, New York, NY, 2008.
- [29] Jan Lunze and Daniel Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, January 2010.
- [30] W. P. M. H. Heemels, M. C. F. Donders, and Andrew R Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, April 2013.
- [31] Manuel Beschi, Sebastián Dormido, José Sanchez, Antonio Visioli, and Luis José Yebra. Event-based PI plus feedforward control strategies for a distributed solar collector field. *IEEE Transactions on Control Systems Technology*, 23(4):1615–1622, July 2014.
- [32] Karl-Erik Arzén. A simple event-based PID controller. In *14th IFAC world congress*, Beijing, China, 1999.
- [33] W. P. M. H. Heemels, J. H. Sandee, and P. P. J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International journal of control*, 81(4):571 – 590, 2008.
- [34] Jan Lunze and Daniel Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211 – 215, 2010.
- [35] W. P. M. H. Heemels, R. J. Gorter, A. Van Zijl, P. P. J. Van Den Bosch, S. Weiland, W. H. A. Hendrix, and M. R. Vonder. Asynchronous measurement and control: a case study on motor synchronization. *Control Engineering Practice*, 7(12):1467 – 1482, 1999.

- [36] D. Lehmann and J. Lunze. Extension and experimental evaluation of an event-based state-feedback approach. *Control Engineering Practice*, 19(2):101–112, 2011.
- [37] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.
- [38] Nicolas Marchand, Sylvain Durand, and Jose Fermi Guerrero Castellanos. A general formula for event-based stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 58(5):1332–1337, 2013.
- [39] Romain Postoyan, Paulo Tabuada, Dragan Nešić, and Adolfo Anta. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 60(4):982–996, 2015.
- [40] Pavankumar Tallapragada and Nikhil Chopra. On event triggered tracking for nonlinear systems. *IEEE Transactions on Automatic Control*, 58(9):2343–2348, 2013.
- [41] Romain Postoyan, Marcos Cesar Bragagnolo, Ernest Galbrun, Jamal Daafouz, Dragan Nešić, and Eugênio B. Castelan. Event-triggered tracking control of unicycle mobile robots. *Automatica*, 52(2):302–308, 2015.
- [42] Li Qiu and EJ Davison. Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica*, 29(2):337–349, March 1993.
- [43] Aurelio Piazzzi and Antonio Visioli. Optimal inversion-based control for the set-point regulation of nonminimum-phase uncertain scalar systems. *IEEE Transactions on Automatic Control*, 46(10):1654–1659, October 2001.
- [44] Hector Perez and Santosh Devasia. Optimal output-transitions for linear systems. *Automatica*, 39(2):181–192, February 2003.
- [45] Tarunraj Singh. Pole-zero, zero-pole canceling input shapers. *ASME Journal of Dynamic Systems, Measurement, and Control*, 134(011015)(1):1–10, January 2012.
- [46] A. Tzes and S. Yurkovich. An adaptive input shaping control scheme for vibration suppression in slewing flexible structures. *IEEE Transactions on Control Systems Technology*, 1(2):114–121, June 1993.
- [47] S. Devasia. Nonlinear minimum-time control with pre- and post-actuation. *Automatica*, 47(7):1379–1387, July 2011.
- [48] Santosh Devasia. Time-optimal control with pre/post actuation for dual-stage systems. *IEEE Transactions on Control Systems Technology*, 20(2):323–334, March 2012.

- [49] Timm Faulwasser, Veit Hagenmeyer, and Rolf Findeisen. Constrained reachability and trajectory generation for flat systems. *Automatica*, 50(4):1151–1159, April 2014.
- [50] Aurelio Piazzzi and Antonio Visioli. Minimum-time system-inversion-based motion planning for residual vibration reduction. *IEEE/ASME Transactions on Mechatronics*, 5(1):12 – 22, 2000.
- [51] Knut Graichen, Michael Treuer, and Michael Zeitz. Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica*, 43(1):63–71, Jan 2007.
- [52] Haiming Wang, Qingze Zou, and Hongbing Xu. Inversion-based optimal output tracking-transition switching with preview for nonminimum-phase linear systems. *Automatica*, 48(7):1364–1371, July 2012.
- [53] Aurelio Piazzzi and Antonio Visioli. Using stable input-output inversion for minimum-time feedforward constrained regulation of scalar systems. *Automatica*, 41(2):305 – 313, 2005.
- [54] S. M Salapaka, T. De, and A. Sebastian. A robust control based solution to the sample-profile estimation problem in fast atomic force microscopy. *International Journal of Robust and Nonlinear Control*, 15(16):821–837, 2005.
- [55] YC Fung. *Biomechanics: Mechanical Properties of Living Tissues*, volume 12. Springer, 1993.
- [56] V. Vadillo-Rodriguez and J.R. Dutcher. Dynamic viscoelastic behavior of individual gram-negative bacterial cells. *Soft Matter*, 5(24):5012–5019, 2009.
- [57] GG Bilodeau. Regular pyramid punch problem. *Journal of applied mechanics*, 59:519, 1992.
- [58] SG Shroff, DR Saner, and R. Lal. Atomic-force microscopy of atrial cellslocal viscoelastic mechanical-properties and imaging of cytoskeleton. *Biophysical Journal*, 66(2):A278–A278, 1994.
- [59] EM Darling, S. Zauscher, and F. Guilak. Viscoelastic properties of zonal articular chondrocytes measured by atomic force microscopy. *Osteoarthritis and cartilage*, 14(6):571–579, 2006.
- [60] R.J. Chen, C.C.K. Lin, and M.S. Ju. Quasi-linear viscoelastic properties of pc-12 neuron-like cells measured using atomic force microscopy. *Journal of the Chinese Institute of Engineers*, 34(3):325–335, 2011.

- [61] M. Ashhab, MV Salapaka, M. Dahleh, and I. Mezić. Melnikov-based dynamical analysis of microcantilevers in scanning probe microscopy. *Nonlinear Dynamics*, 20(3):197–220, 1999.
- [62] D. Croft and S. Devasia. Vibration compensation for high speed scanning tunneling microscopy. *Review of Scientific Instruments*, 70(12):4600–4605, 1999.
- [63] S. Salapaka, A. Sebastian, J.P. Cleveland, and M.V. Salapaka. High bandwidth nanopositioner: A robust control approach. *Review of scientific instruments*, 73(9):3232–3241, 2002.
- [64] A. Isidori. *Nonlinear control systems: An introduction*, 1989.
- [65] O.M. El Rifai and K. Youcef-Toumi. In-contact dynamics of atomic force microscopes. In *Proceedings. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, volume 2, pages 1325–1328. IEEE, 2001.
- [66] PK Hansma, VB Elings, O. Marti, CE Bracker, et al. Scanning tunneling microscopy and atomic force microscopy: application to biology and technology. *Science (New York, NY)*, 242(4876):209, 1988.
- [67] C. Le Grimmelc, E. Lesniewska, M.C. Giocondi, E. Finot, V. Vié, and J.P. Goudonnet. Imaging of the surface of living cells by low-force contact-mode atomic force microscopy. *Biophysical journal*, 75(2):695–703, 1998.
- [68] H.X. You, J.M. Lau, S. Zhang, and L. Yu. Atomic force microscopy imaging of living cells: a preliminary study of the disruptive effect of the cantilever tip on cell morphology. *Ultramicroscopy*, 82(1-4):297–305, 2000.
- [69] S. Tien and S. Devasia. Rapid AFM imaging of large soft samples in liquid with small forces. *Asian Journal of Control*, 11(2):154–165, 2009.
- [70] D.R. Sahoo, A. Sebastian, and M.V. Salapaka. Harnessing the transient signals in atomic force microscopy. *International Journal of Robust and Nonlinear Control*, 15(16):805–820, 2005.
- [71] S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871, 2002.
- [72] G. Schitter, F. Allgöwer, and A. Stemmer. A new control strategy for high-speed atomic force microscopy. *Nanotechnology*, 15(1):108–114, 2004.
- [73] Q. Zou and S. Devasia. Preview-based stable-inversion for output tracking. *ASME J. of Dynamic Systems, Measurement and Control*, 121(4):625–630, Dec, 1999.

- [74] Karl J. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and design of nonlinear control systems: in honor of Alberto Isidori*. Springer-Verlag, New York, NY, 2008.
- [75] W. P. M. H. Heemels and M. C. F. Donders. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.
- [76] Narendra K Gupta. Frequency-shaped cost functionals: Extension of linear-quadratic-Gaussian design methods. *Journal of Guidance, Control, and dynamics*, 3(6):529–535, 1980.
- [77] Luca Consolini and Aurelio Piazzì. Generalized bang-bang control for feedforward constrained regulation. *Automatica*, 45(10):2234–2243, 2009.
- [78] K. J. Bathe. *Finite element procedures in engineering analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [79] Santosh Devasia. Approximated stable inversion for nonlinear systems with nonhyperbolic internal dynamics. *IEEE Transactions on Automatic Control*, 44(7):1419–1425, July 1999.
- [80] Aurelio Piazzì and Antonio Visioli. Optimal noncausal set-point regulation of scalar systems. *Automatica*, 37(1):121–127, January 2001.
- [81] Hassan K Khalil. *Nonlinear systems*. Prentice hall, Upper Saddle River, NJ, third edition, 2002.

## Appendix A

### PERTURBATION ANALYSIS FOR A LINEAR SYSTEM EXAMPLE

Effects of zero locations, maximum input and power values, noise, and sampling times for the flexible structure system in Chapter 4 are studied in this appendix.

#### **A.1 Effects of Zero Locations**

Locations of zeros affect OBR performance by decreasing or increasing undershoot  $z_{us}$  and overshoot  $z_{os}$  of the output which allows the use of larger trigger value  $\bar{y}$  and consequently reject higher disturbance ramp-rate  $\dot{D}^*$ . The plant  $G$  from (4.77) is given by

$$G(s) = \frac{(-0.0060/m_{z,u})(s + 0.951)(s - 1.051 \times m_{z,u})}{(s^2 + 0.0028s + 0.028)(s^2 + 0.0211s + 0.211)}, \quad (\text{A.1})$$

and

$$G(s) = \frac{(-0.0060/m_{z,s})(s + 0.951 \times m_{z,s})(s - 1.051)}{(s^2 + 0.0028s + 0.028)(s^2 + 0.0211s + 0.211)}, \quad (\text{A.2})$$

where the constant  $m_{z,u}$  (unstable zero) and  $m_{z,s}$  (stable zero) are used to study effects of zero locations when compared to the imaginary axis in the complex plane.

##### *A.1.1 Effects of zero locations for SST*

OBR performance when using SST without post-transition depends on both unstable and stable zeros, since both zeros have impacts on both undershoot  $z_{us}$  and overshoot  $z_{os}$  which affects trigger value  $\bar{y}$  and disturbance ramp-rate  $\dot{D}^*$ . As shown in Fig. A.1, undershoot  $z_{us}$  and overshoot  $z_{os}$  increase when zeros are located near the imaginary axis but input  $u$  for single output transition tends to become smaller, see Figs. A.2 and A.3. Effects from unstable zeros in Eq. (A.1) are shown in Fig. A.2 and from stable zeros in Eq. (A.2) are shown in Fig. A.3. Larger undershoot  $z_{us}$  and overshoot  $z_{os}$  when zeros are located near the imaginary axis limit trigger value  $\bar{y}$  to be small and therefore decrease disturbance ramp-rate  $\dot{D}^*$ .

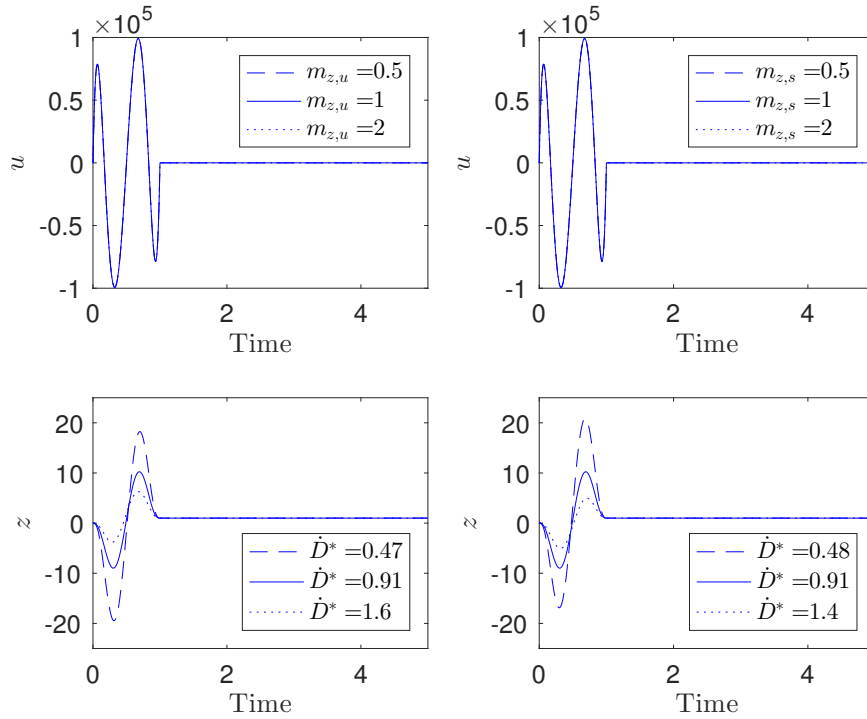


Figure A.1: Effects of zero locations when using SST: (top plots) input  $u$ , (bottom plots) output  $z$ , (left plots) varying unstable zero  $m_{z,u}$ , and (right plots) varying stable zero  $m_{z,s}$ .

### A.1.2 Effects of zero locations for OOT

Only unstable zeros have impacts on OBR performance when using OOT with post-transition by increasing or decreasing undershoot of the output  $z_{us}$  which affects disturbance ramp-rate  $\dot{D}^*$ . As shown in Fig. A.4, undershoot of the output  $z_{us}$  tends to become larger when unstable zeros are located near the imaginary axis, while stable zeros have no impact on the output  $z$ . Similar to SST, input  $u$  tends to be smaller when either unstable or stable zeros approach the imaginary axis as shown in Fig. A.5 and A.6. Disturbance ramp-rate  $\dot{D}^*$  can be higher for systems that unstable zeros are located far away from the imaginary axis due to the smaller undershoot of the output  $z_{us}$ .

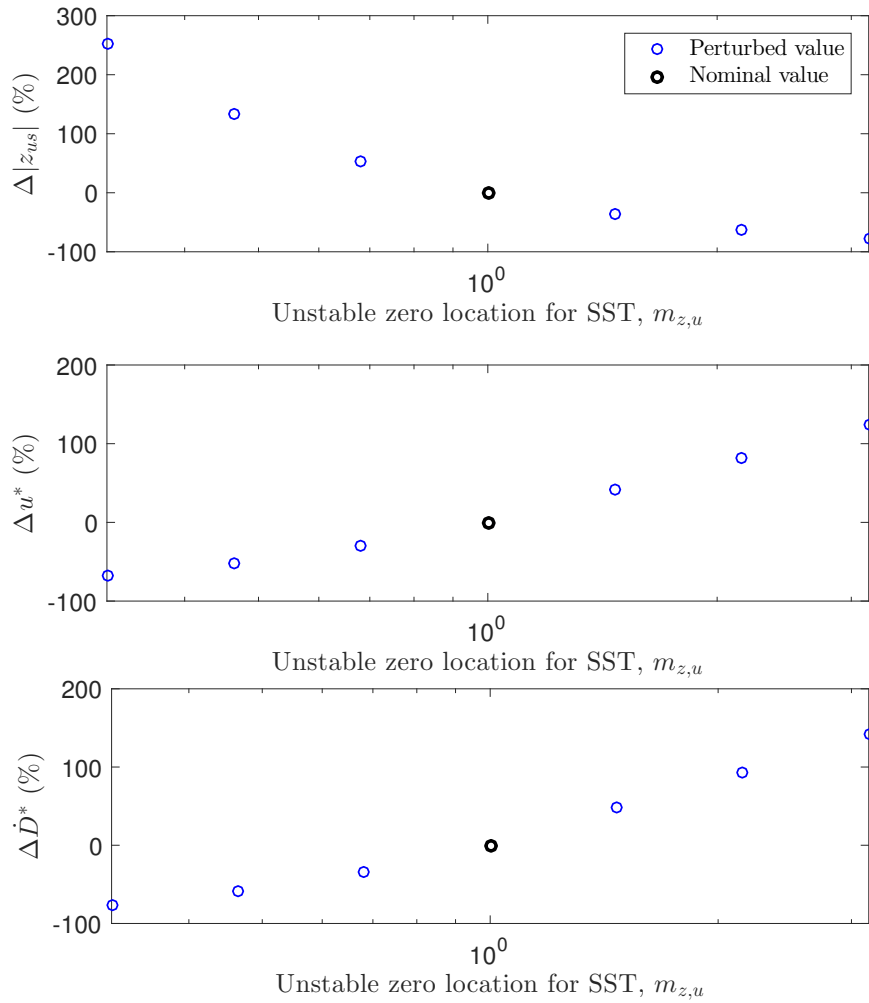


Figure A.2: Effects of unstable zero locations  $m_{z,u}$  using SST to: (top) the undershoot of the plant output  $z_{us}$ , (middle) input magnitude  $u^*$ , (bottom) the maximum disturbance ramp-rate  $\dot{D}^*$ . The trigger time  $T_{trig} = 1$  s.

## A.2 Constraints on the Input and Maximum Allowable Power Values

OBR performance depends on allowable values of input and power, since larger input and power are required to reject higher disturbance ramp-rate  $\dot{D}^*$ . Power  $P$  can be given by

$$P(t) = u(t) \times \dot{x}_1(t) \quad (\text{A.3})$$

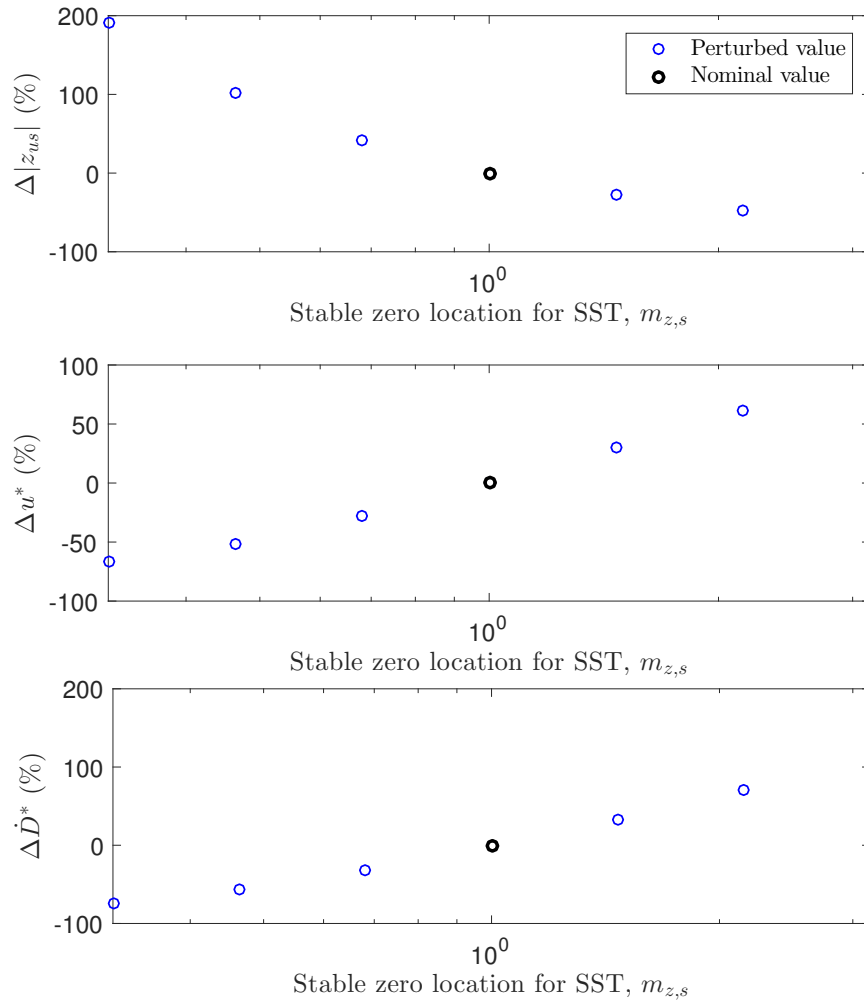


Figure A.3: Effects of stable zero locations  $m_{z,s}$  using SST to: (top) the undershoot of the plant output  $z_{us}$ , (middle) input magnitude  $u^*$ , (bottom) the maximum disturbance ramp-rate  $\dot{D}^*$ . The trigger time  $T_{trig} = 1$ s.

for  $t \geq 0$  where  $u$  is the input and  $\dot{x}_1$  is the velocity of the plant output  $z$  ( $= x_1$ ), see Fig. 4.3.

#### A.2.1 Comparison of undershoot of the output and trigger value between SST and OOT

When using SST, decreasing transition time  $T_{tt}$  substantially increases undershoot of the output  $z_{us}$  but this problem does not occur for OOT. The relatively large undershoot of the

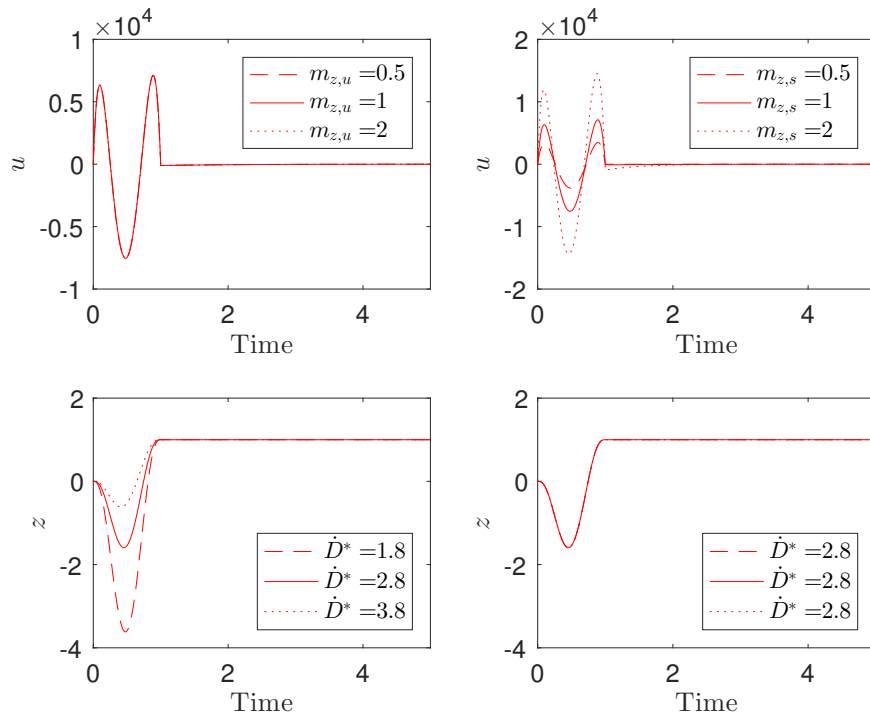


Figure A.4: Effects of zero locations when using OOT: (top plots) input  $u$ , (bottom plots) output  $z$ , (left plots) varying unstable zero  $m_{z,u}$ , and (right plots) varying stable zero  $m_{z,s}$ .

output  $z_{us}$  from using SST limits OBR performance by reducing the trigger value  $\bar{y}$  obtained from Eq. (4.57) in Lemma 2 when compared to OOT. Undershoot of the output  $z_{us}$  together with the input  $u$  between using SST and OOT is compared at different transition time  $T_{tt}$  as shown in Fig. A.7. Trigger values  $\bar{y}$  between using SST and OOT are shown in Fig. A.8 (top plot) as well as disturbance ramp-rate  $\dot{D}^*$  in Fig. A.8 (bottom plot). Larger trigger value  $\bar{y}$  can be used for OOT and therefore better OBR performance can be obtained by using small transition time  $T_{tt}$  as shown in Fig. A.8. However, OBR performance when using SST is limited even with small transition time  $T_{tt}$  because of large undershoot  $z_{us}$  and overshoot  $z_{os}$  of the output during transition.

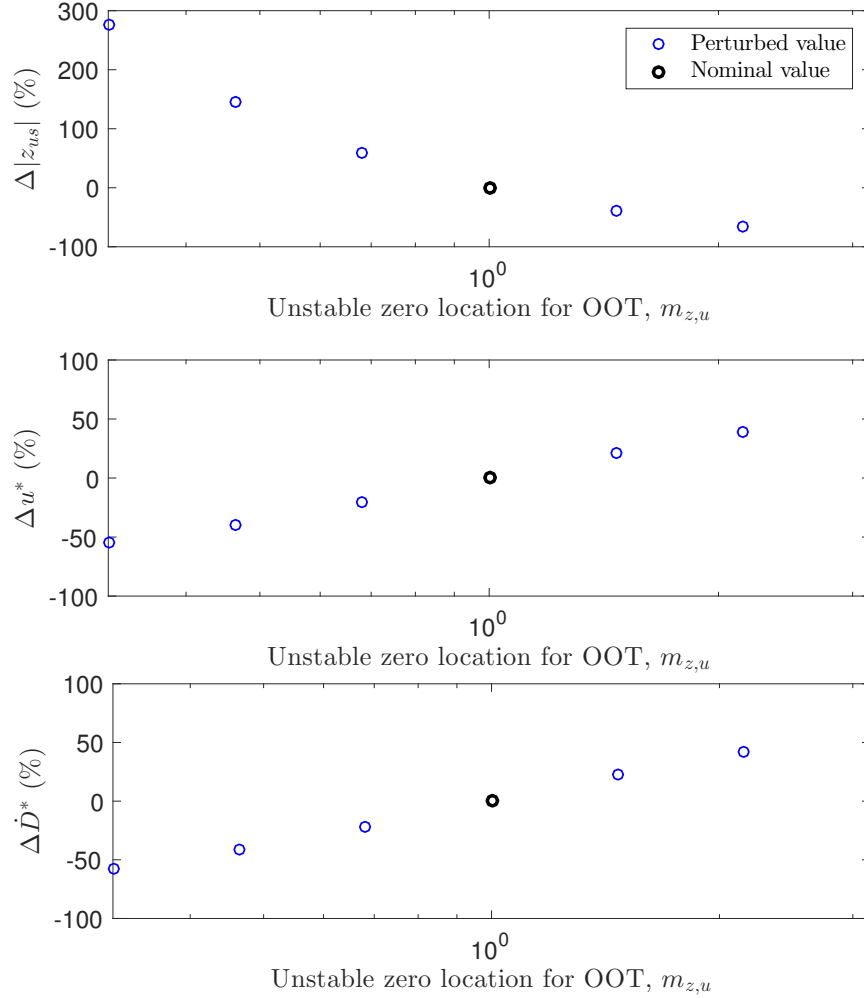


Figure A.5: Effects of unstable zero locations using OOT to: (top) the undershoot of the plant output  $z_{us}$ , (middle) input magnitude  $u^*$ , (bottom) the maximum disturbance ramp-rate  $\dot{D}^*$ . The trigger time  $T_{trig} = 1$  s.

### A.2.2 Maximum input and power required between using SST and OOT

While OBR performance with SST is limited due to large undershoot of the output  $z_{us}$  even with larger input  $u$ , better OBR performance with OOT can be achieved because of smaller undershoot of the output  $z_{us}$ . Maximum input and power between using SST and OOT are compared in Fig. A.9. Performance is limited to maximum disturbance ramp-rate  $\dot{D}^* \approx 1.2$  when using SST even with increasing input magnitude  $U_k^* = 10^5$  as shown in Fig. A.9 with

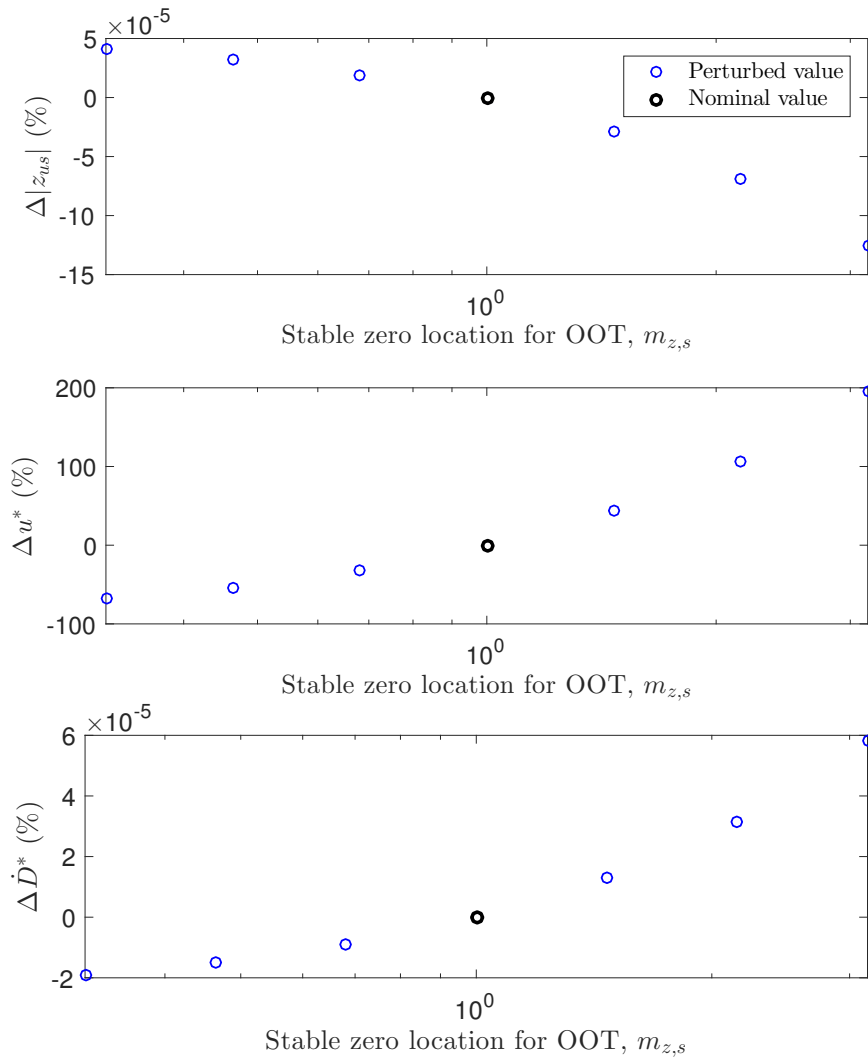


Figure A.6: Effects of stable zero locations using OOT to: (top) the undershoot of the plant output  $z_{us}$ , (middle) input magnitude  $u^*$ , (bottom) the maximum disturbance ramp-rate  $\dot{D}^*$ . The trigger time  $T_{trig} = 1$ s.

transition time  $T_{tt} = 1$ s. In contrast, the limit of maximum disturbance ramp-rate  $\dot{D}^* \approx 5$  which is 5 times higher when using OOT as shown in Fig. A.9 but larger input  $U_k^*$  and power  $P_{max}$  are required.

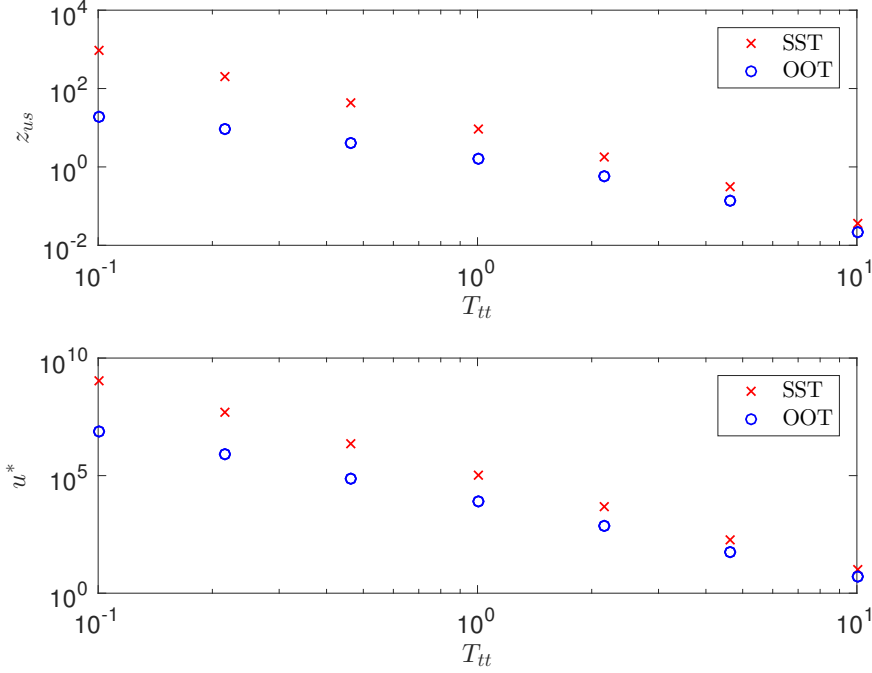


Figure A.7: Comparison of the single output transition between SST and OOT: (top) undershoot of the output  $z_{us}$  (bottom) maximum input  $u$  for single output transition.

### A.3 Effects of Noise

Noise decreases trigger value  $\bar{y}$  and therefore decreases disturbance ramp-rate  $\dot{D}^*$ . Noise  $N$  can be included as an additional disturbance when computing the trigger value  $\bar{y}$ . For example, the allowable noise  $N$  is assumed to be bounded by a trigger value  $\bar{y}$  and from Eq. (4.61)

$$\begin{aligned}
 y_{sup} &\leq \bar{y} + \sup_t \{-\bar{z}_{ff, T_{tt}}(t - t_k)\} + \sup_t \{d(t) - d(t_k)\} + \sup_t \{\mathbf{N}(t) - \mathbf{N}(t_k)\} \\
 &\leq \bar{y} + \bar{y}|z_{us}| + \bar{y} + \bar{y} = \bar{y}(3 + |z_{us}|) \leq \bar{Y},
 \end{aligned} \tag{A.4}$$

then smaller trigger value

$$\bar{y} = \frac{\bar{Y}}{3 + |z_{us}|} \tag{A.5}$$

can be used when compared to the original trigger value without noise

$$\bar{y} = \frac{\bar{Y}}{2 + |z_{us}|} \tag{A.6}$$

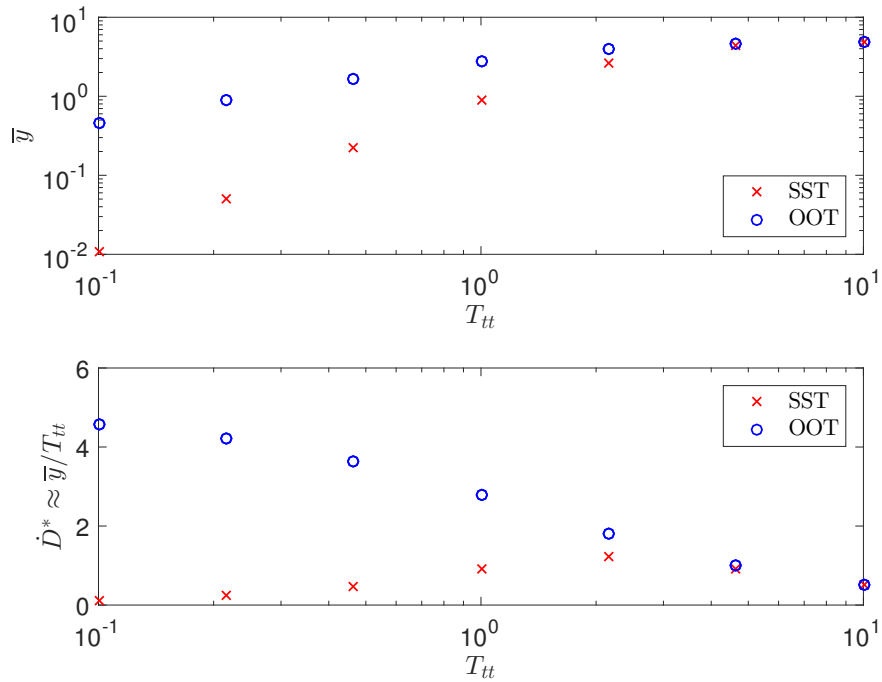


Figure A.8: Comparison of the single output transition between SST and OOT: (top) trigger value obtained from Eq. (4.57) in Lemma 2 and (bottom) estimation  $\bar{y}/T_{tt}$  of disturbance ramp-rate  $\dot{D}^*$  that can be handled between using SST and OOT.

from Eq. (4.57) in Lemma 2 by assuming zero overshoot of the output  $z_{os}$ . The noise  $N$  with different magnitude is added to the output  $y$ . As seen in Figs. A.10 and A.11, lower performance occurs with larger noise magnitude. Moreover, transitions are triggered even without external disturbance when magnitude of the noise  $N$  is too large as shown in Fig. A.10 (bottom right plot). Larger transition time  $T_{tt}$  which decreases undershoot of the output  $z_{us}$  is required to allow larger noise. The smaller trigger value  $\bar{y}$  in Eq. (A.5) due to added noise decreases OBR performance when compared to trigger value  $\bar{y}$  without noise Eq. (A.6) with the same transition time  $T_{tt}$ .

#### A.4 Effects of Sampling Time

Small sampling time is required for better OBR performance. For example, sampling time  $T_s$  needs to be smaller than the transition time  $T_{tt}$  to achieve acceptable accuracy in calculation

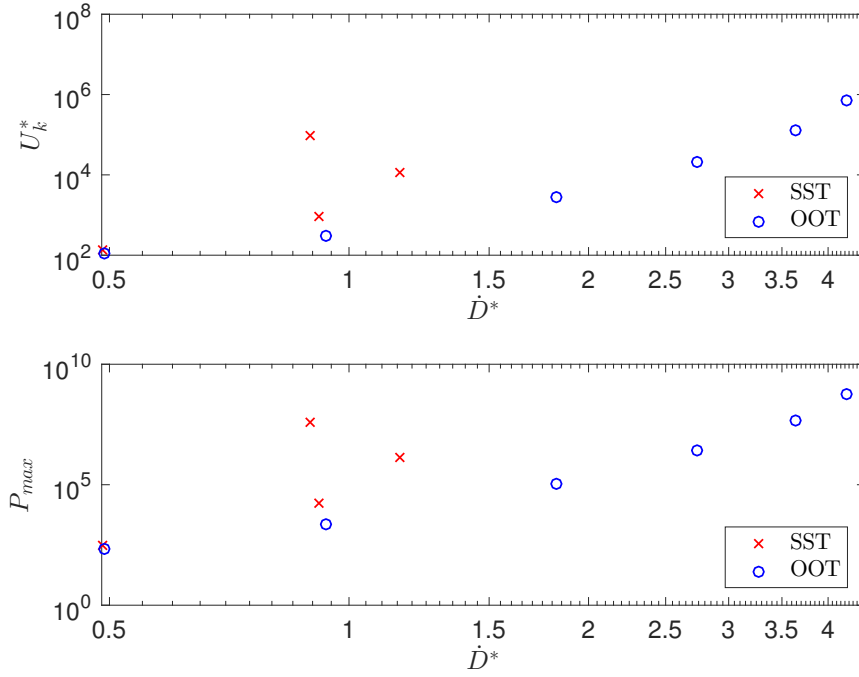


Figure A.9: Comparison of maximum input and power between using SST and OOT (top) maximum input  $U_k^*$  and (bottom) maximum power  $P_{max}$  required for different disturbance ramp rate  $\dot{D}^*$ .

and when applying the input  $u$  to the plant  $G$ . Figures A.12 and A.13 shows effects of sampling time  $T_s$  on the maximum disturbance ramp-rate  $\dot{D}^*$  when using the same trigger time  $T_{trig}$ . There is no difference on the OBR performance with too small sampling time  $T_s < 10^{-5}$ s. On the other hand, if the sampling time  $T_s$  is larger than 0.01s, the OBR performance decreases and output  $y$  cannot be maintained inside boundaries, i.e.,  $y(t) \notin [-\bar{Y}, \bar{Y}]$ , for some  $t > 0$ . Therefore, small sampling time  $T_s$  is required to get better performance.

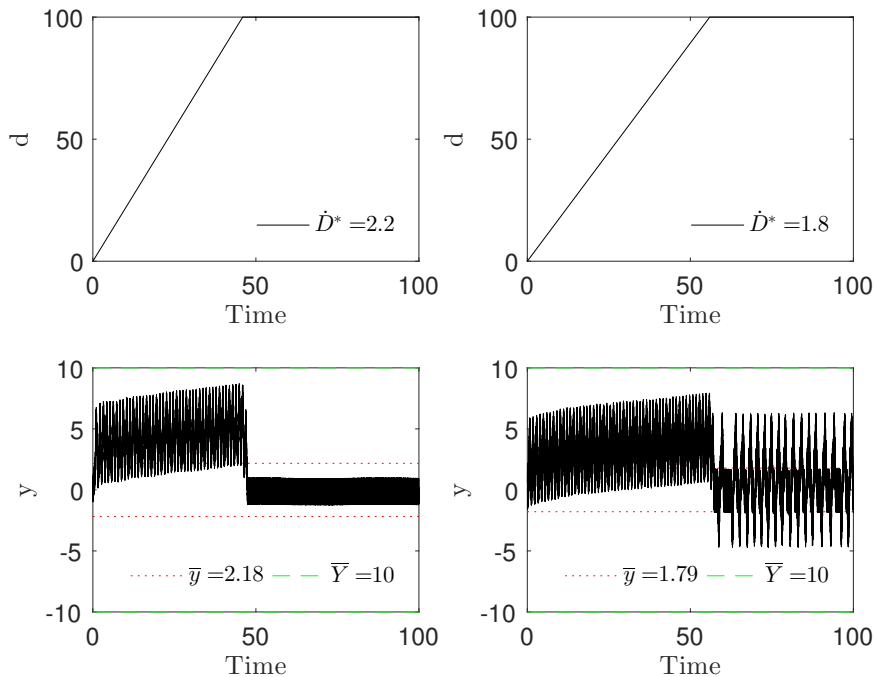


Figure A.10: Tracking results with noise  $N$  added to the output  $y$  (the trigger time  $T_{trig} = 1$ s) with magnitude of: (left) trigger value ( $1 \times \bar{y}$ ) and (right) twice trigger value ( $2 \times \bar{y}$ ).

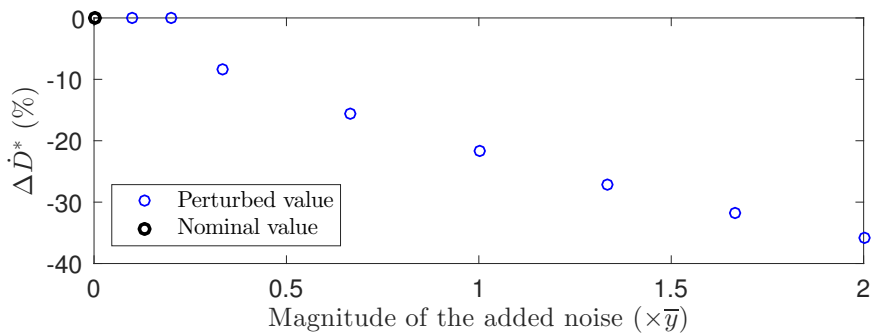


Figure A.11: Effects of noise on the disturbance ramp rate  $\dot{D}^*$  (the trigger time  $T_{trig} = 1$ s).

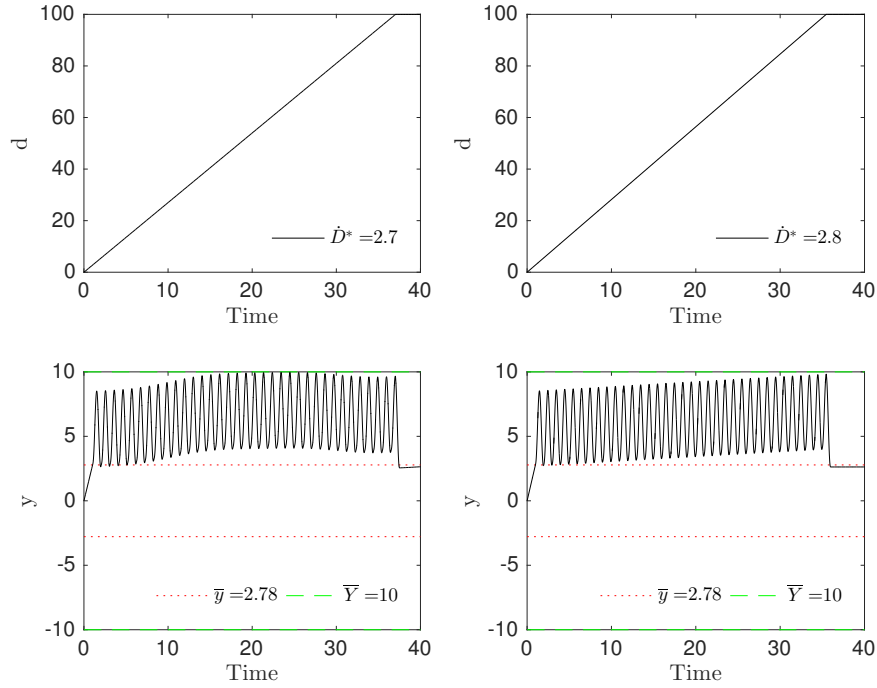


Figure A.12: Tracking results with different sampling time  $T_s$  (the trigger time  $T_{trig} = 1s$ ): (left) sampling time  $T_s = 4 \times 10^{-2}s$  and (right) sampling time  $T_s = 10^{-5}s$ .

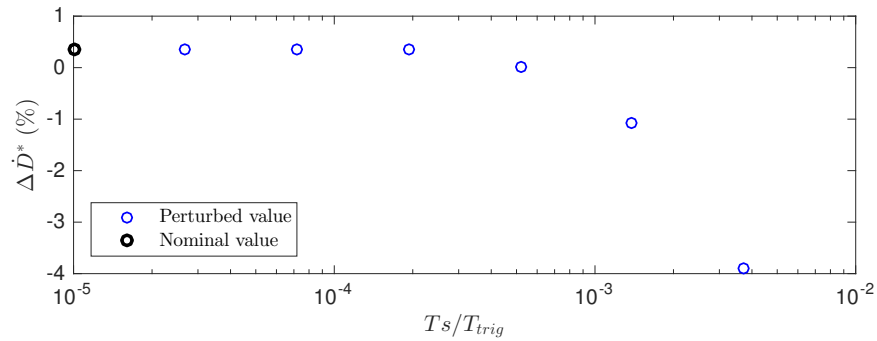


Figure A.13: Effects of sampling time  $T_s$  (the trigger time  $T_{trig} = 1s$ ) on the maximum disturbance ramp-rate  $\dot{D}^*$ .

## Appendix B

### PERTURBATION ANALYSIS FOR A NONLINEAR SYSTEM EXAMPLE

Similar to Appendix A, effects of zero locations, maximum input and power values, noise, and sampling times for the nonlinear system example in Chapter 5 are studied in this appendix.

#### ***B.1 Effects of Zero Locations***

Locations of unstable and stable zeros compared to the imaginary axis affect undershoot of the output  $y_{us}$  and therefore have impacts on tracking performance because trigger value  $\bar{y}$  can be changed. By using nonlinear system example from Eqs. (5.64) - (5.66),

$$\dot{x}_1(t) = -x_2^3(t) - x_3^3(t) + u(t) \quad (\text{B.1})$$

$$\dot{x}_2(t) = x_1(t) - m_z \times x_2(t) - x_2^3(t) + x_3^3(t) \quad (\text{B.2})$$

$$\dot{x}_3(t) = x_1(t) + m_z \times x_3(t) + x_3^3(t) - x_2^3(t) \quad (\text{B.3})$$

where constant  $m_z$  is used to vary locations of both stable and unstable zeros at the same time. As shown in Fig. B.1, smaller transition time  $T_{tt}$  can be used when zeros are located far away from the imaginary axis while undershoot of the output  $y_{us}$  is remained at  $y_{us} = 0.02$  for given trigger value  $\bar{e} = 0.04$  for both S-OBR and O-OBR. Therefore, faster slope of the desired output trajectory can be tracked for nonlinear systems that zeros are located far away from the imaginary axis because smaller undershoot of the output  $y_{us}$  increases trigger value  $\bar{y}$ .

#### ***B.2 Constraints on the Input and Maximum Allowable Power Values***

Higher slope of the desired output trajectory requires larger input and power. However, OBR performance is limited when using S-OBR even with larger allowable input and power

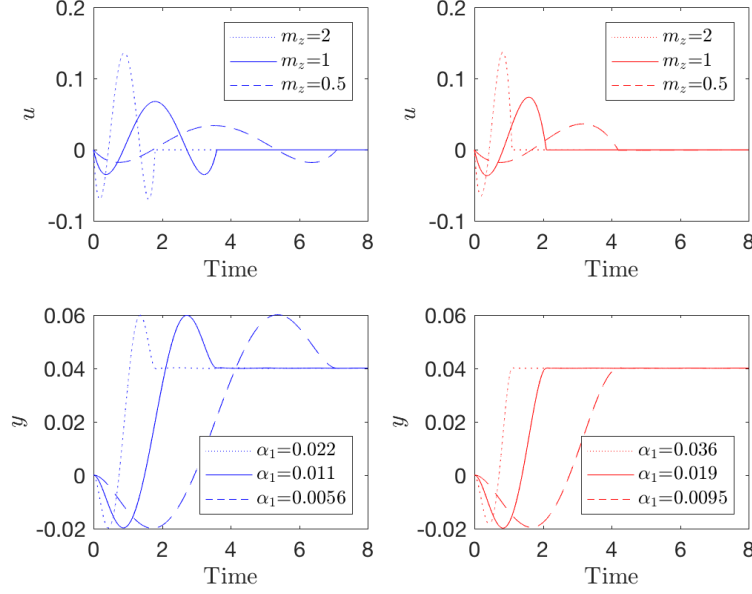


Figure B.1: Effects of zero locations on the slope of desired trajectory  $\alpha_1$ : (top plots) input  $u$ , (bottom plots) output  $y$ , (left plots) S-OBR, and (right plots) O-OBR.

as shown in Fig. B.2 because too large undershoot of the output  $y_{us}$  occurs from faster transition time  $T_{tt}$ . In contrast, O-OBR provides better tracking performance because of relatively smaller undershoot of the output  $y_{us}$  using the same limit of input and power. In addition, much faster output trajectory can be tracked by using O-OBR than S-OBR, but larger input and power are required. Therefore, O-OBR provides better performance than S-OBR because of the ability to track faster slope of the desired output trajectory with smaller input magnitude.

### B.3 Effects of Noise

Noise decreases slope of the desired trajectory because it decreases allowable undershoot of the output  $y_{us}$  and therefore increase transition time  $T_{tt}$ . Noise can be included when undershoot of the output  $y_{us}$  is computed, e.g., from Eq. (5.128)

$$\bar{y}_{us} = 0.1 - 2\bar{e} - |N|, \quad (\text{B.4})$$

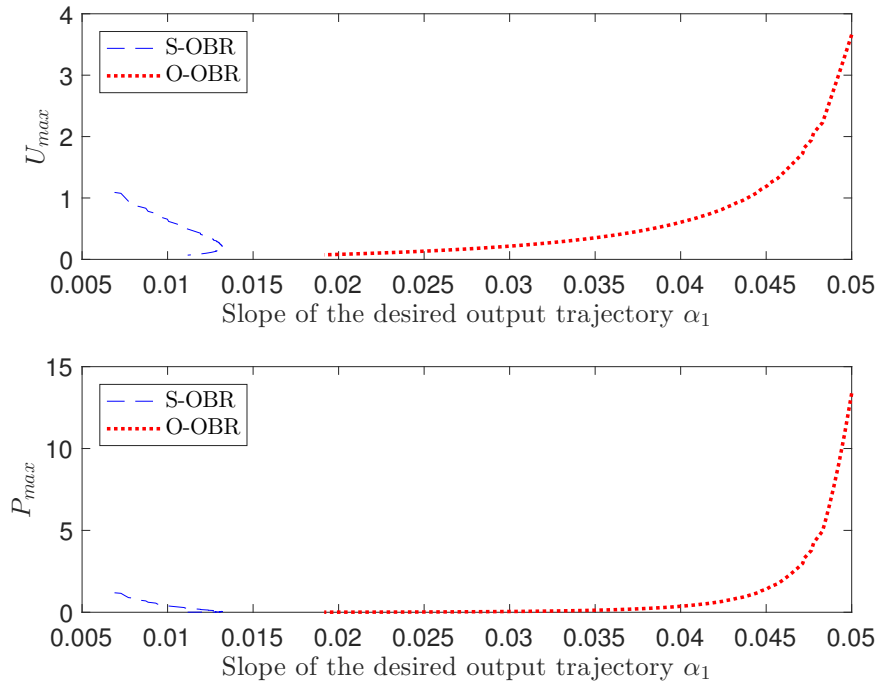


Figure B.2: Comparison of maximum input and power between using S-OBR and O-OBR (top) maximum input  $U_{max}$  and (bottom) maximum power  $P_{max}$  required for different disturbance ramp rate  $\dot{D}^*$ .

with a peak to peak magnitude of noise  $|N|$ . Tracking results with magnitude of noise

$$|N| = \frac{\bar{e}}{2} \text{ and } \bar{e} \quad (\text{B.5})$$

are shown in Fig. B.3. Higher slope of the desired output trajectory  $\alpha_1$  can be obtained if noise magnitude is small as shown in Fig. B.3 (top plots). Therefore, OBR performance is better for system with smaller noise magnitude, since larger allowable undershoot of the output  $y_{us}$  can be used with faster transition time  $T_{tt}$ .

#### B.4 Effects of Sampling Time

Better tracking performance can be achieved by using smaller sampling time. Tracking results are simulated using a fixed time step MATLAB `ode1`<sup>1</sup> function to capture a variation

---

<sup>1</sup><http://www.mathworks.com/matlabcentral/answers/98293-is-there-a-fixed-step-ordinary-differential-equation-ode-solver-in-matlab-8-0-r2012b>

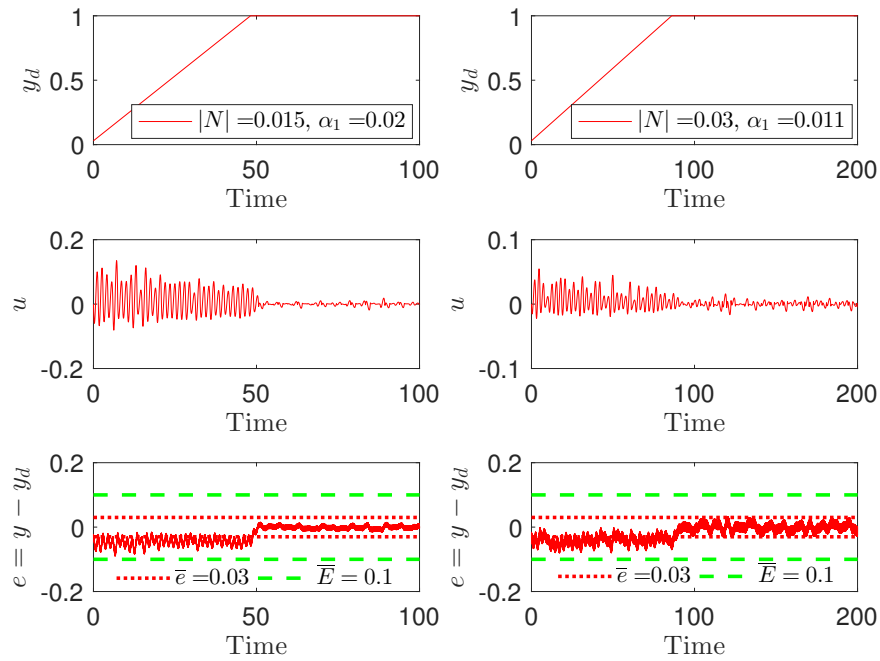


Figure B.3: Effects of noise: (left plots) smaller noise magnitude and (right plots) larger noise magnitude.

of sampling time  $\Delta t$ . However, results with sampling time  $\Delta t > 0.2\text{s}$  cannot be computed using the MATLAB `ode1` due to numerical errors. Tracking results are shown in Fig. B.4 for small  $\Delta t = 0.1\text{s}$  (left plots) and large  $\Delta t = 0.2\text{s}$  (right plots) sampling time under the same transition time  $T_{tt} = 3\text{s}$  and slope of the desired output trajectory. Based on results from the fixed time step MATLAB `ode1` function, smaller sampling time is required to get better tracking performance.

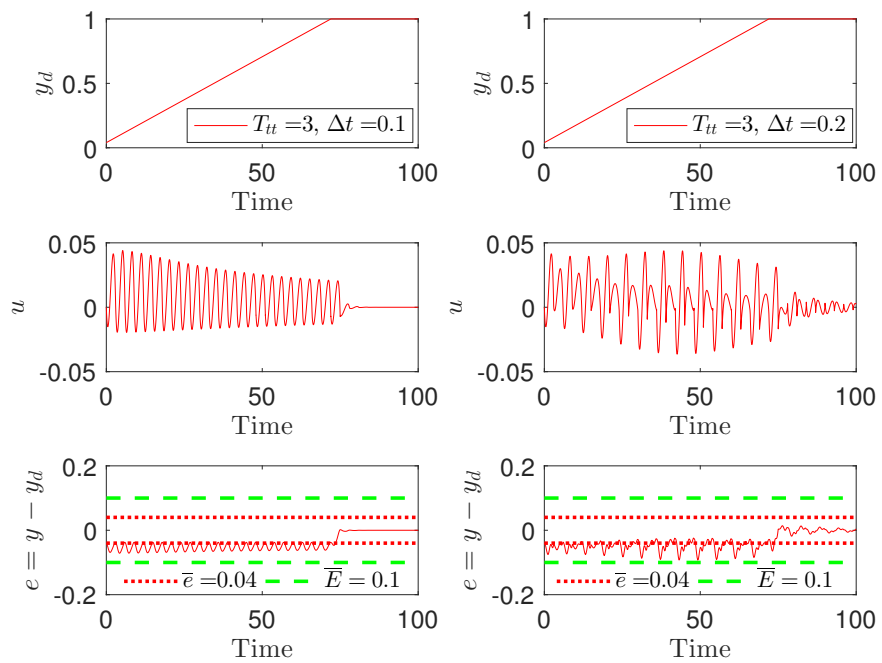


Figure B.4: Effects of sampling time using a fixed time step MATLAB `ode1` function.

## Appendix C

## ADDITIONAL INFORMATION

This appendix provides MATLAB code for a linear system example using SST and OOT in Chapter 4 and a nonlinear system example using S-OBR and O-OBR in Chapter 5.

Table C.1: Summary of the proposed approach for linear and nonlinear systems

	Linear case	Nonlinear case
<b>Given</b>	Transition time $T_{tt}$	Trigger value $\bar{e}$
<b>Single output <math>y</math> transition</b>	$y : y_i = 0 \rightarrow y_f = 1$ Eq. (4.43)	$y : y_i = 0 \rightarrow y_f = \bar{e}$ Eq. (5.34)
<b>Temporary input <math>\bar{u}</math></b>	$\bar{u}_{nom,y:0 \rightarrow 1}$ at fixed transition time $T_{tt}$ Eq. (4.42)	$\bar{u}_{y:0 \rightarrow \bar{e}}$ at different transition time $T_{tt}$ Eq. (5.35)
<b>Undershoot <math>y_{us}</math> and overshoot <math>y_{os}</math> of the output <math>y</math></b>	$y_{us}$ and $y_{os}$ Eqs. (4.44) and (4.45)	$y_{us,T_{tt}}$ and $y_{os,T_{tt}}$ Use equations from linear case
<b>Satisfying required conditions</b>	$\bar{e} = \min \left\{ \frac{\bar{E}}{2+ y_{us} }, \frac{\bar{E}}{1+ y_{os} } \right\}$ (scaling property) Eq. (4.57)	$T_{tt}$ such that $y_{us,T_{tt}} \leq \bar{E} - 2\bar{e}$ and $y_{os,T_{tt}} \leq \bar{E} - \bar{e}$ Use equations from linear case
<b>Input <math>u</math> for single output transition</b>	$u_{inv} = \bar{e} \times \bar{u}_{nom,y:0 \rightarrow 1}$	$u_{inv} = \bar{u}_{y:0 \rightarrow \bar{e},T_{tt}}$
<b>Input <math>u</math> for multiple output transitions</b>	Time-delayed version of the inverse input $\pm u_{inv}$ (superposition) Eq. (4.46)	Updating inverse input $u_{inv}$ using $y : y(T_i) \rightarrow y(T_i) \pm \bar{e}$ at fixed transition time $T_{tt}$
<b>Proposed solution for using post-actuation input during single output transition</b>	Linear optimal control with constraints on the unstable internal dynamics $\dot{\eta}_u$ Chapter 4	Two-point boundary value problem of the unstable internal dynamics $\dot{\eta}_u$ Chapter 5

### ***C.1 MATLAB Code for a Linear System Example***

This MATLAB code can be used for both SST and OOT for a flexible structure system example.

```

1 % % % MATLAB code for Chapter 4 (a flexible structure example)
2 % % % Arom Boekfah and Santosh Devasia, Output-boundary regulation using
3 % % % event-based feedforward for nonminimum-phase systems,
4 % % % IEEE Transactions on Control Systems Technology, 24(1):265-275,
5 % % % January 2016.
6
7 this_case = 1; max_t = 120; % use this line for SST
8 % this_case = 2; max_t = 40; % use this line for OOT
9
10 Yout=10;
11 Dmax=100;
12 tran=1;
13 N = 1e6;
14 delt=max_t/N;
15 t=0:delt:max_t;
16
17 % % % Problem description, two mass connected by a flexible rod
18 Ml=10*[1 0; 0 1]; Mr=1*[2 1; 1 4];
19 M=Ml+Mr;
20 Ke=1*[0 0; 0 1];
21 Ce=.1*Ke;
22 K1=[1 -1;-1 1];
23 C1=.1*K1;
24 Kr = K1+Ke;
25 Cr = C1+Ce;
26 F=[0;1];
27 A= [zeros(2,2) eye(2,2); -inv(M)*Kr -inv(M)*Cr];
28 B= [0;0; inv(M)*F];
29 C= [1 0 0 0]; D=0;
30 Sys=ss(A,B,C,D);

```

```

31
32 % % % Calculation of inverse input
33 By=C*A*B; % relative degree = 2, # of the internal dynamics = 4-2 =2
34 Ay=C*A*A;
35 Ttop=[C; C*A];
36 Tbot=[0 1 0 0; 0 0 0 1];
37 Ttb=[Ttop; Tbot];
38 Tinv=inv(Ttb);
39 TinvL=Tinv(:,1:2);
40 TinvR=Tinv(:,3:4);
41 A_inv_red=Tbot*(A-B*inv(By)*Ay)*TinvR;
42 B_inv_red=[[Tbot*(A-(B*inv(By)*Ay))*TinvL] [Tbot*B*inv(By)]];
43 C_inv_red=-inv(By)*Ay*TinvR;
44 D_inv_red=[[-inv(By)*Ay*TinvL] [inv(By)]];
45 [V,DD] = eigs(A_inv_red);
46 Tsp=V;
47 Teta=inv(Tsp)*Tbot;
48 Tpaper=[Ttop; Teta];
49 Phi=inv(Tpaper);
50 Phi_xi = Phi(:,1:2);
51 Phi_eta = Phi(:,3:4);
52 Ainv = Teta*(A-B*inv(By)*Ay)*Phi_eta;
53 Binv = Teta*[(A-B*inv(By)*Ay)*Phi_xi B*inv(By)];
54 Cinv = [-inv(By)*Ay*Phi_eta];
55 Dinv = [-inv(By)*Ay*Phi_xi inv(By)];
56
57 % % % Check: As<0 and Au>0. This part could be modified with simpler code.
58 auontop = 0; asontop = 0;
59 As=Ainv(1,1);
60 Au=Ainv(2,2);
61 Bsxi=Binv(1,1:2);
62 Bs=Binv(1,3);
63 Buxi=Binv(2,1:2);
64 Bu=Binv(2,3);
65 if As > 0
66     auontop = 1;

```

```

67     Au=Ainv(1,1);
68     As=Ainv(2,2);
69     Buxi=Binv(1,1:2);
70     Bu=Binv(1,3);
71     Bsxi=Binv(2,1:2);
72     Bs=Binv(2,3);
73 else
74     asontop = 1;
75 end
76 zbar=1;
77 calZ = [zbar;0;0];
78 eta_s_f = -inv(As)*[Bsxi Bs] * calZ;
79 eta_u_f = -inv(Au)*[Buxi Bu] * calZ;
80
81 Wf_acc=1/tran;
82 alpha_f=Wf_acc*2*pi;
83 % % % Associated reduced-dimensional state by placing a constraint on both
84 % unstable and stable internal dynamics
85 if this_case == 1 % SST
86     Ad=[0 1 0 0 0; 0 0 1 0 0; 0 0 -alpha_f 0 0; Buxi Bu Au 0; Bsxi Bs 0 As];
87     Bd=[0; 0; alpha_f; 0; 0];
88     Cd=[0 0 1 0 0]; Dd=0;
89     xti=[0;0;0;0;0];
90     xtf=[zbar;0;0;eta_u_f;eta_s_f];
91 end
92
93 % % % Associated reduced-dimensional state by placing a constraint on the
94 % unstable internal dynamics
95 if this_case == 2 % OOT
96     Ad=[0 1 0 0; 0 0 1 0; 0 0 -alpha_f 0; Buxi Bu Au]; Bd=[0; 0; alpha_f; 0];
97     Cd=[0 0 1 0]; Dd=0;
98     xti=[0;0;0;0];
99     xtf=[zbar;0;0;eta_u_f];
100 end
101
102 t_tran=0:delt:tran;

```

```

103 ti=t_tran(1); t_final=t_tran(end);
104 t1=[]; t2=t_tran; t3=max(t_tran)+delt:delt:max_t;
105 Grammian=integral(@(tau) expm(Ad*(tau))*(Bd*Bd')*expm(Ad'*(tau)),...
106     ti,t_final,'ArrayValued',true);
107 G_inv=inv(Grammian);
108 dhattitf=xtf-expm(Ad*(t_final-ti))*xti;
109 usbetween_total=zeros(1,length(t2));
110 for jj=1:length(t_tran)
111     nt=t_tran(jj);
112     usbetween=Bd'*expm(Ad'*(t_final-nt))*G_inv*dhattitf;
113     usbetween_total(jj)=usbetween;
114 end
115 usppt=usbetween_total;
116 [yref,~,xrefd]=lsim(ss(Ad,Bd,Cd,Dd),usppt,t_tran,xti); xrefd=xrefd';
117 z0d = xrefd(1,:);
118 z1d = xrefd(2,:);
119 z2d = xrefd(3,:);
120 eta_ud = xrefd(4,:);
121
122 if this_case == 1 % SST
123     eta_sd = xrefd(5,:);
124 end
125
126 cal_Z_opt = [z0d;z1d;z2d];
127 utran = lsim(ss(Ainv,Binv,Cinv,Dinv),cal_Z_opt,t_tran); utran = utran';
128 t_post = t3; n3=length(t_post);
129 eta_s_opt=lsim(ss(As,[Bsxi Bs],1,0),cal_Z_opt,t_tran);
130 eta_s_opt = eta_s_opt';
131
132 if this_case == 1 % SST
133     eta_s_post=eta_s_f*ones(1,n3); % This line is for SST.
134 end
135
136 if this_case == 2 % OOT
137     eta_s_post=zeros(1,n3);
138     for nt=1:n3

```

```

139         tnt=t_post(nt);
140         eta_s_post(1,nt) = eta_s_f + exp(As*(tnt-tran))*(eta_s_opt(end)-eta_s_f);
141     end
142 end
143
144 if asontop == 1
145     upost = Cinv*[eta_s_post; eta_u_f*ones(1,n3)] + Dinv*calZ*ones(1,n3);
146 end
147 if auontop == 1
148     upost = Cinv*[eta_u_f*ones(1,n3); eta_s_post] + Dinv*calZ*ones(1,n3);
149 end
150 u_opt = [utran upost];
151
152 % % % Apply the inverse input for single output transition to the system
153 [yopt,tt,xopt] = lsim(Sys,u_opt,t); yopt = yopt'; xopt = xopt';
154
155 [min_y,ind_y] = min(yopt); min_y=abs(min_y);
156 [max_y,ind_ymax] = max(yopt);
157 trig_val1=Yout/(2+min_y);
158 trig_val2=Yout/(1+(max_y-1));
159 trig_val = min(trig_val1,trig_val2);
160 max_degree=(trig_val/max(tran));
161
162 % % % Apply OBR to a ramp disturbance
163 degree = max_degree;
164 Peri=degree*t;
165 s=length(t); while Peri(s)>Dmax; s=s-1; end;
166 Per=[Peri(1:s) Peri(s)*ones(1,length(t)-s)];
167 ab=10;
168 ac=fix(tran/delt);
169 ff=length(t);
170 ya=0; x=[0 0 0 0]; z = 0;
171 u=zeros(size(t));
172 count=0;
173 for n=1:ab:ff-ab
174     j=n+ab;

```

```

175     Ts=0:delt:t(ab+1);
176     if count==0
177         if abs(ya(n)) >= trig_val
178             if ya(n)>0;
179                 uadd=-u_opt*trig_val;
180             else
181                 uadd=u_opt*trig_val;
182             end
183             u(n:ff)=u(n:ff)+uadd(1:ff-n+1);
184             count=ceil(ac/ab); % wait until the next trigger time.
185         end
186     else
187         count=count-1;
188     end
189     [z(n:j),ttl,x(n:j,:)]=lsim(Sys,u(n:j),Ts,x(n,:)); z(n:j)=z(n:j)';
190     ya(n:j)=z(n:j)+Per(n:j); % This is the controlled output.
191 end
192 figure
193 subplot(311)
194 plot(t,Per)
195 xlabel('Time')
196 ylabel('$d$')
197 if this_case == 1
198     title('SST')
199 end
200 if this_case == 2
201     title('OOT')
202 end
203 subplot(312)
204 plot(t,u)
205 xlabel('Time')
206 ylabel('$u$')
207 subplot(313)
208 plot(t,ya)
209 xlabel('Time')
210 ylabel('$y = z + d$')

```

## C.2 MATLAB Code for a Nonlinear System Example

MATLAB code for a single output transition using S-OBR and O-OBR and for a trajectory tracking using S-OBR and O-OBR is provided in this section.

### C.2.1 MATLAB Code for a Single Output Transition using S-OBR

```

1  % % % MATLAB code for Chapter 5 S-OBR single output transition for nonlinear systems
2  % % % s_obr_ode.m and s_obr_bc.m are required for a bvp4c function
3  % % % nl_system.m is required for an ode45 function
4  betas = 1; betau = 1;
5  Ttt = 1;
6  Kfb = [3 0 18];
7  delt = 1e-3;
8  t1 = 0:delt:Ttt;
9  t2 = 5; % post-actuation time (just to show constant state after Ttt)
10 t_tran = t1;
11 t_post = t_tran(end)+delt:delt:Ttt+t2;
12 n1 = 1;
13 n2 = length(t1);
14 y = 0; x2 = 0; x3 = 0;
15 dely = 1;
16 yhat0 = y(n1);
17 yhatT = y(n1)+dely;
18 x20 = x2(n1);
19 x30 = x3(n1);
20
21 % % % Solve the nonlinear equations for the final equilibrium states
22 options=optimset('Algorithm','Levenberg-Marquardt');
23 xx = fsolve(@(x) 2*yhatT-x(1)-(1+betas)*x(1)^3+x(2)+(1+betau)*x(2)^3 , [0;0],options);
24 x2T = xx(1);
25 x3T = xx(2);
26
27 % % % Check if all derivatives are zero at equilibrium
28 % yhatT - x2T - x2T^3 + betau*x3T^3 % Check if the x2 dot = 0

```

```

29 % yhatT + x3T + x3T^3 - betas*x2T^3 % Check if the x3 dot = 0
30 % -x2T^3-x3T^3 % Check if y dot = 0
31 % % % If betas = betau = 0, etasT and etauT must match these solutions
32 % etas0 = 0;
33 % etasTi = roots([1 0 1 -yhatT]);
34 % for jj = 1:length(etasTi)
35 %     if imag(etasTi(jj)) == 0
36 %         etasT = etasTi(jj)
37 %     end
38 % end
39 %
40 % etau0 = 0;
41 % etauTi = roots([1 0 1 yhatT]);
42 % for jj = 1:length(etauTi)
43 %     if imag(etauTi(jj)) == 0
44 %         etauT = etauTi(jj)
45 %     end
46 % end
47
48 % % % Solve the boundary value problem of the internal dynamics
49 Param = [Ttt; delt; yhat0; yhatT; x20; x2T; x30; x3T; betau; betas];
50 solinit = bvpinit(linspace(min(t_tran),max(t_tran),50), [(x2T+x20)/2; (x2T+x20)/2], [0,0]);
51 options = [];
52 % % % s_obr_ode.m and s_obr_bc.m are needed
53 sol = bvp4c(@s_obr_ode,@s_obr_bc,solinit,options,Param);
54 pp = sol.parameters
55 x23_tran = deval(sol,t1);
56 x2_tran = x23_tran(1,:);
57 x3_tran = x23_tran(2,:);
58
59 % % % Coordinate transformation to linearized decoupled internal states
60 Aeta = [-1-3*x2T^2 3*betau*x3T^2; -3*betas*x2T^2 1+3*x3T^2];
61 [Tsplitt,~]=eig(Aeta);
62 % % % x = Tsplitt*eta
63 % % % Check if As < 0 and Au > 0!!!
64 % Asplitt=inv(Tsplitt)*Aeta*Tsplitt

```

```

65 % Bsplrit=inv(Tsplrit)
66 % As=Asplrit(1,1)
67 % Bs=Bsplrit(1,:)
68 % Au=Asplrit(2,2)
69 % Bu=Bsplrit(2,:)
70 % if (As > 0) || (Au < 0) % need to check the order in Asplrit
71 %     As
72 %     Au
73 %     return;
74 % end
75 eta_tran = inv(Tsplrit)*x23_tran;
76 etas_tf = eta_tran(1,end);
77 etau_tf = eta_tran(2,end);
78
79 % % % Inverse input calculation
80 ytilde = (pp(1)+2*pp(2))*(t1/Ttt).^2 + (-2*pp(1)-3*pp(2))*(t1/Ttt).^3 +...
81     pp(1)*(t1/Ttt).^4 + pp(2)*(t1/Ttt).^5 ;
82 ytildedot = (pp(1)+2*pp(2))*(2*t1/Ttt^2) + (-2*pp(1)-3*pp(2))*(3*t1.^2/Ttt^3)...
83     + pp(1)*(4*t1.^3/Ttt^4) + pp(2)*(5*t1.^4/Ttt^5) ;
84 yhat = yhat0 + (yhatT - yhat0)*(3*(t1/Ttt).^2 - 2*(t1/Ttt).^3);
85 yhatdot = (yhatT - yhat0)*(6*(t1/Ttt^2) - 6*(t1.^2/Ttt^3));
86 ydot = yhatdot + ytildedot;
87 u_tran = ydot + x2_tran.^3 + x3_tran.^3;
88 u_post = (x2T^3 + x3T^3)*ones(size(t_post));
89 time = [t1 t_post];
90 uff = [u_tran u_post];
91
92 % % % Reference state for feedback input
93 yd = [yhat+ytilde yhatT*ones(size(t_post))];
94 x2d = [x2_tran x2T*ones(size(t_post))];
95 x3d = [x3_tran x3T*ones(size(t_post))];
96
97 % % % Apply feedforward and feedback to the system (nl_system.m is needed)
98 ts = time;
99 ics = [yhat0; x20; x30];
100 [~,xx] = ode45('nl_system',time,ics,[],Kfb,uff,ts,yd,x2d,x3d,Param); xx=xx';

```

```

101 y = xx(1, :);
102 x2 = xx(2, :);
103 x3 = xx(3, :);
104 eta = inv(Tsplit)*[x2;x3];
105 etas = eta(1, :);
106 etau = eta(2, :);
107 ufb = -Kfb*(xx-[yd;x2d;x3d]);
108 figure
109 subplot(221)
110 plot(time,uff)
111 xlabel('Time')
112 ylabel('$u$')
113 subplot(223)
114 plot(time,y)
115 xlabel('Time')
116 ylabel('$y$')
117 subplot(222)
118 plot(time,etas)
119 xlabel('Time')
120 ylabel('$\eta_s$')
121 subplot(224)
122 plot(time,etau)
123 xlabel('Time')
124 ylabel('$\eta_u$')

1 function dx23dt = s_obr_ode(t,x23,p,Param)
2 TT = Param(1); yhat0 = Param(3); yhatT = Param(4);
3 betau = Param(9); betas = Param(10);
4 yhat = yhat0 + (yhatT - yhat0)*(3*(t/TT)^2 - 2*(t/TT)^3);
5 ytilde = (p(1)+2*p(2))*(t/TT)^2 + (-2*p(1)-3*p(2))*(t/TT)^3 + ...
6     p(1)*(t/TT)^4 + p(2)*(t/TT)^5 ;
7 y = yhat + ytilde;
8 dx23dt(1) = y - x23(1) - x23(1)^3 + betau*x23(2)^3;
9 dx23dt(2) = y + x23(2) + x23(2)^3 - betas*x23(1)^3;

```

```

1 function res = s_obr_bc(x23a,x23b,~,Param)
2 x20 = Param(5); x2T = Param(6);
3 x30 = Param(7); x3T = Param(8);
4 res = [x23a(1)-x20
5         x23a(2)-x30
6         x23b(1)-x2T
7         x23b(2)-x3T];

1 function xdifff = nl_system(t_current,x,~,Kfb,uff,ts,yd,x2d,x3d,Param)
2 betau = Param(9); betas = Param(10);
3 uff=interp1(ts',uff',t_current);
4 yd=interp1(ts',yd',t_current);
5 x2d=interp1(ts',x2d',t_current);
6 x3d=interp1(ts',x3d',t_current);
7 ufb=-Kfb*(x-[yd;x2d;x3d]);
8 xdifff=zeros(3,1);
9 xdifff(1) = -x(2)^3 -x(3)^3 + uff + ufb;
10 xdifff(2) = x(1)-x(2)-x(2)^3+betau*x(3)^3;
11 xdifff(3) = x(1)+x(3)+x(3)^3-betas*x(2)^3;

```

### *C.2.2 MATLAB Code for a Single Output Transition using O-OBR*

```

1 % % % MATLAB code for Chapter 5 O-OBR single output transition for nonlinear systems
2 % % % o_obr_ode.m and o_obr_bc.m are required for a bvp4c function
3 % % % nl_system.m is required for an ode45 function
4 betas = 1; betau = 1;
5 Ttt = 1;
6 Kfb = [3 0 18]; % stable up to y* = 2.9
7 deltt = 1e-3;
8 dely = 1;
9 n1 = 1;
10 y = 0; x2 = 0; x3 = 0;
11 yhat0 = y(n1);
12 yhatT = y(n1)+dely;

```

```

13 x20 = x2(n1);
14 x30 = x3(n1);
15 % % % Solve the nonlinear equations for the final equilibrium states
16 options=optimset('Algorithm','Levenberg-Marquardt');
17 xx = fsolve(@(x) 2*yhatT-x(1)-(1+betas)*x(1)^3+x(2)+(1+betau)*x(2)^3 , [0;0],options);
18 x2T = xx(1);
19 x3T = xx(2);
20 % % % Coordinate transformation to linearized decoupled internal states
21 Aeta = [-1-3*x2T^2 3*betau*x3T^2; -3*betas*x2T^2 1+3*x3T^2];
22 Beta = 1;
23 [Tsplitt,~]=eig(Aeta);
24 % % % % x = Tsplitt*eta
25 Asplitt=inv(Tsplitt)*Aeta*Tsplitt;
26 Bsplitt=inv(Tsplitt)*Beta;
27 As=Asplitt(1,1);
28 Bs=Bsplitt(1,:);
29 Au=Asplitt(2,2);
30 Bu=Bsplitt(2,:);
31 C=1; D=0;
32 if (As > 0) || (Au < 0) % need to check the order in Asplitt
33     As
34     Au
35     return;
36 end
37 eta0 = inv(Tsplitt)*[x20;x30];
38 etas0 = eta0(1);
39 etau0 = eta0(2);
40 etaT = inv(Tsplitt)*[x2T;x3T];
41 etasT = etaT(1);
42 etauT = etaT(2);
43
44 nsteps=10; % number of iterations
45 % % % Initialized desired output trajectory and internal states
46 pp = 1;
47 t1 = 0:delt:Ttt;
48 yhat = yhat0 + (yhatT - yhat0)*(3*(t1/Ttt).^2 - 2*(t1/Ttt).^3);

```

```

49 ytilde = pp*((t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4);
50 t_tran = t1;
51 t2 = 5; % post-actuation time
52 tpost = t_tran(end)+delt:delt:Ttt+t2;
53 t = [t_tran tpost];
54 yd = [yhat+ytilde yhatT*ones(size(tpost))];
55 etau = zeros(size(yd)); etas = zeros(size(yd));
56 x2 = zeros(size(yd)); x3 = zeros(size(yd));
57 etau_total=etau; % store solution of the unstable eta at each iteration
58 etas_total=etas; % store solution of the stable eta at each iteration
59 pp_total = [];
60
61 % % % Iteration process
62 for ab=1:nsteps,
63     pp = 1;
64     ytilde = pp*((t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4);
65     yhatd = [yhat yhatT*ones(size(tpost))];
66     ytilded = [ytilde zeros(size(tpost))];
67     % % % Solve the boundary value problem of the internal dynamics
68     Param_c = [Ttt; betau; betas; yhat0; yhatT; x2T; x3T; etau0; etauT;...
69         Au; Bu(1,1); Bu(1,2)];
70     Param = [Param_c*ones(1,length(t)); t; x2; x3; yhatd; ytilded];
71     solinit = bvpinit(linspace(min(t),max(t),50), (etau0+etauT)/2,0);
72     options = [];
73     % % % o_obr_ode.m and o_obr_bc.m are needed
74     sol = bvp4c(@o_obr_ode,@o_obr_bc,solinit,options,Param);
75     pp = sol.parameters
76     iteration_steps_left = nsteps-ab
77     pp_total = [pp_total; pp];
78     % % % % Update the desired output trajectory with new p
79     ytilde = pp*((t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4);
80     yd = [yhat+ytilde yhatT*ones(size(tpost))];
81     YY=[ yd+3*x2T^2*x2-x2.^3-3*betau*x3T^2*x3+betau*x3.^3;...
82         yd-3*x3T^2*x3+x3.^3+3*betas*x2T^2*x2-betas*x2.^3];
83     % % % % Solve the unstable internal dynamics backward in time
84     Yu = YY;

```

```

85     etau_tf = -inv(Au)*Bu*Yu(:,length(Yu));
86     Yu = fliplr(Yu);
87     [etaul,~]=lsim(-Au,-Bu,C,D,Yu',t,etau_tf);
88     etau = fliplr(etaul');
89     % % % Solve the stable internal dynamics forward in time
90     Ys = YY;
91     etas_to = etas0;
92     [etas1,~]=lsim(As,Bs,C,D,Ys',t,etas_to);
93     etas = etas1';
94     etau_total=[etau_total; etau];
95     etas_total=[etas_total; etas];
96     % % % Root mean squared difference between (m)th and (m-1)th internal states
97     RMSE_u(ab) = sqrt(mean((etau_total(ab+1,:)-etau_total(ab,:)).^2));
98     RMSE_s(ab) = sqrt(mean((etas_total(ab+1,:)-etas_total(ab,:)).^2));
99
100    x23_ref=Tsplitt*[etas;etau];
101    x2=x23_ref(1,:);
102    x3=x23_ref(2,:);
103 end
104 % % % Inverse input calculation
105 yhatdot = (yhatT - yhat0)*(6*(t1/Ttt^2) - 6*(t1.^2/Ttt^3));
106 ytildedot = pp*((2*t1/Ttt^2) - (6*t1.^2/Ttt^3) + (4*t1.^3/Ttt^4));
107 ydot = [ytildedot + yhatdot zeros(size(tpost))];
108 % % % Reference state for feedback input (yd is already obtained during iteration)
109 x2d = x2;
110 x3d = x3;
111 time = t;
112 uff = ydot + x2d.^3 + x3d.^3;
113
114 % % % Apply feedforward and feedback to the system (nl_system.m is needed)
115 ts = time;
116 ics = [yhat0; x20; x30];
117 Param = [Ttt; delt; yhat0; yhatT; x20; x2T; x30; x3T; betau; betas];
118 [~,xx] = ode45('nl_system',time,ics,[],Kfb,uff,ts,yd,x2d,x3d,Param); xx=xx';
119 y = xx(1,:);
120 x2 = xx(2,:);

```

```

121 x3 = xx(3,:);
122 eta = inv(Tsplit)*[x2;x3];
123 etas = eta(1,:);
124 etau = eta(2,:);
125 ufb = -Kfb*(xx-[yd;x2d;x3d]);
126 figure
127 subplot(221)
128 plot(time,uff)
129 xlabel('Time')
130 ylabel('$u$')
131 subplot(223)
132 plot(time,y)
133 xlabel('Time')
134 ylabel('$y$')
135 subplot(222)
136 plot(time,etas)
137 xlabel('Time')
138 ylabel('$\eta_s$')
139 subplot(224)
140 plot(time,etau)
141 xlabel('Time')
142 ylabel('$\eta_u$')

1 function detau = o_obr_ode(t_current,etau,pp,Param)
2 betau = Param(2,1); betas = Param(3,1);
3 x2T = Param(6,1); x3T = Param(7,1);
4 Au = Param(10,1); Bu = Param(11:12,1)';
5 ts = Param(13,:);
6 x2s = Param(14,:); x3s = Param(15,:);
7 yhats = Param(16,:); ytildes = Param(17,:);
8 x2=interp1(ts',x2s',t_current);
9 x3=interp1(ts',x3s',t_current);
10 yhat=interp1(ts',yhats',t_current);
11 ytilde=interp1(ts',ytildes',t_current);
12 P1 = [ yhat+3*x2T^2*x2-x2.^3-3*betau*x3T^2*x3+betau*x3.^3;...

```

```

13     yhat=-3*x3T^2*x3+x3.^3+3*betas*x2T^2*x2-betas*x2.^3];
14 P2 = pp*[ytilde; ytilde];
15 Per = P1+P2;
16 detau = Au*etau + Bu*Per;

1 function res = o_obr_bc(etaua,etaub,~,Param)
2   etau0 = Param(8,1);
3   etauT = Param(9,1);
4   res = [etaua-etau0
5         etaub-etauT];

1 function xdiff = nl_system(t_current,x,~,Kfb,uff,ts,yd,x2d,x3d,Param)
2   betau = Param(9); betas = Param(10);
3   uff=interp1(ts',uff',t_current);
4   yd=interp1(ts',yd',t_current);
5   x2d=interp1(ts',x2d',t_current);
6   x3d=interp1(ts',x3d',t_current);
7   ufb=-Kfb*(x-[yd;x2d;x3d]);
8   xdiff=zeros(3,1);
9   xdiff(1) = -x(2)^3 -x(3)^3 + uff + ufb;
10  xdiff(2) = x(1)-x(2)-x(2)^3+betau*x(3)^3;
11  xdiff(3) = x(1)+x(3)+x(3)^3-betas*x(2)^3;

```

### *C.2.3 MATLAB Code for a Trajectory Tracking using S-OBR and O-OBR*

```

1 % % % Trajectory tracking using S-OBR and O-OBR
2 % % % Need functions s_obr_ode.m, s_obr_bc.m, o_obr_ode.m, o_obr_bc.m, and o_obr_fun.m
3 % % % nl_system.m is required for an ode45 function
4 this_case = 1; init_t = 3.5; % use this line for S-OBR
5 % this_case = 2; init_t = 2; % use this line for O-OBR
6 e_max = 1;
7 E_bar = .1*e_max;
8 trig_val = .04;

```

```

9 max_z_us = E_bar-2*trig_val;
10 Ttt_all = init_t:.1:5;
11 for ii = 1:length(Ttt_all)
12     Ttt = Ttt_all(ii);
13     delt = 1e-3;
14     fl = 2*round(1.25/trig_val);
15     max_t = fl*Ttt;
16     time = 0:delt:max_t;
17     err_y = ones(size(time));
18     y = zeros(size(time));
19     x2 = zeros(size(time));
20     etas = x2;
21     x3 = zeros(size(time));
22     etau = x3;
23     u = zeros(size(time));
24     ufb = zeros(size(time));
25     n_step = fix(Ttt/delt);
26     Kfb = [3 0 18];
27     % % % Desired trajectory
28     al_2 = trig_val;
29     al_1 = al_2/Ttt;
30     y_ramp_i = al_1*time + al_2;
31     s=length(time); while y_ramp_i(s)>e_max; s=s-1; end;
32     y_ramp=[y_ramp_i(1:s) y_ramp_i(s)*ones(1,length(y_ramp_i)-s)];
33     err_y(1) = y(1) - y_ramp(1);
34     for jj = 1:n_step:length(time)-n_step
35         n1 = jj;
36         n2 = n1+n_step;
37         dely = -err_y(n1);
38         t_tran = time(n1:n2) - min(time(n1));
39         t1 = t_tran; t2 = []; % post-transition is not required here
40         beta_vals = [1;1]; betas = beta_vals(1); betau = beta_vals(2);
41         yhat0 = y(n1);
42         yhatT = y(n1)+dely;
43         x20 = x2(n1);
44         x30 = x3(n1);

```

```

45 options=optimset('Algorithm','Levenberg-Marquardt','Display','off');
46 xx = fsolve(@(x) 2*yhatT-x(1)-(1+betas)*x(1)^3+x(2)+(1+betau)*x(2)^3,...
47     [0;0],options);
48 x2T = xx(1);
49 x3T = xx(2);
50 Param = [Ttt; delt; yhat0; yhatT; x20; x2T; x30; x3T; betau; betas];
51 if this_case == 1 % S-OBR
52     solinit = bvpinit(linspace(min(t_tran),max(t_tran),50),...
53         [(x2T+x20)/2;(x3T+x30)/2],[0,0]);
54     options = [];
55     % % % s_obr_ode.m and s_obr_bc.m are needed
56     sol = bvp4c(@s_obr_ode,@s_obr_bc,solinit,options,Param);
57     pp = sol.parameters;
58     x23_tran = deval(sol,t1);
59     x2_tran = x23_tran(1,:);
60     x3_tran = x23_tran(2,:);
61     ytilde = (pp(1)+2*pp(2))*(t1/Ttt).^2 + ...
62         (-2*pp(1)-3*pp(2))*(t1/Ttt).^3 + pp(1)*(t1/Ttt).^4 + ...
63         pp(2)*(t1/Ttt).^5 ; % 5th order to match 4 bcs
64     yhat = yhat0 + (yhatT - yhat0)*(3*(t1/Ttt).^2 - 2*(t1/Ttt).^3);
65     u_post = (x2T^3 + x3T^3)*ones(size(t2));
66     yd = [yhat+ytilde yhatT*ones(size(t2))];
67     x2d = [x2_tran x2T*ones(size(t2))];
68     x3d = [x3_tran x3T*ones(size(t2))];
69     ytildedot = (pp(1)+2*pp(2))*(2*t1/Ttt^2) + ...
70         (-2*pp(1)-3*pp(2))*(3*t1.^2/Ttt^3) + pp(1)*(4*t1.^3/Ttt^4)...
71         + pp(2)*(5*t1.^4/Ttt^5) ;
72     yhatdot = (yhatT - yhat0)*(6*(t1/Ttt^2) - 6*(t1.^2/Ttt^3));
73     ydot = ytildedot + yhatdot;
74     u_tran = ydot + x2_tran.^3 + x3_tran.^3;
75     uff = [u_tran u_post];
76 end
77 if this_case == 2 % O-OBR
78     % % % o_obr_fun.m is needed
79     [uff,yd,x2d,x3d] = o_obr_fun(max(t_tran),beta_vals,delt,dely,...
80         y(n1),x2(n1),x3(n1));

```

```

81         uff = uff(1:n_step+1);
82         yd = yd(1:n_step+1);
83         x2d = x2d(1:n_step+1);
84         x3d = x3d(1:n_step+1);
85     end
86     ts = t_tran;
87     ics = [y(n1); x2(n1); x3(n1)];
88     % % % Apply feedforward and feedback to the system (nl_system.m is needed)
89     [~,xx] = ode45('nl_system',t_tran,ics,[],Kfb,uff,ts,yd,x2d,x3d,Param);
90     xx=xx';
91     y(1,n1:n2) = xx(1,:);
92     x2(1,n1:n2) = xx(2,:);
93     x3(1,n1:n2) = xx(3,:);
94     yhatT = y(n1)+dely;
95     options=optimset('Algorithm','Levenberg-Marquardt','Display','off');
96     yy = fsolve(@(x) 2*yhatT-x(1)-(1+betas)*x(1)^3+x(2)+(1+betau)*x(2)^3,...
97         [0;0],options);
98     x2T = yy(1);
99     x3T = yy(2);
100    Aeta = [-1-3*x2T^2 3*betau*x3T^2; -3*betas*x2T^2 1+3*x3T^2];
101    Beta = 1;
102    [Tsplitt,~]=eig(Aeta);
103    Asplitt=inv(Tsplitt)*Aeta*Tsplitt;
104    Bsplitt=inv(Tsplitt)*Beta;
105    As=Asplitt(1,1);
106    Bs=Bsplitt(1,:);
107    Au=Asplitt(2,2);
108    Bu=Bsplitt(2,:);
109    C=1; D=0;
110    if (As > 0) || (Au < 0) % need to check the order in Asplitt
111        As
112        Au
113        return;
114    end
115    eta = inv(Tsplitt)*[x2(1,n1:n2);x3(1,n1:n2)];
116    etas(1,n1:n2) = eta(1,:);

```

```

117     etau(1,n1:n2) = eta(2,:);
118     ufb(1,n1:n2) = -Kfb*(xx-[yd;x2d;x3d]);
119     u(1,n1:n2) = uff;
120     err_y(1,n1:n2) = y(1,n1:n2) - y_ramp(1,n1:n2);
121     % % % Check if undershoot of the output is smaller than
122     % % % the allowable value or not
123     if n1 == 1
124         min_z1 = abs(min(y(1,n1:n2)));
125         if min_z1 > max_z_us
126             break;
127         end
128     end
129 end
130 % % % Transition time is obtained
131 if max(abs(err_y)) <= E_bar
132     % % % Ttt for given undershoot of the output is obtained
133     Ttt
134     figure
135     subplot(311)
136     plot(time,y_ramp)
137     ylabel('$y_d$')
138     subplot(312)
139     plot(time,u)
140     ylabel('$u$')
141     subplot(313)
142     plot(time,err_y)
143     ylabel('$e = y-y_d$')
144     xlabel('Time')
145     break;
146 end
147 end

1 function dx23dt = s_obr_ode(t,x23,p,Param)
2 TT = Param(1); yhat0 = Param(3); yhatT = Param(4);
3 betau = Param(9); betas = Param(10);

```

```

4 yhat = yhat0 + (yhatT - yhat0)*(3*(t/TT)^2 - 2*(t/TT)^3);
5 ytilde = (p(1)+2*p(2))*(t/TT)^2 + (-2*p(1)-3*p(2))*(t/TT)^3 + ...
6     p(1)*(t/TT)^4 + p(2)*(t/TT)^5 ;
7 y = yhat + ytilde;
8 dx23dt(1) = y - x23(1) - x23(1)^3 + betau*x23(2)^3;
9 dx23dt(2) = y + x23(2) + x23(2)^3 - betas*x23(1)^3;

1 function res = s_obr_bc(x23a,x23b,~,Param)
2 x20 = Param(5); x2T = Param(6);
3 x30 = Param(7); x3T = Param(8);
4 res = [x23a(1)-x20
5       x23a(2)-x30
6       x23b(1)-x2T
7       x23b(2)-x3T];

1 % % % o_obr_fun.m for trajectory tracking with O-OBR
2 % % % This code is slightly different from O-OBR for a single output transition.
3 % % % Lines 60-68 below are replaced by lines 73-86 for faster calculation.
4 function [uff,yd,x2d,x3d] = o_obr_fun(Ttt,beta_vals,delt,dely,y,x2,x3)
5 betas = beta_vals(1); betau = beta_vals(2);
6 n1 = 1;
7 yhat0 = y(n1);
8 yhatT = y(n1)+dely;
9 x20 = x2(n1);
10 x30 = x3(n1);
11 options=optimset('Algorithm','Levenberg-Marquardt','Display','off');
12 xx = fsolve(@(x) 2*yhatT-x(1)-(1+betas)*x(1)^3+x(2)+(1+betau)*x(2)^3 , [0;0],options);
13 x2T = xx(1);
14 x3T = xx(2);
15 Aeta = [-1-3*x2T^2 3*betau*x3T^2; -3*betas*x2T^2 1+3*x3T^2];
16 Beta = 1;
17 [Tsplitt,~]=eig(Aeta);
18 Asplitt=inv(Tsplitt)*Aeta*Tsplitt;
19 Bsplitt=inv(Tsplitt)*Beta;

```

```

20 As=Asplit(1,1);
21 Bs=Bsplit(1,:);
22 Au=Asplit(2,2);
23 Bu=Bsplit(2,:);
24 C=1; D=0;
25 if (As > 0) || (Au < 0) % need to check the order in Asplit
26     As
27     Au
28     return;
29 end
30 eta0 = inv(Tsplit)*[x20;x30];
31 etas0 = eta0(1);
32 etau0 = eta0(2);
33 etaT = inv(Tsplit)*[x2T;x3T];
34 etasT = etaT(1);
35 etauT = etaT(2);
36
37 nsteps=10; % number of iterations
38 % % % Initialized desired output trajectory and internal states
39 pp = 1;
40 t1 = 0:delt:Ttt;
41 yhat = yhat0 + (yhatT - yhat0)*(3*(t1/Ttt).^2 - 2*(t1/Ttt).^3);
42 ytilde = pp*(t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4;
43 t_tran = t1;
44 t2 = 5; % post-actuation time
45 tpost = t_tran(end)+delt:delt:Ttt+t2;
46 t = [t_tran tpost];
47 yd = [yhat+ytilde yhatT*ones(size(tpost))];
48 etau = zeros(size(yd)); etas = zeros(size(yd));
49 x2 = zeros(size(yd)); x3 = zeros(size(yd));
50 etau_total=etau;
51 etas_total=etas;
52 pp_total = [];
53 % % % Iteration process
54 for ab=1:nsteps,
55     pp = 1;

```

```

56 ytilde = pp*((t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4);
57
58 % % % This code is similar to O-OBR for single output transition
59 % % % However, bvp4c runs very slow in iteration process
60 % yhatd = [yhat yhatT*ones(size(tpost))];
61 % ytilded = [ytilde zeros(size(tpost))];
62 % Param_c = [Ttt; betau; betas; yhat0; yhatT; x2T; x3T; etau0; ...
63 % etauT; Au; Bu(1,1); Bu(1,2)];
64 % Param = [Param_c*ones(1,length(t)); t; x2; x3; yhatd; ytilded];
65 % solinit = bvpinit(linspace(min(t),max(t),50),(etau0+etauT)/2,0);
66 % options = [];
67 % sol = bvp4c(@o_obr_ode,@o_obr_bc,solinit,options,Param);
68 % pp = sol.parameters;
69
70 % % % The approach below is much faster than bvp4c with same results
71 % % % Solve for exact value of unknown parameter p by matching
72 % % % the initial unstable internal state
73 Yc = [ [yhat yhatT*ones(size(tpost))] + 3*x2T^2*x2 - x2.^3 - ...
74 3*betau*x3T^2*x3 + betau*x3.^3; ...
75 [yhat yhatT*ones(size(tpost))] - 3*x3T^2*x3 + x3.^3 + ...
76 3*betas*x2T^2*x2 - betas*x2.^3];
77 Yp = [[ytilde zeros(size(tpost))]; [ytilde zeros(size(tpost))]];
78 etauc_tf = -inv(Au)*Bu*Yc(:,length(Yc));
79 Yc = fliplr(Yc);
80 [etauc1,~]=lsim(-Au,-Bu,C,D,Yc',t,etauc_tf);
81 etauc = fliplr(etauc1');
82 etaup_tf = -inv(Au)*Bu*Yp(:,length(Yp));
83 Yp = fliplr(Yp);
84 [etaup1,~]=lsim(-Au,-Bu,C,D,Yp',t,etaup_tf);
85 etaup = fliplr(etaup1');
86 pp = (etau0-etauc(1))/etaup(1);
87
88 pp_total = [pp_total; pp];
89 % % % Update the desired output trajectory with new pp
90 ytilde = pp*((t1/Ttt).^2 - 2*(t1/Ttt).^3 + (t1/Ttt).^4);
91 yd = [yhat+ytilde yhatT*ones(size(tpost))];

```

```

92     YY=[ yd+3*x2T^2*x2-x2.^3-3*betau*x3T^2*x3+betau*x3.^3;...
93         yd-3*x3T^2*x3+x3.^3+3*betas*x2T^2*x2-betas*x2.^3];
94     % % % % Solve the unstable internal dynamics backward in time
95     Yu = YY;
96     etau_tf = -inv(Au)*Bu*Yu(:,length(Yu));
97     Yu = fliplr(Yu);
98     [etau1,~]=lsim(-Au,-Bu,C,D,Yu',t,etau_tf);
99     etau = fliplr(etau1');
100    % % % % Solve the stable internal dynamics forward in time
101    Ys = YY;
102    etas_to = etas0;
103    [etas1,~]=lsim(As,Bs,C,D,Ys',t,etas_to);
104    etas = etas1';
105    etau_total=[etau_total; etau];
106    etas_total=[etas_total; etas];
107    % % % % Root mean squared difference between (m)th and (m-1)th internal states
108    RMSE_u(ab) = sqrt(mean((etau_total(ab+1,:)-etau_total(ab,:)).^2));
109    RMSE_s(ab) = sqrt(mean((etas_total(ab+1,:)-etas_total(ab,:)).^2));
110    x23_ref=Tsplrit*[etas;etau];
111    x2=x23_ref(1,:);
112    x3=x23_ref(2,:);
113 end
114 % % % Inverse input calculation
115 yhatdot = (yhatT - yhat0)*(6*(t1/Ttt^2) - 6*(t1.^2/Ttt^3));
116 ytildedot = pp*((2*t1/Ttt^2) - (6*t1.^2/Ttt^3) + (4*t1.^3/Ttt^4));
117 ydot = [ytildedot + yhatdot zeros(size(tpost))];
118 % % % Reference state for feedback input (yd is already obtained during iteration)
119 x2d = x2;
120 x3d = x3;
121 uff = ydot + x2d.^3 + x3d.^3;

1 function detau = o_obr_ode(t_current,etau,pp,Param)
2     betau = Param(2,1); betas = Param(3,1);
3     x2T = Param(6,1); x3T = Param(7,1);
4     Au = Param(10,1); Bu = Param(11:12,1)';

```

```

5 ts = Param(13,:);
6 x2s = Param(14,:); x3s = Param(15,:);
7 yhats = Param(16,:); ytildes = Param(17,:);
8 x2=interp1(ts',x2s',t_current);
9 x3=interp1(ts',x3s',t_current);
10 yhat=interp1(ts',yhats',t_current);
11 ytilde=interp1(ts',ytildes',t_current);
12 P1 = [ yhat+3*x2T^2*x2-x2.^3-3*betau*x3T^2*x3+betau*x3.^3;...
13       yhat-3*x3T^2*x3+x3.^3+3*betas*x2T^2*x2-betas*x2.^3];
14 P2 = pp*[ytilde; ytilde];
15 Per = P1+P2;
16 detau = Au*etau + Bu*Per;

1 function res = o_obr_bc(etaua,etaub,~,Param)
2 eta0 = Param(8,1);
3 etaT = Param(9,1);
4 res = [etaua-eta0
5       etaub-etaT];

1 function xdiff = nl_system(t_current,x,~,Kfb,uff,ts,yd,x2d,x3d,Param)
2 betau = Param(9); betas = Param(10);
3 uff=interp1(ts',uff',t_current);
4 yd=interp1(ts',yd',t_current);
5 x2d=interp1(ts',x2d',t_current);
6 x3d=interp1(ts',x3d',t_current);
7 ufb=-Kfb*(x-[yd;x2d;x3d]);
8 xdiff=zeros(3,1);
9 xdiff(1) = -x(2)^3 -x(3)^3 + uff + ufb;
10 xdiff(2) = x(1)-x(2)-x(2)^3+betau*x(3)^3;
11 xdiff(3) = x(1)+x(3)+x(3)^3-betas*x(2)^3;

```

## VITA

Arom Boekfah is a Ph.D. student in the department of mechanical engineering at the University of Washington, Seattle, WA, USA, and is currently on leave from a Lecturer with Mahidol University, Nakhon Pathom, Thailand, since 2009.

He received the Bachelor of Engineering in mechanical engineering with the second class honors from Mahidol University, Thailand, in 2003, the Master of Engineering in mechanical engineering from Kasetsart University, Thailand, in 2005, and the Master of Science in aerospace and mechanical engineering from the University of Southern California, Los Angeles, CA, USA, in 2011. He started his Ph.D. program with the University of Washington under the support of the Royal Thai Government and Mahidol University in 2011.

In addition to an article published in IEEE Transactions on Control Systems Technology, he submitted three conference papers to ASME DSCC in 2013 which was selected as a DSCC 2013 Best Paper Award Finalist in Theory, ACC in 2015, and ASME DSCC in 2016. After his Ph.D. graduation in 2016, he will resume his position as a Lecturer with Mahidol University, Thailand.

Email: [boekfah@uw.edu](mailto:boekfah@uw.edu), [arom.boe@mahidol.ac.th](mailto:arom.boe@mahidol.ac.th).