Data-Driven Retrospective Cost Adaptive Control

by

Syed Aseem Ul Islam

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Dennis S. Bernstein, Chair
Professor Anouck Girard
Professor Ilya V. Kolmanovsky
Professor Chinedum Okwudire

The first principle is that you must not fool yourself and you are the easiest person to fool.

– Richard P. Feynman

Syed Aseem Ul Islam

aseemisl@umich.edu

ORCID iD: 0000-0001-8075-2170

I would like to thank my parents and my siblings, whose undying support allowed me to pursue my dreams. I would especially like to thank my parents, Saif and Naheed, and my siblings, Najeem, Faheem, and Rida, for supporting my goals, often at personal cost to themselves. I would like to thank my wife, Muneeza, whose love and support are a constant support, without which my journey would not be possible, and who has helped me with her technical knowledge on numerous occasions.

I would like to thank my advisor, Professor Dennis S. Bernstein, who has been instrumental in shaping the person I am today. He has taught me most of what I know academically and has taught numerous life lessons that will stay with me for the rest of my life. I hope that our relationship continues to grow and that I can continue to learn from every aspect of his being; be it work ethic, dedication to his craft, kindness, understanding, and the determination of purpose.

Finally, I would like to thank my colleagues, without whom my PhD experience would have been hollow. I hope that we continue to collaborate in the future as well.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation develops data-driven retrospective cost adaptive control (DDRCAC) and applies it to flight control. DDRCAC combines retrospective cost adaptive control (RCAC), a direct adaptive control technique for sampled-data systems, with online system identification based on recursive least squares (RLS) with variable-rate forgetting (VRF). DDRCAC uses elements of the identified model to construct the target model, which defines the retrospective performance variable. Using RLS-VRF, optimization of the retrospective performance variable updates the controller coefficients. This dissertation investigates the ability of RLS-VRF to provide the modeling information needed to construct the target model, especially nonminimum-phase (NMP) zeros, which are needed to prevent NMP-zero cancellation. A decomposition of the retrospective performance variable is derived and used to assess target-model matching and closed-loop performance. These results are illustrated by single-input, single-output (SISO) and multiple-input, multiple-output (MIMO) examples with a priori unknown dynamics. Finally, DDRCAC is applied to several simulated flight control problems, including an aircraft that transitions from minimum-phase to NMP lateral dynamics, an aircraft with flexible modes, aeroelastic wing flutter, and a nonlinear planar missile.

CHAPTER 1

Introduction

In direct adaptive control, the controller gains are updated in response to the actual dynamics of the controlled system. Unlike fixed-gain robust control, which trades off performance with prior modeling uncertainty, direct adaptive control uses partial modeling information for online self-tuning. Direct adaptive control is especially of interest for time-varying systems [1, 2]. The theory of direct adaptive control has been extensively developed [3–6], and numerous successful applications to aerospace systems have been reported [7, 8]. The research challenge in direct adaptive control is to determine the minimal modeling information needed to facilitate fast, accurate, and reliable control.

As an alternative to direct adaptive control, indirect adaptive control performs online identification to update the required modeling information for use by a fixed-gain controller [5, chapter 7], [4, pp.Chapter 397, 4671-4]. In particular, the combination of online identification and fixed-gain control is justified by the certainty equivalence principle [9, p. 2738]. Indirect adaptive control is advantageous for applications where the required modeling information is either difficult or impossible to obtain before operation due, for example, to unpredictable changes in the dynamics of the controlled system. By further reducing the dependence on prior modeling, indirect adaptive control facilitates control under extremely limited a priori modeling information. Indirect adaptive control can thus be viewed as a further step in the evolution of control from strong model dependence to model-free control.

Model-free control is a longstanding goal in control theory, and the challenges are far

from trivial. In fact, the interplay between identification and control is a longstanding problem in control theory [10–12]. This interplay is addressed by dual control, where the objective is to determine probing signals that enhance the speed and accuracy of the concurrent identification [13–15].

The goal of this dissertation is to extend RCAC by incorporating online model identification; this method is called *data-driven RCAC (DDRCAC)*. DDRCAC depends on system identification performed concurrently with controller adaptation, where the modeling details are extracted from the identified model in order to construct the target model. Since RCAC is based on recursive least squares (RLS) to update the controller coefficients, RLS is also used for system identification within DDRCAC. Unlike standard least squares, which uses constant-rate forgetting [16], online identification in the present paper takes advantage of RLS with variable-rate forgetting [17].

Note that DDRCAC updates controller coefficients using the minimization of the retrospective cost, as in RCAC, which is a direct adaptive control technique. Furthermore, DDRCAC uses a distinct online identification algorithm to learn features of the system under control, like in indirect adaptive control. However, the controller coefficients are not a function of the identified model parameters through the use of the certainty equivalence principle is *not*. Consequently, DDRCAC is neither a direct adaptive control technique nor an indirect adaptive control, and thus, can thus be viewed is a hybrid direct/indirect adaptive control method that uses online system identification to obtain approximate, limited modeling information required by a direct adaptive control algorithm.

1.1 Retrospective Cost Adaptive Control

RCAC was originally developed within the context of feedback active noise control experiments in an acoustic duct for both tonal and broadband disturbances [18]. Broadband feedback disturbance rejection using RCAC is further considered in [19–22]. In [23, 24]

connections between broadband feedback disturbance rejection using RCAC and linear-quadratic-Gaussian (LQG) control, an H_2 optimal method, are considered. The modeling information required by RCAC resides in the target model, which serves as an essential model of the closed-loop transfer function from the virtual external control perturbation to the retrospective performance variable. As shown in [24], the essential modeling information for discretized single-input, single-output (SISO) systems includes the sign of the leading numerator coefficient, the relative degree, and all nonminimum-phase (NMP) zeros. Numerical examples show that, under sufficiently aggressive tuning, RCAC may cancel unmodeled NMP zeros [25].

1.2 Retrospective Cost Variable Decomposition

In [22, 24], $G_{\mathrm{f}}(\mathbf{q})$ is analyzed as a target model for a specific closed-loop transfer function $\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})$, which is called the intercalated transfer function. To assist in analyzing the effectiveness of RCAC, and thus DDRCAC, and to obtain deeper insight into the modeling information required by the target model $G_{\mathrm{f}}(\mathbf{q})$, this dissertation shows that the retrospective performance variable can be decomposed into the sum of a performance term and a model-matching term. The performance term consists of a closed-loop transfer function, whereas the model-matching term involves the difference between a closed-loop transfer function and the target model driven by the virtual external control perturbation. A crucial insight arises from the observation that, at each step, RLS minimizes the magnitude of the retrospective performance variable by forcing the performance term and the model-matching term to have similar magnitudes but opposite signs. As the controller converges, the virtual external control perturbation, and thus the model-matching term, converges to zero, which, in turn, drives the performance term to zero. By preventing the performance term from diverging when the controller converges, this mechanism prevents RLS from converging to a controller that is destabilizing or has poor performance.

1.3 Data-Driven Retrospective Cost Adaptive Control

The goal of this dissertation is to extend RCAC by incorporating online model identification, that is, to extend RCAC from a direct adaptive control technique to a hybrid direct/indirect adaptive control method; this method is called *data-driven RCAC (DDRCAC)*. Like indirect adaptive control an attempt is made to identify the system. However, unlike indirect adaptive control, the controller coefficients are not functions of the identified model coefficients through the use of the certainty equivalence principle. Since RCAC is a direct adaptive control method, it possesses robustness to the modeling information required for the construction of the target model $G_{\mathrm{f}}(\mathbf{q})$, which is taken advantage of by DDRCAC to partially compensate for the deficiencies of concurrent closed-loop identification.

As in all applications of system identification, persistency is needed to guarantee that the identified model captures the true system dynamics [26–28]. Persistency may be provided by the commands and disturbances, or it may be self-generated by the controller. Beyond persistency, since online identification and learning occur during closed-loop operation, the control input is correlated with the measurements due to disturbances and sensor noise. When RLS is used for closed-loop identification, as in this dissertation, this correlation may obstruct *consistency*, and thus lead to asymptotic bias in the parameter estimates [29–31]. Alternative identification methods, such as instrumental variables, provide consistency despite signal correlation, albeit at higher computational cost [32].

This dissertation describes the elements of DDRCAC and investigates the effectiveness of this approach on numerical examples. These examples include synthetic examples that emphasize specific challenges as well as illustrative flight-control problems. The synthetic examples are focused on three key issues, namely, NMP zeros, consistency, and persistency. Since, as noted above, RCAC may cancel unmodeled NMP zeros, the highest priority is to extract information about the NMP zeros from the identified model; this information is embedded in the numerator of the identified model, which, in the case of a multiple-input, multiple-output (MIMO) system, is a matrix polynomial. These examples are motivated by

the fact, as noted in [8], that the stability of finite transmission zeros is a standard assumption in output-feedback adaptive control. Furthermore, since lack of consistency may occur when RLS is used for closed-loop system identification, the effect of bias is examined. In particular, the bias arising from sensor noise within closed-loop system identification under DDRCAC is shown to be less severe than the bias arising from sensor noise within closed-loop system identification under fixed-gain control. Finally, in cases where the commands and disturbances provide limited persistency, these examples highlight self-generated persistency, that is, persistency due to the controller.

1.4 Dissertation Outline

This dissertation is organized as follows.

Chapter 2 Summary

In Chapter 2, we present concepts that are used in several places throughout this dissertation. In particular, we present digital filtering and a novel extension of digital filtering to filter sequences that are also functions. Next, we present some results on feedback control of MIMO systems. Finally, we present a derivation of the recursive least squares (RLS) algorithm.

Chapter 3 Summary

In Chapter 3, we present the framework for the sampled-data control of continuous-time systems.

Chapter 4 Summary

In Chapter 4, we present the RCAC algorithm. In particular, we present a precise definition for the retrospective performance variable.

Chapter 5 Summary

In Chapter 5, we present the retrospective performance variable decomposition, which decomposes an extension of the retrospective performance variable into a one-step predicted performance term and a target-model-matching term. This decomposition provides vital insight into why RCAC converges to a controller that provides closed-loop command-following and disturbance-rejection performance in the presence of sensor noise. Finally, we explore the construction of the target model $G_{\mathrm{f}}(\mathbf{q})$ to facilitate the convergence of RCAC, and present feasibility conditions that are subsequently confirmed through numerical studies.

Chapter 6 Summary

In Chapter 6, we present simulations for MIMO adaptive control using RCAC. These simulations demonstrate that in the MIMO case RCAC may create cascade zeros, which are transmission zeros created by the cascade of two non-square systems, and subsequently cancel them with controller poles, which results in the divergence of the control signal u_k. This demonstrates the additional challenges for MIMO control using RCAC.

Chapter 7 Summary

In Chapter 7, we present an application of RCAC to the control of a model of an electrical grid. This control problem is MIMO, decentralized, and nonlinear. Furthermore, in an effort to simplify the final controller, a fixed-structure controller structure is used to pick the order and structure of SISO entries of the MIMO controller. Additionally, $G_{\mathrm{f}}(\mathbf{q})$ is constructed by a systematic process of elimination that produces a target model $G_{\mathrm{f}}(\mathbf{q})$ that is defined by two scalars. This study also demonstrates the concepts of nominal simulation tuning (NST) and perturbed simulation testing (PST). NST is the process of arriving at tuning parameters and a suitable target model using a single scenario. These choices are then

applied to several simulations that are perturbed from the nominal simulation in meaningful ways; this is PST. NST and PST provide a road map for the application of adaptive control to real-world systems.

Chapter 8 Summary

In Chapter 8, we present online identification using RLS. In particular, we investigate the efficacy of RLS for closed- and open-loop online identification. This forms the the basis for data-driven RCAC (DDRCAC).

Chapter 9 Summary

In Chapter 9, we present the DDRCAC algorithm, which consists of RLS-based identification (RLSID) and RLS-based adaptive control (RLSAC), both of which use RLS with data-dependent variable rate forgetting. Next, several numerical examples explore the applicability of DDRCAC on NMP systems. In particular, DDRCAC's robustness to tuning parameters and sensor noise is investigated. Finally, MIMO examples that failed with RCAC are revisted using DDRCAC, which demonstrate that DDRCAC does not have the tendency to create and cancel NMP cascade zeros.

Chapter 10 Summary

In Chapter 10, we apply DDRCAC to several challenging flight-control problems. First, DDRCAC is applied to roll-angle command following for a hypersonic aircraft that undergoes an unknown transition from minimum phase (MP) to NMP dynamics. This example demonstrates DDRCAC's ability to readapt to changes in the nature of the command, as well as to the unknown change from MP to NMP dynamics. Second, DDRCAC is applied for pitch-rate command following of a flexible aircraft, which has 12 lightly damped modes, which demonstrates DDRCAC's ability to converge to a controller for a high-order, lightly

damped system, with no prior knowledge of the dynamics. Third, DDRCAC is applied for flutter suppression of the benchmark active control technology (BACT) wing, which demonstrates DDRCAC's ability to stabilize a NMP, unstable system that undergoes an unknown change in dynamics. Finally, DDRCAC is applied to normal-acceleration command following for a nonlinear planar missile, which shows that DDRCAC has potential as a simulation-based offline tuning technique for assessing achievable performance without requiring explicit knowledge of the underlying equations of motion.

Chapter 11 Summary

In Chapter 11, we present the optimal predictor and filter for discrete-time real-time applications. A derivation for the one-step predictor is presented, which forms the basis for the two-step predictor, which in turn forms the basis for the two-step filter. The optimal two-step filter is more commonly known as the Kalman filter. Special emphasis is placed on the real-time implementation of these filters. Finally, we compare the accuracy of the filter and predictor numerically.

In Chapter 12, we summarize the conclusions and contributions of this dissertation, and discuss future work.

CHAPTER 2

Preliminaries

In this chapter we introduce notation and terminology for discrete-time filtering in terms of the forward-shift operator $\mathbf{q}$. In particular, we define a notion of filtering a sequence that is also a function. This notion of filtering is required for derivation and analysis of the retrospective cost variable decomposition later in this dissertation. Furthermore, we consider pole-zero cancellation in products of MIMO transfer functions that are present during control of MIMO systems. These pole-zero cancellations play an important role in the adaptive control of MIMO systems. Additionally, we present a derivation of the recursive least squares (RLS) algorithm that is fundamental to this dissertation. The material in Chapters 2.1–2.3 and 2.4 is adapted from [33] and [16], respectively.

2.1 Data Filtering

Define the proper discrete-time filter

$$G(\mathbf{q}) \triangleq D(\mathbf{q})^{-1}N(\mathbf{q}), \tag{2.1}$$

where $\mathbf{q}$ is the forward-shift operator, $N(\mathbf{q}) = N_0\mathbf{q}^n + \cdots + N_n \in \mathbb{R}[\mathbf{q}]^{p\times m}$ and $D(\mathbf{q}) = I_p\mathbf{q}^n + D_1\mathbf{q}^{n-1} + \cdots + D_n \in \mathbb{R}[\mathbf{q}]^{p\times p}$ are polynomial matrices and $\det D(\mathbf{q}) \neq 0$.

Definition 2.1. *The output* $(y_k)_{k=-n}^{\infty} \subset \mathbb{R}^p$ *of* (2.1) *with input* $(u_k)_{k=-n}^{\infty} \subset \mathbb{R}^m$ *is given by*

the data filter

$$y_k + D_1 y_{k-1} + \cdots + D_n y_{k-n} = N_0 u_k + \cdots + N_n u_{k-n}. \tag{2.2}$$

For convenience, (2.2) is written as either

$$D(\mathbf{q}) y_k = N(\mathbf{q}) u_k \tag{2.3}$$

or

$$y_k = G(\mathbf{q}) u_k. \tag{2.4}$$

Example 1. *Data filtering.* Let $N(\mathbf{q}) = 2\mathbf{q} + 3$ and $D(\mathbf{q}) = \mathbf{q}^2 + 4\mathbf{q} + 5$, which yields the input-output difference equation

$$y_k = -4y_{k-1} - 5y_{k-2} + 2u_{k-1} + 3u_{k-2}. \tag{2.5}$$

With the data $(u_k)_{k=-2}^{0} = (6, 7, 8)$ and $(y_k)_{k=-2}^{-1} = (10, 11)$, (2.5) yields

$$y_0 = -4y_{-1} - 5y_{-2} + 2u_{-1} + 3u_{-2} = -62, \tag{2.6}$$

$$y_1 = -4y_0 - 5y_{-1} + 2u_0 + 3u_{-1} = 230. \tag{2.7}$$

$\diamond$

2.2 Frozen-input-argument (FIA) Filtering

Definition 2.1 is now extended to the case where the input u_k is a function of an independent variable x_k.

Definition 2.2. *Let* $D_1, \dots D_n \in \mathbb{R}^{p\times p}$, *let* $N_0, \dots N_n \in \mathbb{R}^{p\times m}$, *let* $y_{k-n}, \dots, y_{-1} \in \mathbb{R}^p$ *be initial output data, let* $(x_k)_{k=-n}^{\infty} \subset \mathbb{R}^r$, *and, for all* $k \geq -n$, *let* $u_k \colon \mathbb{R}^r \to \mathbb{R}^m$. *Then, the* FIA sequence $(y_k(x_k))_{k=0}^{\infty}$ *is given by the* fixed-input-argument (FIA) filter

$$y_k(x_k) + D_1 y_{k-1}(x_{k-1}) + \cdots + D_n y_{k-n}(x_{k-n}) = N_0 u_k(x_k) + \cdots + N_n u_{k-n}(x_k), \quad (2.8)$$

where, for all $k \in [-n, -1]$, $y_k(x_k) \triangleq y_k$.

Note that, at each step k, the arguments of $u_{k-n}, \dots, u_k$ in (2.8) are fixed at the current input value x_k over the interval $[k-n, k]$. In contrast, the left-hand side defines the current output $y_k(x_k)$, which depends on the past output values $y_{k-n}(x_{k-n}), \dots, y_{k-1}(x_{k-1})$. For convenience, (2.8) is written as either

$$D(\mathbf{q})y_k(x_k) = N(\mathbf{q})u_k(x_{\overline{k}}) \quad (2.9)$$

or

$$y_k(x_k) = G(\mathbf{q})u_k(x_{\overline{k}}). \quad (2.10)$$

As a special case, note that

$$u_{k+r}(x_k) = \mathbf{q}^r u_k(x_{\overline{k}}). \quad (2.11)$$

Example 2. *FIA filtering.* Let $N(\mathbf{q}) = 2\mathbf{q} + 3$ and $D(\mathbf{q}) = \mathbf{q}^2 + 4\mathbf{q} + 5$, and for all $k \geq -n$, define

$$u_k(x) \triangleq z_k x + 1. \quad (2.12)$$

The corresponding FIA filter is thus given by

$$y_k(x_k) = -4y_{k-1}(x_{k-1}) - 5y_{k-2}(x_{k-2}) + 2(z_{k-1}x_k + 1) + 3(z_{k-2}x_k + 1). \tag{2.13}$$

With the data $(z_k)_{k=-2}^{0} = (14, 15, 16)$, $(x_k)_{k=0}^{1} = (19, 20)$, and $(y_k)_{k=-2}^{-1} = (10, 11)$, (2.13) yields

$$y_0(x_0) = -4y_{-1} - 5y_{-2} + 2(z_{-1}x_0 + 1) + 3(z_{-2}x_0 + 1) = 1279, \tag{2.14}$$

$$y_1(x_1) = -4y_0(x_0) - 5y_{-1} + 2(z_0x_1 + 1) + 3(z_{-1}x_1 + 1) = -3626. \tag{2.15}$$

$\diamond$

2.3 Products of MIMO Transfer Functions

Definition 2.3. *Let $P \in \mathbb{R}[\mathbf{z}]^{l_1 \times l_2}$. Then the* normal rank of P *is defined by*

$$\operatorname{rank} P \stackrel{\triangle}{=} \max_{\mathbf{z} \in \mathbb{C}} \operatorname{rank} P(\mathbf{z}). \tag{2.16}$$

Definition 2.4. *Let (A, B, C, D) be a realization of $G \in \mathbb{R}(\mathbf{z})_{\text{prop}}^{l_1 \times l_2}$, where $A \in \mathbb{R}^{n \times n}$. Then the* Rosenbrock system matrix $\mathcal{R}_{(A,B,C,D)} \in \mathbb{R}[\mathbf{z}]^{(n+l_1)\times(n+l_2)}$ *of (A, B, C, D) is the polynomial matrix*

$$\mathcal{R}_{(A,B,C,D)}(\mathbf{z}) \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{z}I - A & B \\ C & -D \end{bmatrix}, \tag{2.17}$$

and $\mathbf{z}_0 \in \mathbb{C}$ is an invariant zero *of (A, B, C, D) if*

$$\operatorname{rank} \mathcal{R}_{(A,B,C,D)}(\mathbf{z}_0) < \operatorname{rank} \mathcal{R}_{(A,B,C,D)}. \tag{2.18}$$

If, in addition, (A, B, C, D) is minimal, then $\mathcal{R}_{(A,B,C,D)}$ is denoted by $\mathcal{R}_G$, and $\mathbf{z}_0 \in \mathbb{C}$ is a

transmission zero *of* G *if*

$$\operatorname{rank} \mathcal{R}_G(\mathbf{z}_0) < \operatorname{rank} \mathcal{R}_G. \tag{2.19}$$

Definition 2.5. *Let* (A, B, C, D) *be a realization of* $G \in \mathbb{R}(\mathbf{z})^{l_1 \times l_2}_{\text{prop}}$. *Then* $\text{IZ}(A, B, C, D)$ *is the multiset of invariant zeros of* (A, B, C, D)*, and* $\text{TZ}(G)$ *is the multiset of transmission zeros of* G.

Definition 2.6. *Let* $G_1 \in \mathbb{R}(\mathbf{z})^{l_1 \times l_2}_{\text{prop}}$ *and* $G_2 \in \mathbb{R}(\mathbf{z})^{l_2 \times l_3}_{\text{prop}}$ *with minimal realizations* (A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2)*, respectively. Define* $G_{12} \triangleq G_1 G_2$*, and consider its realization*

$$A_{12} \triangleq \begin{bmatrix} A_1 & B_1 C_2 \\ 0 & A_2 \end{bmatrix}, \quad B_{12} \triangleq \begin{bmatrix} B_1 D_2 \\ B_2 \end{bmatrix}, \quad C_{12} \triangleq \begin{bmatrix} C_1 & D_1 C_2 \end{bmatrix}, \quad D_{12} \triangleq D_1 D_2. \tag{2.20}$$

Then $\mathbf{z}_0 \in \mathbb{C}$ *is a* cascade zero *of* $G_1 G_2$*, if, counting repetitions, it is an invariant zero of* (2.20) *but not a transmission zero of either* G_1 *or* G_2*. The multiset of cascade zeros of* $G_1 G_2$ *is denoted by*

$$\text{CZ}(G_1, G_2) \triangleq \text{IZ}(A_{12}, B_{12}, C_{12}, D_{12}) \backslash [\text{TZ}(G_1) \cup \text{TZ}(G_2)]. \tag{2.21}$$

Related results are found in [34, 35]. Squaring is discussed in [36–38] and used in [39] to eliminate NMP zeros. It has been shown numerically that $\text{CZ}(G_1, G_2)$ is independent of the basis used by the minimal realizations (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2), and thus, $\text{CZ}(G_1, G_2)$ is well defined. A proof of this is future work.

The following result shows that cascade zeros of square transfer functions $G_1 G_2$ exist only in the case $l_1 \leq l_2$. Part of the credit for the proof of this result goes to Muneeza Azmat.

Proposition 2.1. *Let* $G_1 \in \mathbb{R}(\mathbf{z})^{l_1 \times l_2}_{\text{prop}}$ *and* $G_2 \in \mathbb{R}(\mathbf{z})^{l_2 \times l_1}_{\text{prop}}$ *with minimal realizations*

(A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2)*, respectively, where* $A_1 \in \mathbb{R}^{n_1 \times n_1}$ *and* $A_2 \in \mathbb{R}^{n_2 \times n_2}$*, and assume that* G_1 *and* G_2 *have full normal rank. Define* $G_{12} \stackrel{\triangle}{=} G_1 G_2$ *and consider its realization* (2.20)*. If* $\mathrm{CZ}(G_1, G_2)$ *is not empty, then* $l_1 < l_2$*.*

Proof. Suppose that $l_1 \geq l_2$, and let $\mathbf{z} \in \mathrm{CZ}(G_1, G_2)$. Since $\mathbf{z}$ is not a transmission zero of either G_1 or G_2, G_1 has full column rank, and G_2 has full row rank, it follows from [40, Proposition 16.10.3] that

$$\operatorname{rank} \begin{bmatrix} \mathbf{z} I_{n_1} - A_1 & B_1 \\ C_1 & -D_1 \end{bmatrix} = n_1 + l_2, \tag{2.22}$$

$$\operatorname{rank} \begin{bmatrix} \mathbf{z} I_{n_2} - A_2 & B_2 \\ C_2 & -D_2 \end{bmatrix} = n_2 + l_2. \tag{2.23}$$

Next, note that

$$\mathcal{R}_{(A_{12}, B_{12}, C_{12}, D_{12})}(\mathbf{z}) = \begin{bmatrix} \mathbf{z} I_{n_1} - A_1 & -B_1 C_2 & B_1 D_2 \\ 0 & \mathbf{z} I_{n_2} - A_2 & B_2 \\ C_1 & D_1 C_2 & -D_1 D_2 \end{bmatrix} = N_1(\mathbf{z}) N_2(\mathbf{z}), \tag{2.24}$$

where

$$N_1(\mathbf{z}) \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{z} I_{n_1} - A_1 & 0 & -B_1 \\ 0 & I_{n_2} & 0 \\ C_1 & 0 & D_1 \end{bmatrix} \in \mathbb{R}[\mathbf{z}]^{(n_1+n_2+l_1) \times (n_1+n_2+l_2)}, \tag{2.25}$$

$$N_2(\mathbf{z}) \stackrel{\triangle}{=} \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & \mathbf{z} I_{n_2} - A_2 & B_2 \\ 0 & C_2 & -D_2 \end{bmatrix} \in \mathbb{R}[\mathbf{z}]^{(n_1+n_2+l_2) \times (n_1+n_2+l_1)}. \tag{2.26}$$

It follows from (2.22) and (2.23) that

$$\operatorname{rank} N_1(\mathbf{z}) = \operatorname{rank} N_2(\mathbf{z}) = n_1 + n_2 + l_2. \tag{2.27}$$

Next, Sylvester's inequality [40, p. 292, 294] implies

$$\begin{aligned}\operatorname{rank} N_1(\mathbf{z}) + \operatorname{rank} N_2(\mathbf{z}) - n_1 - n_2 - l_2 &\leq \operatorname{rank} N_1(\mathbf{z})N_2(\mathbf{z}) \\ &\leq \min\{\operatorname{rank} N_1(\mathbf{z}), \operatorname{rank} N_2(\mathbf{z})\}. \end{aligned} \tag{2.28}$$

It follows from (2.24)–(2.28) that

$$\operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}(\mathbf{z}) = n_1 + n_2 + l_2, \tag{2.29}$$

which shows that there are no values of $\mathbf{z}$ such that

$$\operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}(\mathbf{z}) < \operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})},$$

and thus, $\mathbf{z} \notin \mathrm{CZ}(G_1, G_2)$, which is a contradiction. □

Definition 2.7. *Let* $G_1 \in \mathbb{R}(\mathbf{z})^{l_1 \times l_2}_{\mathrm{prop}}$ *and* $G_2 \in \mathbb{R}(\mathbf{z})^{l_2 \times l_3}_{\mathrm{prop}}$. *Then the product* $G_1G_2 \in \mathbb{R}(\mathbf{z})^{l_1 \times l_1}_{\mathrm{prop}}$ *is* down squared *if* $l_1 < l_2$ *and* up squared *if* $l_1 > l_2$.

Definition 2.8. *Let* $G_1 \in \mathbb{R}(\mathbf{z})^{l_1 \times l_2}_{\mathrm{prop}}$ *and* $G_2 \in \mathbb{R}(\mathbf{z})^{l_2 \times l_3}_{\mathrm{prop}}$ *with minimal realizations* (A_1, B_1, C_1, D_1) *and* (A_2, B_2, C_2, D_2), *respectively. Define* $G_{12} \stackrel{\triangle}{=} G_1G_2$, *and consider its realization* (2.20). *Then* $\mathbf{z}_0 \in \mathbb{C}$ *is an* evanescent zero *of* G_1G_2, *if, counting repetitions, it is a cascade zero of* (2.20) *but not a transmission zero of* G_{12}. *The multiset of evanescent zeros of* (2.20) *is denoted by*

$$\mathrm{EZ}(G_1, G_2) \stackrel{\triangle}{=} \mathrm{CZ}(G_1, G_2) \backslash \mathrm{TZ}(G_{12}). \tag{2.30}$$

Example 3. *Cascade and evanescent zeros.* Consider the transfer functions

$$G_1(\mathbf{z}) = \frac{1}{\mathbf{z}(\mathbf{z}-3)} \begin{bmatrix} \mathbf{z} & -1 \end{bmatrix}, \quad G_2(\mathbf{z}) = \frac{1}{\mathbf{z}(\mathbf{z}-4)} \begin{bmatrix} \mathbf{z}-1 \\ 4\mathbf{z}-6 \end{bmatrix}, \tag{2.31}$$

which have minimal realizations (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2), respectively, where

$$A_1 \triangleq \begin{bmatrix} 0 & 0 \\ 1 & 3 \end{bmatrix}, \; B_1 \triangleq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \; C_1 \triangleq \begin{bmatrix} 0 & 1 \end{bmatrix}, \; D_1 \triangleq \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{2.32}$$

$$A_2 \triangleq \begin{bmatrix} 4 & 0 \\ 1 & 0 \end{bmatrix}, \; B_2 \triangleq \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \; C_2 \triangleq \begin{bmatrix} 0.5 & -0.5 \\ 2 & -3 \end{bmatrix}, \; D_2 \triangleq \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.33}$$

The Rosenbrock system matrices for (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) are

$$\mathcal{R}_{G_1}(\mathbf{z}) \triangleq \begin{bmatrix} \mathbf{z} & 0 & 0 & -1 \\ -1 & \mathbf{z}-3 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathcal{R}_{G_1}(\mathbf{z}) \triangleq \begin{bmatrix} \mathbf{z}-4 & 0 & 2 \\ -1 & \mathbf{z} & 0 \\ 0.5 & -0.5 & 0 \\ 2 & -3 & 0 \end{bmatrix}, \tag{2.34}$$

which show that $\operatorname{rank} \mathcal{R}_{G_1}(\mathbf{z}) = \operatorname{rank} \mathcal{R}_{G_1}$ and $\operatorname{rank} \mathcal{R}_{G_2}(\mathbf{z}) = \operatorname{rank} \mathcal{R}_{G_2}$, and thus $\mathrm{TZ}(G_1)$ and $\mathrm{TZ}(G_2)$ are empty. Next, consider the product $G_{12} \triangleq G_1 G_2$ with the realization (2.20), which has the Rosenbrock system matrix

$$\mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}(\mathbf{z}) \triangleq \begin{bmatrix} \mathbf{z} & 0 & 2 & -3 & 0 \\ -1 & \mathbf{z}-3 & -0.5 & 0.5 & 0 \\ 0 & 0 & \mathbf{z}-4 & 0 & 2 \\ 0 & 0 & -1 & \mathbf{z} & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{2.35}$$

It can be shown that $\operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}(2) < \operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}$ and $\operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}(3) < \operatorname{rank} \mathcal{R}_{(A_{12},B_{12},C_{12},D_{12})}$. Since $\mathrm{TZ}(G_1)$ and $\mathrm{TZ}(G_2)$ are empty, it follows that $\mathbf{z} = 2$ and $\mathbf{z} = 3$ are elements of $\mathrm{CZ}(G_1, G_2)$. Next, consider the product of the transfer functions in (2.31)

$$G_{12}(\mathbf{z}) \triangleq G_1(\mathbf{z}) G_2(\mathbf{z}) = \frac{(\mathbf{z}-2)(\mathbf{z}-3)}{\mathbf{z}^2(\mathbf{z}-3)(\mathbf{z}-4)} = \frac{\mathbf{z}-2}{\mathbf{z}^2(\mathbf{z}-4)}, \tag{2.36}$$

where the cascade zero at 3 is cancelled by a pole of G_1, and thus $\mathbf{z} = 3$ is not an element off $\mathrm{TZ}(G_{12})$. Therefore, $\mathbf{z} = 3$ is an element of $\mathrm{EZ}(G_1, G_2)$. $\diamond$

2.4 Recursive Least Squares

Many estimation and control problems involve a process of the form

$$y_k = \phi_k \theta, \tag{2.37}$$

where $k = 0, 1, 2, \ldots$ is the discrete-time step corresponding to the continuous-time step size T_s, the scalar or vector $y_k \in \mathbb{R}^p$ is the measurement at step k, the matrix $\phi_k \in \mathbb{R}^{p \times n}$ is the regressor at step k whose entries consist of current and past data, and $\theta \in \mathbb{R}^n$ is a column vector of n unknown parameters. The objective is to use y_k and ϕ_k to estimate the components of θ. In applications, y_k and ϕ_k are corrupted by noise, and thus (2.37) does not hold exactly. This motivates the need for the least squares estimates of θ given below.

The measurements y_k and the data in ϕ_k are typically obtained from a continuous-time process and as such are available at the sample times kT_s, where T_s is the sample interval. The batch approach to this problem is to collect a large amount of data and then apply least squares optimization to the collected data to compute an estimate of θ. In particular, collecting data over the time window $i = 0, \ldots, k$, it follows from (2.37) that

$$Y = \Phi \theta, \tag{2.38}$$

where

$$Y \triangleq \begin{bmatrix} y_0 \\ \vdots \\ y_k \end{bmatrix}, \quad \Phi \triangleq \begin{bmatrix} \phi_0 \\ \vdots \\ \phi_k \end{bmatrix}. \tag{2.39}$$

Note that (2.38) has the form $Ax = b$, A denotes Φ, x denotes θ, and b denotes Y.

In the presence of noise corrupting the data Y and Φ, (2.38) may not have a solution. In this case, it is useful to replace (2.38) by a least-squares optimization problem of the form

$$\begin{aligned} J_k(\hat{\theta}) &\triangleq \sum_{i=0}^{k} (y_i - \phi_i\hat{\theta})^{\mathrm{T}}(y_i - \phi_i\hat{\theta}) + (\hat{\theta} - \theta_0)^{\mathrm{T}} R(\hat{\theta} - \theta_0) \\ &= (Y - \Phi\hat{\theta})^{\mathrm{T}}(Y - \Phi\hat{\theta}) + (\hat{\theta} - \theta_0)^{\mathrm{T}} R(\hat{\theta} - \theta_0), \end{aligned} \tag{2.40}$$

where R is a positive-semidefinite (and thus, by definition, symmetric) matrix and θ_0 is an initial estimate of θ. Assuming that R is chosen such that the inverse in (2.41) below exists, the regularization term $(\hat{\theta} - \theta_0)^{\mathrm{T}} R(\hat{\theta} - \theta_0)$ weights the initial estimate and ensures that J_k has a unique global minimizer. In particular, the *batch least-squares* (BLS) minimizer of (2.40) is given by

$$\theta_{\mathrm{opt},R} = (\Phi^{\mathrm{T}}\Phi + R)^{-1}(\Phi^{\mathrm{T}}Y + R\theta_0). \tag{2.41}$$

Note that the inverse required to compute (2.41) is of size $n \times n$, and thus the computational requirement of the inverse is of order n^3. In addition to the inverse, three matrix multiplications are needed. Note also that the memory needed to store Φ grows with k. Furthermore, if Φ has full column rank, then R can be set to zero, and thus (2.41) becomes

$$\theta_{\mathrm{opt},0} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}Y. \tag{2.42}$$

In the case where (2.38) has a solution and Φ has full column rank, (2.42) is the unique solution of (2.38).

In many applications, computational speed and memory are limited. One way to alleviate these requirements is to recursively update an estimate of $\theta_{\mathrm{opt},R}$ using each additional measurement y_k. A recursive algorithm of this type is especially convenient for real-time applications. *Recursive least squares* (RLS) is an iterative implementation of BLS that

significantly reduces the computational and storage requirements of BLS.

Variations of RLS have been studied for more than half a century. An early exposition is given in [41], which emphasizes the real-time utility of RLS relative to BLS. Applications of RLS to adaptive control are discussed in [42, pp. 41, 103]. Numerous extensions of RLS have been developed to address initialization, forgetting, and numerical stability, for example, [43–46]. The development of RLS that is closest to this dissertation is given in [47, pp. 26–28].

This subsection provides a statement and proof of the RLS algorithm. This result involves a recursive algorithm for optimizing J_k at each step k. The optimization of J_k updates the estimate θ_k of θ as measurements and data become available. As an extension of (2.40), the cost function (2.47) below includes a forgetting factor λ, which provides higher weighting to more recent measurements and data.

The following result is the matrix inversion lemma [40, p. 304].

Lemma 2.1. *Let $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times p}$, and $V \in \mathbb{R}^{p\times n}$, and assume that A, C, and $A + UCV$ are nonsingular. Then*

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}. \tag{2.43}$$

The following result is the quadratic minimization lemma.

Lemma 2.2. *Let $A \in \mathbb{R}^{n\times n}$, assume that A is positive definite, let $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$, and define $f : \mathbb{R}^n \to \mathbb{R}$ by*

$$f(x) \triangleq x^{\mathrm{T}}Ax + 2b^{\mathrm{T}}x + c. \tag{2.44}$$

Then the unique minimizer of f is

$$x_{\mathrm{opt}} = -A^{-1}b, \tag{2.45}$$

and the minimum value of f is

$$f(x_{\text{opt}}) = c - b^{\mathrm{T}} A^{-1} b. \tag{2.46}$$

Theorem 2.1. *For all $k \geq 0$, let $\phi_k \in \mathbb{R}^{p \times n}$ and $y_k \in \mathbb{R}^p$. Furthermore, let $\theta_0 \in \mathbb{R}^n$, let $P_0 \in \mathbb{R}^{n \times n}$ be positive definite, and let $\lambda \in (0, 1]$. Furthermore, for all $k \geq 0$, denote the minimizer of the function*

$$J_k(\hat{\theta}) \triangleq \sum_{i=0}^{k} \lambda^{k-i} (y_i - \phi_i \hat{\theta})^{\mathrm{T}} (y_i - \phi_i \hat{\theta}) + \lambda^{k+1} (\hat{\theta} - \theta_0)^{\mathrm{T}} P_0^{-1} (\hat{\theta} - \theta_0) \tag{2.47}$$

by

$$\theta_{k+1} \triangleq \underset{\hat{\theta} \in \mathbb{R}^n}{\operatorname{argmin}} \; J_k(\hat{\theta}). \tag{2.48}$$

Then, for all $k \geq 0$, θ_{k+1} is given by

$$P_{k+1} = \frac{1}{\lambda} P_k - \frac{1}{\lambda} P_k \phi_k^{\mathrm{T}} (\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1} \phi_k P_k, \tag{2.49}$$

$$\theta_{k+1} = \theta_k + P_{k+1} \phi_k^{\mathrm{T}} (y_k - \phi_k \theta_k). \tag{2.50}$$

Proof. Note that $J_0(\hat{\theta})$ can be written as

$$J_0(\hat{\theta}) = \hat{\theta}^{\mathrm{T}} A_0 \hat{\theta} + 2 b_0^{\mathrm{T}} \hat{\theta} + c_0,$$

where

$$\begin{aligned} A_0 &\triangleq \phi_0^{\mathrm{T}} \phi_0 + \lambda P_0^{-1}, \\ b_0 &\triangleq -\phi_0^{\mathrm{T}} y_0 - \lambda P_0^{-1} \theta_0, \\ c_0 &\triangleq y_0^{\mathrm{T}} y_0 + \lambda \theta_0 P_0^{-1} \theta_0. \end{aligned}$$

Defining

$$P_1 \stackrel{\triangle}{=} A_0^{-1},$$

it follows from Lemma 2.1 with $A = P_0^{-1}$, $C = \frac{1}{\lambda}I$, $U = \phi_0^{\mathrm{T}}$, and $V = \phi_0$ that

$$\begin{aligned} P_1 &= \frac{1}{\lambda}(P_0^{-1} + \frac{1}{\lambda}\phi_0^{\mathrm{T}}\phi_0)^{-1} \\ &= \frac{1}{\lambda}P_0 - \frac{1}{\lambda}P_0\phi_0^{\mathrm{T}}(\lambda I + \phi_0 P_0 \phi_0^{\mathrm{T}})^{-1}\phi_0 P_0. \end{aligned}$$

Hence, (2.49) is satisfied for $k = 0$. In addition, since A_0 is positive definite, it follows from Lemma 2.2 that the unique minimizer θ_1 of J_0 is given by

$$\begin{aligned} \theta_1 &= -A_0^{-1}b_0 \\ &= P_1\phi_0^{\mathrm{T}}y_0 + \lambda P_1 P_0^{-1}\theta_0 \\ &= P_1\phi_0^{\mathrm{T}}y_0 + P_1(P_1^{-1} - \phi_0^{\mathrm{T}}\phi_0)\theta_0 \\ &= P_1\phi_0^{\mathrm{T}}y_0 + (I - P_1\phi_0^{\mathrm{T}}\phi_0)\theta_0 \\ &= \theta_0 + P_1\phi_0^{\mathrm{T}}(y_0 - \phi_0\theta_0). \end{aligned}$$

Hence, (2.50) is satisfied for $k = 0$.

Now, let $k \geq 1$. Then $J_k(\hat{\theta})$ can be written as

$$J_k(\hat{\theta}) = \hat{\theta}^{\mathrm{T}}A_k\hat{\theta} + 2b_k^{\mathrm{T}}\hat{\theta} + c_k,$$

where

$$A_k \triangleq \sum_{i=0}^{k} \lambda^{k-i} \phi_i^{\mathrm{T}} \phi_i + \lambda^{k+1} P_0^{-1},$$
$$b_k \triangleq -\sum_{i=0}^{k} \lambda^{k-i} \phi_i^{\mathrm{T}} y_i - \lambda^{k+1} P_0^{-1} \theta_0,$$
$$c_k \triangleq \sum_{i=0}^{k} \lambda^{k-i} y_i^{\mathrm{T}} y_i + \lambda^{k+1} \theta_0 P_0^{-1} \theta_0.$$

Furthermore, A_k and b_k can be written recursively as

$$A_k = \lambda A_{k-1} + \phi_k^{\mathrm{T}} \phi_k,$$
$$b_k = \lambda b_{k-1} - \phi_k^{\mathrm{T}} y_k.$$

Defining

$$P_{k+1} \triangleq A_k^{-1},$$

it follows from Lemma 2.1 with $A = P_k^{-1}$, $C = \frac{1}{\lambda} I$, $U = \phi_k^{\mathrm{T}}$, and $V = \phi_k$ that

$$\begin{aligned} P_{k+1} &= [\lambda(A_{k-1} + \frac{1}{\lambda} \phi_k^{\mathrm{T}} \phi_k)]^{-1} \\ &= \frac{1}{\lambda}(P_k^{-1} + \frac{1}{\lambda} \phi_k^{\mathrm{T}} \phi_k)^{-1} \\ &= \frac{1}{\lambda} P_k - \frac{1}{\lambda} P_k \phi_k^{\mathrm{T}} (\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1} \phi_k P_k. \end{aligned}$$

Hence, (2.49) is satisfied. Furthermore, the minimizer θ_{k+1} of J_k is given by

$$\theta_{k+1} = -A_k^{-1} b_k.$$

Since A_k is positive definite, it follows from Lemma 2.2 that

$$\begin{aligned}
\theta_{k+1} &= -A_k^{-1} b_k \\
&= A_k^{-1}(\phi_k^{\mathrm{T}} y_k - \lambda b_{k-1}) \\
&= A_k^{-1}(\phi_k^{\mathrm{T}} y_k + \lambda A_{k-1}\theta_k) \\
&= A_k^{-1}[\phi_k^{\mathrm{T}} y_k + (A_k - \phi_k^{\mathrm{T}}\phi_k)\theta_k] \\
&= A_k^{-1}\phi_k^{\mathrm{T}} y_k + (I - A_k^{-1}\phi_k^{\mathrm{T}}\phi_k)\theta_k \\
&= \theta_k + P_{k+1}\phi_k^{\mathrm{T}}(y_k - \phi_k\theta_k).
\end{aligned}$$

Hence, (2.50) is satisfied. □

Note that, if $\lambda = 1$, R is positive definite, and $P_0 \triangleq R^{-1}$, then the RLS minimizer θ_{k+1} given by (2.48) is equal to the BLS minimizer $\theta_{\mathrm{opt},R}$ given by (2.41).

The notation θ_{k+1} for the minimizer of J_k emphasizes the fact that θ_{k+1}, which is based on data available up to step k, is not available until the update given by (2.49), (2.50) is completed, which occurs at step $k+1$. Real-time implementation of the RLS update is discussed below.

The derivation of RLS given by the proof of Theorem 2.1 is an extension of the RLS derivation given in [47]. In particular, the derivation given in [47, pp. 26–28] is based on the cost function (2.47) but without the regularization term involving P_0. As can be seen in the proof of Theorem 2.1, this term guarantees that A_0 is nonsingular in the first step and that A_k is nonsingular in the inductive step. In the case of BLS, the role of P_0 is played by the matrix R.

The computational requirements of RLS are primarily determined by n. In particular, P_k is $n \times n$, and thus the computational requirement for updating P_k given by (2.49) is of order n^2. Furthermore, the inverse in (2.49) is of size $p \times p$, which, since typically $p << n$, is much less demanding than the inverse required by BLS. In addition, the storage requirements of RLS are of order n^2, which does not grow with k. Consequently, the

computational and memory requirements of RLS are signficantly less than those of BLS.

2.4.1 Alternative θ_k Update with Inverse

Considering the following Lemma that is required for reformulating the updates of P_k and θ_k.

Lemma 2.3. *Let $A \in \mathbb{R}^{n\times n}$ be positive semidefinite, let $B \in \mathbb{R}^{n\times m}$, and let $C \in \mathbb{R}^{n\times n}$ be positive definite. Then,*

$$[I - AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}B]AB^{\mathrm{T}} = AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}C. \tag{2.51}$$

Proof. Note that

$$\begin{aligned} AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}C &= AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}(C + BAB^{\mathrm{T}} - BAB^{\mathrm{T}}) \\ &= AB^{\mathrm{T}}[I - (C + BAB^{\mathrm{T}})^{-1}BAB^{\mathrm{T}}] \\ &= AB^{\mathrm{T}} - AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}BAB^{\mathrm{T}} \\ &= [I - AB^{\mathrm{T}}(C + BAB^{\mathrm{T}})^{-1}B]AB^{\mathrm{T}}. \end{aligned}$$

□

The following result is a variation of Theorem 2.1. In this formulation, the updates of P_k and θ_k are reversed.

Theorem 2.2. *For all $k \geq 0$, let $\phi_k \in \mathbb{R}^{p\times n}$ and $y_k \in \mathbb{R}^p$. Furthermore, let $\theta_0 \in \mathbb{R}^n$, let $P_0 \in \mathbb{R}^{n\times n}$ be positive definite, and let $\lambda \in (0, 1]$. Furthermore, for all $k \geq 0$, denote the minimizer of the function* (2.47) *by* (2.48)*. Then, for all $k \geq 0$, θ_{k+1} is given by*

$$\theta_{k+1} = \theta_k + P_k\phi_k^{\mathrm{T}}(\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1}(y_k - \phi_k\theta_k), \tag{2.52}$$

$$P_{k+1} = \frac{1}{\lambda}P_k - \frac{1}{\lambda}P_k\phi_k^{\mathrm{T}}(\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1}\phi_k P_k. \tag{2.53}$$

Proof. Using (2.49) to substitute P_{k+1} into (2.50) yields

$$\begin{aligned}\theta_{k+1} &= \theta_k + [\frac{1}{\lambda}P_k - \frac{1}{\lambda}P_k\phi_k^{\mathrm{T}}(\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1}\phi_k P_k]\phi_k^{\mathrm{T}}(y_k - \phi_k\theta_k) \\ &= \theta_k + \frac{1}{\lambda}[I - P_k\phi_k^{\mathrm{T}}(\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1}\phi_k]P_k\phi_k^{\mathrm{T}}(y_k - \phi_k\theta_k) \\ &= \theta_k + P_k\phi_k^{\mathrm{T}}(\lambda I + \phi_k P_k \phi_k^{\mathrm{T}})^{-1}(y_k - \phi_k\theta_k),\end{aligned}$$

where the last equality follows from Lemma 2.3. Hence, (2.52) holds. Finally, (2.53) is identical to (2.49). □

2.4.2 Real-Time Implementation of RLS

In many applications, it is desirable to implement RLS so that the estimate θ_k is available in real time without latency. Note that the estimate θ_{k+1} given by (2.50) depends on measurements available up to and including step k, namely, y_k and ϕ_k. Since time is needed to compute θ_{k+1}, the updated estimate θ_{k+1} is not available at time k; rather, it is available at the next step, namely, $k+1$. Consequently, the minimizer of J_k is denoted by θ_{k+1}, where the subscript $k+1$ conveys the fact that the minimizer of J_k is not available until step $k+1$. In contrast, the notation used in [47, p. 27] is θ_k. Figure 2.1 shows how the measurements and data that are available at step k are used during the time interval $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}}]$ to compute the next estimate θ_{k+1}.

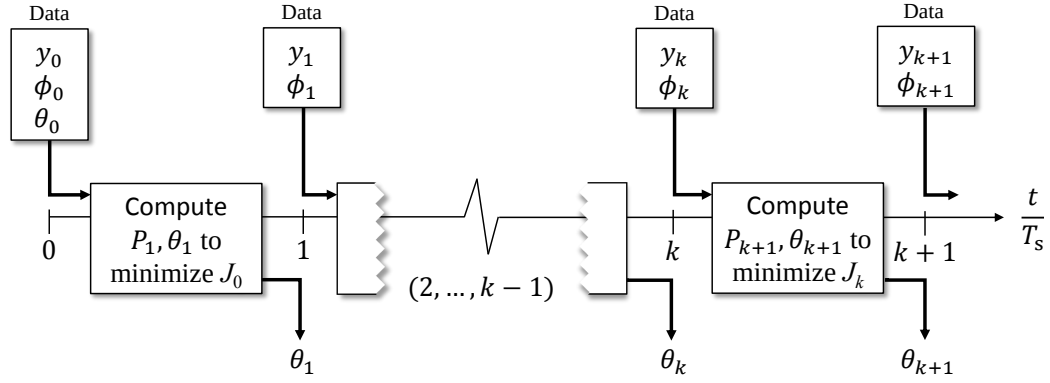

Figure 2.1: Real-Time Implementation of RLS. Data at step k, which corresponds to time $t = kT_s$, are used to compute the minimizer θ_{k+1} of the cost J_k. Due to the time needed for the computation, the estimate θ_{k+1} of the unknown parameter θ is not available until step $k + 1$.

CHAPTER 3

Sampled-Data Adaptive-Control Architecture

This chapter describes the adaptive control architecture considered in this dissertation. All of the examples in this dissertation consider continuous-time systems under sampled-data control using discrete-time adaptive controllers. In particular, consider the adaptive control architecture shown in Figure 3.1, where a realization of $G(s) \triangleq [G_u(s) \;\; G_w(s)]$ is given by

$$\dot{x}(t) = Ax(t) + Bu(t) + B_w w(t), \tag{3.1}$$

$$y(t) = Cx(t) + D_u u(t) + v(t), \tag{3.2}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control, $w(t) \in \mathbb{R}^l$ is the disturbance, $y(t) \in \mathbb{R}^p$ is the noisy measurement of the system output, $v(t) \in \mathbb{R}^p$ is the sensor noise, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $B_w \in \mathbb{R}^{n\times l}$, $C \in \mathbb{R}^{p\times n}$, and $D_u \in \mathbb{R}^{p\times m}$.

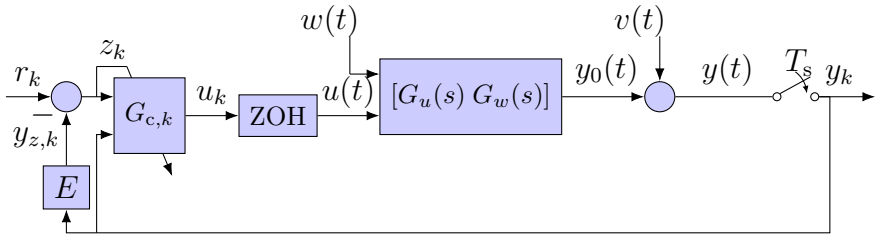

Figure 3.1: Command following and disturbance rejection under sampled-data adaptive control. The objective is to follow commands r_k to the performance variable $y_{z,k} = Ey_k$. All sample-and-hold operations are synchronous.

Define

$$G_u(s) \triangleq C(sI_n - A)^{-1}B + D_u, \tag{3.3}$$

$$G_w(s) \triangleq C(sI_n - A)^{-1}B_w + D_u, \tag{3.4}$$

where $G_u \in \mathbb{R}(s)^{p\times m}_{\text{prop}}$ and $G_w \in \mathbb{R}(s)^{p\times l}_{\text{prop}}$ are proper $p \times m$ and $p \times l$ transfer functions, respectively. The disturbance $w(t)$ is matched if there exists $\overline{U} \in \mathbb{R}^{m\times m}$ such that $B_w = B\overline{U}$; otherwise, the disturbance is unmatched. The system output $y_0(t) \in \mathbb{R}^p$ is corrupted by sensor noise $v(t)$ and sampled to produce $y_k \in \mathbb{R}^p$. The sampling operation can be realized as $y_k \triangleq y_0(kT_\text{s}) + v_k$, where $v_k \triangleq v(kT_\text{s}) \in \mathbb{R}^p$ is the sampled sensor noise and $T_\text{s} \in \mathbb{R}$ is the sample time. In this dissertation the statistics of the sampled sensor noise v_k are specified. The performance variable is $y_{z,k} \triangleq Ey_k \in \mathbb{R}^q$, where the matrix $E \in \mathbb{R}^{q\times p}$ selects components of y_k or a linear combination of the components of y_k that are required to follow the command $r_k \in \mathbb{R}^q$. The command-following error is thus $z_k \triangleq r_k - y_{z,k} \in \mathbb{R}^q$. The inputs to the adaptive feedback controller $G_{\text{c},k}(\mathbf{q})$ are the measurement y_k and the command-following error z_k. The adaptive feedback controller produces the discrete-time control $u_k \in \mathbb{R}^m$ at each step k. The continuous-time control $u(t)$ is produced by applying a zero-order-hold operator to u_k. Note that z_k serves as the adaptation variable, as denoted by the diagonal line in Figure 3.1 passing through $G_{\text{c},k}(\mathbf{q})$. The objective is to minimize the magnitude of the command-following error z_k in the presence of the disturbance $w(t)$ and sensor noise $v(t)$.

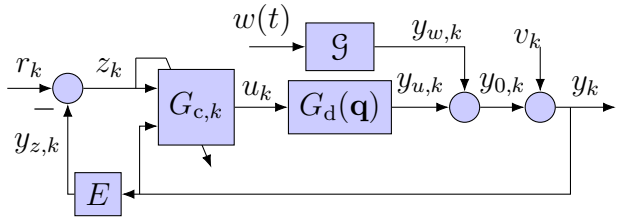

Figure 3.2: Equivalent representation of Figure 3.1. The exact discretization $G_\text{d}(\mathbf{q})$ of $G_u(s)$ operates on u_k to generate $y_{u,k}$.

Figure 3.2 shows an equivalent representation of Figure 3.1, where $w(t)$ and $y_{w,k}$ are related by the operator

$$y_{w,k} \triangleq \mathcal{G}[w(t)] = C \int_{(k-1)T_\mathrm{s}}^{kT_\mathrm{s}} e^{A(kT_\mathrm{s}-\tau)} B_w w(\tau) \mathrm{d}\tau. \tag{3.5}$$

Note that Figure 3.2 shows two transfer functions in feedback, namely, $G_\mathrm{d}(\mathbf{q})$ and $EG_\mathrm{d}(\mathbf{q})$, which are, respectively, the transfer functions from u_k to y_k and u_k to $y_{z,k}$. Furthermore, $G_\mathrm{d} \in \mathbb{R}(\mathbf{q})^{p\times m}_{\mathrm{prop}}$, where $\mathbf{q}$ is the forward-shift operator, is the exact discretization of $G_u(s)$ using zero-order-hold and sampling operations. For details, see [48, pp. 11]. Consequently,

$$y_k = \mathcal{G}[w(t)] + G_\mathrm{d}(\mathbf{q})u_k + v_k, \tag{3.6}$$

$$z_k = r_k - Ey_k. \tag{3.7}$$

Note that the argument $\mathbf{q}$ of G_d in (3.6) reflects the fact that (3.6) is a time-domain equation whose solution depends on the initial conditions of the input-output system. Using the Z-transform variable $\mathbf{z}$ in place of the forward-shift operator $\mathbf{q}$ would account for the forced response of (3.6) but would implicitly assume zero initial conditions and thus would omit the free response. The distinction between $\mathbf{z}$ and $\mathbf{q}$ in accounting for initial conditions and the resulting free response is discussed in [49, 50]. Since $G_\mathrm{d}(\mathbf{z})$ and $G_\mathrm{d}(\mathbf{q})$ have the same form, the argument has no effect on the algebraic properties of G_d such as poles and zeros.

In order to compute the intersample response of (3.5), the disturbance $w(t)$ is assumed to be piecewise constant within each subinterval of the interval kT_s to $(k+1)T_\mathrm{s}$, where each subinterval has length $T_\mathrm{s}/10$. In particular, letting $\overline{w}_{k,i}$ denote the approximate value

of $w(t)$ for $t \in [(k+\frac{i}{10})T_\mathrm{s}, (k+\frac{i+1}{10})T_\mathrm{s}]$, for $i = 0, \ldots, 9$, it follows that

$$y_{w,k+1} = C \int_{kT_\mathrm{s}}^{(k+1)T_\mathrm{s}} e^{A[(k+1)T_\mathrm{s}-\tau]} B_w w(\tau) d\tau \tag{3.8}$$

$$\approx C \left[\int_{kT_\mathrm{s}}^{kT_\mathrm{s}+\frac{1}{10}T_\mathrm{s}} e^{A[(k+1)T_\mathrm{s}-\tau]} d\tau B_w \overline{w}_{k,0} + \ldots + \int_{kT_\mathrm{s}+\frac{9}{10}T_\mathrm{s}}^{(k+1)T_\mathrm{s}} e^{A[(k+1)T_\mathrm{s}-\tau]} d\tau B_w \overline{w}_{k,9} \right] \tag{3.9}$$

$$= C \left[\int_{\frac{9}{10}T_\mathrm{s}}^{T_\mathrm{s}} e^{A\tau} d\tau B_w \overline{w}_{k,0} + \ldots + \int_{0}^{\frac{1}{10}T_\mathrm{s}} e^{A\tau} d\tau B_w \overline{w}_{k,9} \right]. \tag{3.10}$$

Within each subinterval, the MATLAB function ODE45 is used to integrate the dynamics of $G(s)$. For all examples in this dissertation, the ODE45 relative and absolute tolerances are set to 2.22045×10^{-14} and 10^{-14}, respectively, which determine the variable step lengths during each subinterval. In the case where $w(t)$ is stochastic, the standard deviation of $\overline{w}_{k,i}$ is specified.

Figure 3.3 shows the intersample response of $G_w(s) = \frac{s-1}{s^2-3s+2}$, where $\overline{w}_{k,i}$ is zero-mean, Gaussian white noise with standard deviation 1 simulated with $T_\mathrm{s} = 0.01$ s/step. In all subsequent numerical examples, the intersample response is computed but not shown.

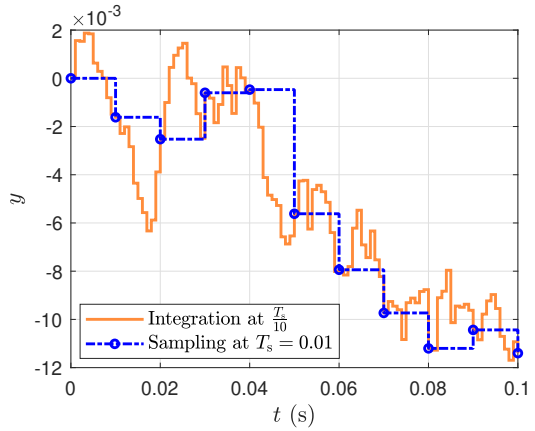

Figure 3.3: Numerical integration of $G_w(s)$ using ODE45 within each subinterval of size $T_\mathrm{s}/10$, where $T_\mathrm{s} = 0.01$ s/step. The intersample response is plotted in orange, and the blue dash-dots show the sampled response.

Chapters 4, 8, and 9 consider SISO continuous-time transfer functions with $G_u(s) =$

$G_w(s)$ of the form

$$G_u(s) = 10e^{-n_{\rm d}T_{\rm s}s}\frac{(s-a)(s-b)(s-c)\prod_{i=1}^{3}(s^2+2\bar{\zeta}_i\bar{\omega}_i s+\bar{\omega}_i^2)}{\prod_{i=1}^{5}(s^2+2\zeta_i\omega_i s+\omega_i^2)}, \tag{3.11}$$

where $n_{\rm d}$ is a nonnegative integer, the parameters $a, b, c, n_{\rm d}$ are given in Table 3.1, and $\bar{\zeta}_1 = 0.96$, $\bar{\zeta}_2 = 0.22$, $\bar{\zeta}_3 = 0.8$, $\bar{\omega}_1 = 54$, $\bar{\omega}_2 = 38$, $\bar{\omega}_3 = 8$, $\zeta_1 = 0.4$, $\zeta_2 = 0.15$, $\zeta_3 = 0.05$, $\zeta_4 = 0.06$, $\zeta_5 = 0.05$, $\omega_1 = 4$, $\omega_2 = 25$, $\omega_3 = 35$, $\omega_4 = 65$, and $\omega_5 = 96$. The transfer function (3.11) with the parameters in Table 3.1 are used to investigate the performance of RCAC, RLSID, and DDRCAC in later sections.

Table 3.1: Special cases of $G_u(s)$ given by (3.11). For each case, the values of $a, b, c, n_{\rm d}$ and the type of zeros are shown.

Case	a	b	c	$n_{\rm d}$	Zeros
1	10	-30	-20	2	1 real NMP
2	10	-30	-20	0	1 real NMP
3	$10+10\jmath$	$10-10\jmath$	-20	2	2 complex NMP

The time delay of $n_{\rm d}T_{\rm s}$, where $n_{\rm d}$ is a nonnegative integer, is included in $G_u(s)$ as $e^{-n_{\rm d}T_{\rm s}}$. Choosing the time delay to be a multiple of $T_{\rm s}$ facilitates investigation of the effect of uncertain discrete-time relative degree on the performance of the closed-loop discrete-time system. Note that (3.11) can be exactly discretized by separately considering the rational and exponential factors. In particular, the rational part of (3.11) is exactly discretized with a zero-order-hold (ZOH) discretization computed using MATLAB command `c2d`, whereas the exponential part of (3.11) is exactly discretized by the factor $\mathbf{q}^{-n_{\rm d}}$ in $G_{\rm d}(\mathbf{q})$. Note that the exact discretization of (3.11) has relative degree $n_{\rm d}+1$.

For all examples in this dissertation, (3.11) is simulated by using a minimal realization whose initial state is zero. Hence, $E = 1$, $p = q = m = l = 1$, and $B = B_w$ in (3.1), (3.2).

CHAPTER 4

Retrospective Cost Adaptive Control

In this chapter, we present the RCAC algorithm, which forms the basis of DDRCAC [33].

4.1 Controller Structure

Consider the strictly proper, discrete-time dynamic compensator

$$u_k = \sum_{i=1}^{n_c} P_{i,k} u_{k-i} + \sum_{i=1}^{n_c} Q_{i,k} \tilde{y}_{k-i}, \tag{4.1}$$

where $k \geq 0$, $u_k \in \mathbb{R}^m$ is the requested control, n_c is the controller window length, $\tilde{y}_k \in \mathbb{R}^{l_y}$, and $Q_{1,k}, \ldots, Q_{n_c,k} \in \mathbb{R}^{m \times l_y}$ and $P_{1,k}, \ldots, P_{n_c,k} \in \mathbb{R}^{m \times m}$ are the numerator and denominator controller coefficient matrices, respectively. For convenience, a "cold" startup is assumed, where $Q_{1,0}, \ldots, Q_{n_c,0}$, $P_{1,0}, \ldots, P_{n_c,0}$, $u_{-n_c}, \ldots, u_{-1}$, and $\tilde{y}_{-n_c}, \ldots, \tilde{y}_{-1}$ are defined to be zero, and thus $u_0 = 0$. The controller (4.1) can be written as

$$u_k = \phi_{c,k} \theta_{c,k}, \tag{4.2}$$

where

$$\phi_{\mathrm{c},k} \triangleq \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-n_\mathrm{c}} \\ \tilde{y}_{k-1} \\ \vdots \\ \tilde{y}_{k-n_\mathrm{c}} \end{bmatrix}^{\mathrm{T}} \otimes I_m \in \mathbb{R}^{m \times l_{\theta_\mathrm{c}}}, \tag{4.3}$$

is the *controller regressor*, $l_{\theta_\mathrm{c}} \triangleq n_\mathrm{c} m(m + l_y)$, and the *controller coefficient vector* is defined by

$$\theta_{\mathrm{c},k} \triangleq \mathrm{vec}\left[P_{1,k} \cdots P_{n_\mathrm{c},k}\, Q_{1,k} \cdots Q_{n_\mathrm{c},k} \right] \in \mathbb{R}^{l_{\theta_\mathrm{c}}}. \tag{4.4}$$

In terms of $\mathbf{q}$, the controller (4.1) can be expressed as

$$u_k = G_{\mathrm{c},k}(\mathbf{q})\tilde{y}_k, \tag{4.5}$$

where

$$N_{\mathrm{c},k}(\mathbf{q}) \triangleq Q_{1,k}\mathbf{q}^{n_\mathrm{c}-1} + \cdots + Q_{n_\mathrm{c},k}, \tag{4.6}$$

$$D_{\mathrm{c},k}(\mathbf{q}) \triangleq I_m\mathbf{q}^{n_\mathrm{c}} - P_{1,k}\mathbf{q}^{n_\mathrm{c}-1} - \cdots - P_{n_\mathrm{c},k}, \tag{4.7}$$

$$G_{\mathrm{c},k}(\mathbf{q}) \triangleq D_{\mathrm{c},k}^{-1}(\mathbf{q})N_{\mathrm{c},k}(\mathbf{q}). \tag{4.8}$$

The signal $\tilde{y}_k$ is constructed from z_k, y_k, and r_k. In the simplest case, $\tilde{y}_k = z_k$, whereas, when additional measurements are available, $\tilde{y}_k = [\ z_k^{\mathrm{T}}\ y_k^{\mathrm{T}}\ ]^{\mathrm{T}}$. Alternatively, feedforward action can be included by setting $\tilde{y}_k = [\ z_k^{\mathrm{T}}\ r_k^{\mathrm{T}}\ ]^{\mathrm{T}}$. More generally, the components of $\tilde{y}_k$ can be arbitrary, fixed linear combinations of the components of z_k, y_k, and r_k. Fixed, nonlinear functions of z_k, y_k, and r_k can also be included in $\tilde{y}_k$; however, this is outside the scope of this dissertation.

4.2 Retrospective Performance Variable

Next, define the filtered signals

$$u_{\mathrm{f},k} \triangleq G_{\mathrm{f}}(\mathbf{q})u_k, \tag{4.9}$$

$$\phi_{\mathrm{f},k} \triangleq G_{\mathrm{f}}(\mathbf{q})\phi_{\mathrm{c},k}, \tag{4.10}$$

where, for startup, $u_{\mathrm{f},k}$ and $\phi_{\mathrm{f},k}$ are initialized at zero and thus are computed as the forced responses of (9.9) and (9.10), respectively. Unless specified otherwise, the same filter initialization is for all filters in the subsequent development. The $q \times m$ filter $G_{\mathrm{f}}(\mathbf{q})$ has the form

$$G_{\mathrm{f}}(\mathbf{q}) \triangleq D_{\mathrm{f}}(\mathbf{q})^{-1}N_{\mathrm{f}}(\mathbf{q}), \tag{4.11}$$

where

$$N_{\mathrm{f}}(\mathbf{q}) \triangleq N_{\mathrm{f},0}\mathbf{q}^{n_{\mathrm{f}}} + N_{\mathrm{f},1}\mathbf{q}^{n_{\mathrm{f}}-1} + \cdots + N_{\mathrm{f},n_{\mathrm{f}}}, \tag{4.12}$$

$$D_{\mathrm{f}}(\mathbf{q}) \triangleq I_q\mathbf{q}^{n_{\mathrm{f}}} + D_{\mathrm{f},1}\mathbf{q}^{n_{\mathrm{f}}-1} + \cdots + D_{\mathrm{f},n_{\mathrm{f}}}, \tag{4.13}$$

n_{f} is the filter window length, and $N_{\mathrm{f},0}, \ldots, N_{\mathrm{f},n_{\mathrm{f}}} \in \mathbb{R}^{q\times m}$ and $D_{\mathrm{f},1}, \ldots, D_{\mathrm{f},n_{\mathrm{f}}} \in \mathbb{R}^{q\times q}$ are the numerator and denominator coefficients of $G_{\mathrm{f}}(\mathbf{q})$, respectively.

Equivalently, (4.9) and (4.10) can be written as

$$u_{\mathrm{f},k} = -DU_{\mathrm{f},k} + NU_k, \tag{4.14}$$

$$\phi_{\mathrm{f},k} = -D\Phi_{\mathrm{f},k} + N\Phi_{\mathrm{c},k}, \tag{4.15}$$

where

$$U_{\mathrm{f},k} \triangleq \begin{bmatrix} u_{\mathrm{f},k-1} \\ \vdots \\ u_{\mathrm{f},k-n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{f}}q}, \quad U_k \triangleq \begin{bmatrix} u_k \\ \vdots \\ u_{k-n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{(n_{\mathrm{f}}+1)m}, \tag{4.16}$$

$$\Phi_{\mathrm{f},k} \triangleq \begin{bmatrix} \phi_{\mathrm{f},k-1} \\ \vdots \\ \phi_{\mathrm{f},k-n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{f}}q \times l_{\theta_{\mathrm{c}}}}, \quad \Phi_{\mathrm{c},k} \triangleq \begin{bmatrix} \phi_{\mathrm{c},k} \\ \vdots \\ \phi_{\mathrm{c},k-n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{(n_{\mathrm{f}}+1)m \times l_{\theta_{\mathrm{c}}}}, \tag{4.17}$$

$$N \triangleq \begin{bmatrix} N_{\mathrm{f},0} \cdots N_{\mathrm{f},n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{q \times m(n_{\mathrm{f}}+1)}, \quad D \triangleq \begin{bmatrix} D_{\mathrm{f},1} \cdots D_{\mathrm{f},n_{\mathrm{f}}} \end{bmatrix} \in \mathbb{R}^{q \times qn_{\mathrm{f}}}. \tag{4.18}$$

Next, in order to update the controller coefficient vector (4.4), define the retrospective performance variable

$$\hat{z}_k(\theta_{\mathrm{c}}) \triangleq z_k - (u_{\mathrm{f},k} - \phi_{\mathrm{f},k}\theta_{\mathrm{c}}), \tag{4.19}$$

where z_k is given by (3.7) and θ_{c} is a generic variable for optimization. Note that $u_{\mathrm{f},k}$ depends on u_k and thus on the current controller coefficient vector $\theta_{\mathrm{c},k}$. The retrospective performance variable $\hat{z}_k(\theta_{\mathrm{c}})$ is used to determine the updated controller coefficient vector $\theta_{\mathrm{c},k+1}$ by minimizing a function of $\hat{z}_k(\theta_{\mathrm{c}})$. The optimized value of $\hat{z}_k$ is thus given by

$$\hat{z}_k(\theta_{\mathrm{c},k+1}) = z_k - (u_{\mathrm{f},k} - \phi_{\mathrm{f},k}\theta_{\mathrm{c},k+1}), \tag{4.20}$$

which shows that the updated controller coefficient vector $\theta_{\mathrm{c},k+1}$ is "applied" retrospectively with the filtered controller regressor $\phi_{\mathrm{f},k}$. Furthermore, note that the filter $G_{\mathrm{f}}(\mathbf{q})$ is used to obtain $\phi_{\mathrm{f},k}$ from ϕ_k by means of (4.10) but ignores past changes in the controller coefficient vector, as can be seen by the product $\phi_{\mathrm{f},k}\theta_{\mathrm{c},k+1}$ in (4.20). Consequently, the filtering used to construct (4.20) ignores changes in the controller coefficient vector over the window $[k - n_{\mathrm{f}}, k]$. The effect of the actual time-dependence of $\theta_{\mathrm{c},k}$ is analyzed in later sections.

Using (4.14) and (4.15), (4.19) can be expressed as

$$\hat{z}_k(\theta_{\mathrm{c}}) = z_k + D(U_{\mathrm{f},k} - \Phi_{\mathrm{f},k}\theta_{\mathrm{c}}) - N(U_k - \Phi_{\mathrm{c},k}\theta_{\mathrm{c}}). \tag{4.21}$$

In the case where $G_{\mathrm{f}}(\mathbf{q})$ is a finite-impulse-response (FIR) transfer function, and thus $D = 0$, it follows from (4.21) that

$$\hat{z}_k(\theta_{\mathrm{c}}) = z_k - NU_k + N\Phi_{\mathrm{c},k}\theta_{\mathrm{c}}. \tag{4.22}$$

4.3 Online Optimization Using RLS

In order to account for the control effort, define

$$z_{\mathrm{c},k}(\theta_{\mathrm{c}}) \triangleq \begin{bmatrix} E_z\hat{z}_k(\theta_{\mathrm{c}}) \\ E_u\phi_{\mathrm{c},k}\theta_{\mathrm{c}} \end{bmatrix} \in \mathbb{R}^{q+r_1}, \tag{4.23}$$

where the performance weighting $E_z \in \mathbb{R}^{q\times q}$ is nonsingular, and $E_u \in \mathbb{R}^{r_1\times m}$ is the control weighting. If $E_u = 0$, then all expressions involving E_u in (4.23), as well as in all subsequent expressions, are omitted, and $r_1 = 0$. Using (4.19), it follows that (4.23) can be expressed as

$$z_{\mathrm{c},k}(\theta_{\mathrm{c}}) = y_{\mathrm{c},k} - \phi_{\mathrm{fc},k}\theta_{\mathrm{c}}, \tag{4.24}$$

where

$$y_{\mathrm{c},k} \triangleq \begin{bmatrix} E_z z_k - E_z u_{\mathrm{f},k} \\ 0_{r\times 1} \end{bmatrix} \in \mathbb{R}^{q+r_1}, \quad \phi_{\mathrm{fc},k} \triangleq \begin{bmatrix} -E_z\phi_{\mathrm{f},k} \\ -E_u\phi_{\mathrm{c},k} \end{bmatrix} \in \mathbb{R}^{(q+r_1)\times l_{\theta_{\mathrm{c}}}}. \tag{4.25}$$

Using (4.23), define the retrospective cost

$$J_k(\theta_{\mathrm{c}}) \triangleq \sum_{i=0}^{k} z_{\mathrm{c},i}(\theta_{\mathrm{c}})^{\mathrm{T}} z_{\mathrm{c},i}(\theta_{\mathrm{c}}) + (\theta_{\mathrm{c}} - \theta_{\mathrm{c},0})^{\mathrm{T}} P_{\mathrm{c},0}^{-1} (\theta_{\mathrm{c}} - \theta_{\mathrm{c},0}), \tag{4.26}$$

and note that

$$z_{\mathrm{c},k}(\theta_{\mathrm{c}})^{\mathrm{T}} z_{\mathrm{c},k}(\theta_{\mathrm{c}}) = \hat{z}_k(\theta_{\mathrm{c}})^{\mathrm{T}} R_z \hat{z}_k(\theta_{\mathrm{c}}) + \theta_{\mathrm{c}}^{\mathrm{T}} \phi_{\mathrm{c},k}^{\mathrm{T}} R_u \phi_{\mathrm{c},k} \theta_{\mathrm{c}}, \tag{4.27}$$

where $R_z \triangleq E_z^{\mathrm{T}} E_z \in \mathbb{R}^{q \times q}$ is positive definite and $R_u \triangleq E_u^{\mathrm{T}} E_u \in \mathbb{R}^{m \times m}$ is positive semidefinite. For all $k \geq 0$, the minimizer $\theta_{\mathrm{c},k+1}$ of (4.26) is given by the recursive least squares (RLS) solution [16]

$$P_{\mathrm{c},k+1} = P_{\mathrm{c},k} - P_{\mathrm{c},k} \phi_{\mathrm{fc},k}^{\mathrm{T}} (I_{q+r_1} + \phi_{\mathrm{fc},k} P_{\mathrm{c},k} \phi_{\mathrm{fc},k}^{\mathrm{T}})^{-1} \phi_{\mathrm{fc},k} P_{\mathrm{c},k}, \tag{4.28}$$

$$\theta_{\mathrm{c},k+1} = \theta_{\mathrm{c},k} + P_{\mathrm{c},k+1} \phi_{\mathrm{fc},k}^{\mathrm{T}} (y_{\mathrm{c},k} - \phi_{\mathrm{fc},k} \theta_{\mathrm{c},k}). \tag{4.29}$$

Using the updated controller coefficient vector given by (4.29), the requested control at step $k+1$ is given by

$$u_{k+1} = \phi_{\mathrm{c},k+1} \theta_{\mathrm{c},k+1}. \tag{4.30}$$

Although $\theta_{\mathrm{c},0}$ can be chosen arbitrarily, $\theta_{\mathrm{c},0} = 0$ is chosen in all examples in order to reflect the absence of additional modeling information. Finally, $P_{\mathrm{c},0} = p_{\mathrm{c},0} I_{l_{\theta_{\mathrm{c}}}}$, where $p_{\mathrm{c},0} \in (0, \infty)$ is a tuning parameter.

CHAPTER 5

Retrospective PerformanceVariable Decomposition

This chapter shows that the retrospective performance variable can be decomposed into the sum of a performance term and a model-matching term. A more restrictive version of the results in this section is given in [51] and a more detailed treatment is presented in [33]. For simplicity, this section focuses on the case where $\tilde{y}_k \triangleq z_k$.

Since the optimized controller coefficient vector is time-dependent, the retrospective performance variable defined by (4.19) must be modified to ignore the time-dependence of $\theta_{\mathrm{c},k+1}$. To do this, the terms $u_{\mathrm{f},k} - \phi_{\mathrm{f},k}\theta_{\mathrm{c}}$ in (4.19) are replaced by a filtered version of $u_k - \phi_{\mathrm{c},k}\theta_{\mathrm{c}}$ in which the controller coefficient vector is constrained to be $\theta_{\mathrm{c},k+1}$ over the filtering window. By defining

$$\tilde{u}_k(\theta_{\mathrm{c}}) \triangleq u_k - \phi_{\mathrm{c},k}\theta_{\mathrm{c}}, \tag{5.1}$$

the filtered signal $\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1})$ is given by a fixed-input-argument (FIA) filter with input $\tilde{u}_k(\theta_{\mathrm{c},k+1})$ as defined by Definition 2.2 in Chapter 2. In particular, $\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1})$ is defined to be the output of the FIA filter

$$\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1}) \triangleq G_{\mathrm{f}}(\mathbf{q})\tilde{u}_k(\theta_{\mathrm{c},\overline{k+1}}), \tag{5.2}$$

which ignores the change in the argument $\theta_{\mathrm{c},k+1}$ of $\tilde{u}_k$ over the interval $[k-n_{\mathrm{f}},k]$ in accordance with retrospective optimization. Note that, by the definition of FIA filtering, the filtered signal $\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1})$ is a function of the time-dependent controller coefficient vector $\theta_{\mathrm{c},k+1}$. Equivalently, (5.2) can be written as

$$\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1}) = -D\widetilde{U}_{\mathrm{f},k} + N\widetilde{U}_k(\theta_{\mathrm{c},k+1}), \tag{5.3}$$

where

$$\widetilde{U}_{\mathrm{f},k} \triangleq \begin{bmatrix} \tilde{u}_{\mathrm{f},k-1}(\theta_{\mathrm{c},k}) \\ \vdots \\ \tilde{u}_{\mathrm{f},k-n_{\mathrm{f}}}(\theta_{\mathrm{c},k-n_{\mathrm{f}}+1}) \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{f}}q}, \quad \widetilde{U}_k(\theta_{\mathrm{c}}) \triangleq \begin{bmatrix} \tilde{u}_k(\theta_{\mathrm{c}}) \\ \vdots \\ \tilde{u}_{k-n_{\mathrm{f}}}(\theta_{\mathrm{c}}) \end{bmatrix} \in \mathbb{R}^{(n_{\mathrm{f}}+1)m}. \tag{5.4}$$

Using (5.2), the definition (4.19) of $\hat{z}_k(\theta_{\mathrm{c}})$ is replaced by

$$\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) \triangleq z_k - \tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1}). \tag{5.5}$$

Using (5.1), (5.3), and (5.4), it follows that (5.5) can be written as

$$\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) = z_k + D\widetilde{U}_{\mathrm{f},k} - N(U_k - \Phi_{\mathrm{c},k}\theta_{\mathrm{c},k+1}). \tag{5.6}$$

Note that the difference between $\hat{z}_k(\theta_{\mathrm{c},k+1})$ given by (4.21) and $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1})$ given by (5.6) is due to the fact that $U_{\mathrm{f},k} - \Phi_{\mathrm{f},k}\theta_{\mathrm{c}}$ in (4.21) is replaced by $\widetilde{U}_{\mathrm{f},k}$ in (5.6). Hence, $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1})$ is not generally $\hat{z}_k(\theta_{\mathrm{c},k+1})$. However, if, for all k, $\theta_{\mathrm{c},k+1} = \theta_{\mathrm{c}}$, then $\tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1}) = u_{\mathrm{f},k} - \phi_{\mathrm{c},k}\theta_{\mathrm{c}}$, and thus $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) = \hat{z}_k(\theta_{\mathrm{c}})$.

The following result presents the *retrospective performance-variable decomposition*, which shows that the retrospective performance variable is a combination of the closed-loop performance and the extent to which the updated closed-loop transfer function from $\tilde{u}_k(\theta_{\mathrm{c},k+1})$ to z_k matches the filter $G_{\mathrm{f}}(\mathbf{q})$. Henceforth, $G_{\mathrm{f}}(\mathbf{q})$ is called the *target model* since it serves as the target for the closed-loop transfer function from $\tilde{u}_k(\theta_{\mathrm{c},k+1})$ to z_k.

Proposition 5.1. *Assume that, for all* $k \geq 0$, $\tilde{y}_k \stackrel{\triangle}{=} z_k$, *and* $G_{\rm d}(\mathbf{q})$ *and* $G_{\rm f}(\mathbf{q})$ *are strictly proper. Then, for all* $k \geq 0$,

$$\hat{z}_{{\rm ext},k}(\theta_{{\rm c},k+1}) = z_{{\rm opp},k}(\theta_{{\rm c},k+1}) + z_{{\rm tmp},k}(\theta_{{\rm c},k+1}), \tag{5.7}$$

where the one-step predicted performance $z_{{\rm opp},k}(\theta_{{\rm c},k+1})$ *and the* target-model matching performance $z_{{\rm tmp},k}(\theta_{{\rm c},k+1})$ *are defined by*

$$z_{{\rm opp},k}(\theta_{{\rm c},k+1}) \stackrel{\triangle}{=} \widetilde{G}_{zw,k+1}(\mathbf{q})(r_k - Ev_k - E\mathcal{G}[w(t)]), \tag{5.8}$$

$$z_{{\rm tmp},k}(\theta_{{\rm c},k+1}) \stackrel{\triangle}{=} [\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q}) - G_{\rm f}(\mathbf{q})]\tilde{u}_k(\theta_{{\rm c},\overline{k+1}}), \tag{5.9}$$

and

$$\widetilde{G}_{zw,k+1}(\mathbf{q}) \stackrel{\triangle}{=} [I_q + EG_{\rm d}(\mathbf{q})G_{{\rm c},k+1}(\mathbf{q})]^{-1}, \tag{5.10}$$

$$\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q}) \stackrel{\triangle}{=} -\mathbf{q}^{n_{\rm c}}[I_q + EG_{\rm d}(\mathbf{q})G_{{\rm c},k+1}(\mathbf{q})]^{-1}EG_{\rm d}(\mathbf{q})D^{-1}_{{\rm c},k+1}(\mathbf{q}). \tag{5.11}$$

Proof. It follows from (5.8) and (5.10) that

$$z_{{\rm opp},k}(\theta_{{\rm c},k+1}) = r_k - Ev_k - E\mathcal{G}[w(t)] - EG_{\rm d}(\mathbf{q})G_{{\rm c},k+1}(\mathbf{q})z_{{\rm opp},k}(\theta_{{\rm c},k+1}). \tag{5.12}$$

Furthermore, defining the FIA filter output (see Definition 2.2 in Appendix B)

$$\tilde{z}_{{\rm tmp},k}(\theta_{{\rm c},k+1}) \stackrel{\triangle}{=} \widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})\tilde{u}_k(\theta_{{\rm c},\overline{k+1}}), \tag{5.13}$$

it follows from (5.11) and (5.13) that

$$\tilde{z}_{{\rm tmp},k}(\theta_{{\rm c},k+1}) = -EG_{\rm d}(\mathbf{q})D^{-1}_{{\rm c},k+1}(\mathbf{q})\mathbf{q}^{n_{\rm c}}\tilde{u}_k(\theta_{{\rm c},\overline{k+1}}) - EG_{\rm d}(\mathbf{q})G_{{\rm c},k+1}(\mathbf{q})\tilde{z}_{{\rm tmp},k}(\theta_{{\rm c},k+1}). \tag{5.14}$$

Now, replacing $\mathbf{q}^{n_c}\tilde{u}_k(\theta_{c,\overline{k+1}})$ with $\tilde{u}_{k+n_c}(\theta_{c,k+1})$ in (5.14) yields

$$\tilde{z}_{\mathrm{tmp},k}(\theta_{c,k+1}) = -EG_d(\mathbf{q})D_{c,k+1}^{-1}(\mathbf{q})\tilde{u}_{k+n_c}(\theta_{c,k+1}) - EG_d(\mathbf{q})G_{c,k+1}(\mathbf{q})\tilde{z}_{\mathrm{tmp},k}(\theta_{c,k+1}). \tag{5.15}$$

Combining (5.12) and (5.13) yields

$$\begin{aligned} z_{\mathrm{opp},k}(\theta_{c,k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{c,k+1}) = r_k - Ev_k - E\mathcal{G}[w(t)] - EG_d(\mathbf{q})D_{c,k+1}^{-1}(\mathbf{q})\tilde{u}_{k+n_c}(\theta_{c,k+1}) \\ - EG_d(\mathbf{q})G_{c,k+1}(\mathbf{q})[z_{\mathrm{opp},k}(\theta_{c,k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{c,k+1})]. \end{aligned} \tag{5.16}$$

Next, replacing k with $k + n_c$ in (5.1) and setting $\theta_c = \theta_{c,k+1}$ yields

$$\tilde{u}_{k+n_c}(\theta_{c,k+1}) = u_{k+n_c} - \phi_{c,k+n_c}\theta_{c,k+1}. \tag{5.17}$$

Hence, using

$$\phi_{c,k+n_c}\theta_{c,k+1} = \sum_{i=1}^{n_c} P_{i,k+1}u_{k+n_c-i} + \sum_{i=1}^{n_c} Q_{i,k+1}z_{k+n_c-i},$$

it follows from (5.17) that

$$\tilde{u}_{k+n_c}(\theta_{c,k+1}) = u_{k+n_c} - \sum_{i=1}^{n_c} P_{i,k+1}u_{k+n_c-i} - \sum_{i=1}^{n_c} Q_{i,k+1}z_{k+n_c-i}. \tag{5.18}$$

Using (4.6) and (4.7), note that (5.18) can be written as

$$\tilde{u}_{k+n_c}(\theta_{c,k+1}) = D_{c,k+1}(\mathbf{q})u_k - N_{c,k+1}(\mathbf{q})z_k,$$

which can be combined with (5.16) to obtain

$$z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}) = r_k - Ev_k - E\mathcal{G}[w(t)] - EG_{\mathrm{d}}(\mathbf{q})u_k$$
$$+ EG_{\mathrm{d}}(\mathbf{q})G_{\mathrm{c},k+1}(\mathbf{q})z_k - EG_{\mathrm{d}}(\mathbf{q})G_{\mathrm{c},k+1}(\mathbf{q})[z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})]. \quad (5.19)$$

Using (3.6) and (3.7), it follows from (5.19) that

$$(I_q + EG_{\mathrm{d}}(\mathbf{q})G_{\mathrm{c},k+1}(\mathbf{q}))[z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})] = (I_q + EG_{\mathrm{d}}(\mathbf{q})G_{\mathrm{c},k+1}(\mathbf{q}))z_k, \quad (5.20)$$

which implies that

$$z_k = z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}). \quad (5.21)$$

Next, substituting (5.21) into (5.5) yields

$$\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) = z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \tilde{z}_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}) - \tilde{u}_{\mathrm{f},k}(\theta_{\mathrm{c},k+1}). \quad (5.22)$$

Hence, substituting (5.2) and (5.13) into (5.22) and using (5.9) yields

$$\begin{aligned}\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) &= z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + \widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})\tilde{u}_k(\theta_{\mathrm{c},\overline{k+1}}) - G_{\mathrm{f}}(\mathbf{q})\tilde{u}_k(\theta_{\mathrm{c},\overline{k+1}}) \\ &= z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + [\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q}) - G_{\mathrm{f}}(\mathbf{q})]\tilde{u}_k(\theta_{\mathrm{c},\overline{k+1}}) \\ &= z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}). \qquad \square\end{aligned}$$

In the case where $\tilde{y}_k = z_k$, y_k, and u_k are scalar, that is, $l_y = q = p = m = 1$, (5.10)

and (5.11) have the form

$$\widetilde{G}_{zw,k+1}(\mathbf{q}) = \frac{D_{\mathrm{d}}(\mathbf{q})D_{\mathrm{c},k+1}(\mathbf{q})}{D_{\mathrm{d}}(\mathbf{q})D_{\mathrm{c},k+1}(\mathbf{q}) + EN_{\mathrm{d}}(\mathbf{q})N_{\mathrm{c},k+1}(\mathbf{q})}, \tag{5.23}$$

$$\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q}) = \frac{-\mathbf{q}^{n_{\mathrm{c}}}EN_{\mathrm{d}}(\mathbf{q})}{D_{\mathrm{d}}(\mathbf{q})D_{\mathrm{c},k+1}(\mathbf{q}) + EN_{\mathrm{d}}(\mathbf{q})N_{\mathrm{c},k+1}(\mathbf{q})}, \tag{5.24}$$

where

$$G_{\mathrm{d}}(\mathbf{q}) \triangleq \frac{N_{\mathrm{d}}(\mathbf{q})}{D_{\mathrm{d}}(\mathbf{q})}. \tag{5.25}$$

5.1 Analysis of the Retrospective Performance-Variable Decomposition

Assuming $E_z = I$, $E_u = 0$, and using (4.23) and (5.7), it follows from (4.26) that

$$J_k(\theta_{\mathrm{c},k+1}) = \sum_{i=0}^{k} \hat{z}_i^{\mathrm{T}}(\theta_{\mathrm{c},i+1})\hat{z}_{,i}(\theta_{\mathrm{c},i+1}) + (\theta_{\mathrm{c},i+1} - \theta_{\mathrm{c},0})^{\mathrm{T}}P_{\mathrm{c},0}^{-1}(\theta_{\mathrm{c},i+1} - \theta_{\mathrm{c},0}). \tag{5.26}$$

In the case where $p_{\mathrm{c},0}$ is large, using RLS to minimize (5.26) yields

$$\hat{z}_k(\theta_{\mathrm{c},k+1}) \approx 0. \tag{5.27}$$

Furthermore, it is observed numerically and shown in Figure 5.2 that using RLS to minimize (5.26) yields

$$\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1}) \approx \hat{z}_k(\theta_{\mathrm{c},k+1}), \tag{5.28}$$

which, using (5.7), implies that

$$z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) + z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}) \approx 0, \tag{5.29}$$

that is,

$$z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) \approx -z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1}). \tag{5.30}$$

The following example illustrates this property.

Example 4. *Minimization of $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1})$ and its decomposition for a SISO System.* Let

$$G_u(s) = \frac{100(s-10)(s+8)}{(s+11)(s^2-0.6s+900)}, \tag{5.31}$$

and, for $T_\mathrm{s} = 0.01$ s/step, let $G_\mathrm{d}(\mathbf{q})$ denote the ZOH discretization of $G_u(s)$. Assume that the w is matched, that is, $G_u(s) = G_w(s)$, and let $\overline{w}_{k,i}$ be zero-mean, Gaussian white noise with standard deviation 1. For disturbance rejection with nonnoisy measurements, that is, with $r_k = 0$ and $v_k = 0$, adaptive control is applied with $E_z = 1$, $E_u = 0$, $E = 1$, $G_\mathrm{f}(\mathbf{q}) = -0.9988\frac{(\mathbf{q}-1.1628)}{\mathbf{q}^2}$, $n_\mathrm{c} = 16$, and $p_{\mathrm{c},0} = 10$. Figures 5.1(f) and (h) shows that, for all $0.04 \le t \le 0.7$, $z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1})$ and $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ have large magnitudes and approximately sum to zero. In particular, Figure 5.1(h) shows $\frac{|z_{\mathrm{opp},k}+z_{\mathrm{tmp},k}|}{|z_{\mathrm{opp},k}|+|z_{\mathrm{tmp},k}|}$, which is small when $z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1})$ and $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ have large magnitudes with opposite signs, and close to 1 when $z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1})$ and $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ have small magnitudes. Figure 5.1(g) shows that $\widetilde{G}_{z\tilde{u},400}(\mathbf{q})$ and $G_\mathrm{f}(\mathbf{q})$ have similar frequency responses, and thus the controller update promotes matching between the closed-loop transfer function $\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})$ and the target model $G_\mathrm{f}(\mathbf{q})$.

Next, in order to compare $\hat{z}_k(\theta_{\mathrm{c},k+1})$ and $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1})$ for the case where $G_\mathrm{f}(\mathbf{q})$ is IIR, the simulation is repeated with $G_\mathrm{f}(\mathbf{q}) = -0.9988\frac{(\mathbf{q}-1.1628)}{\mathbf{q}^2+0.1\mathbf{q}+0.01}$. Figure 5.2 shows that the error between $\hat{z}_k(\theta_{\mathrm{c},k+1})$ and $\hat{z}_{\mathrm{ext},k(\theta_{\mathrm{c},k+1})}$ is less than 10^{-1} for all t. ◇

Proposition 5.2. *Assume that $\overline{\theta}_\mathrm{c} \stackrel{\triangle}{=} \lim_{k\to\infty} \theta_{\mathrm{c},k+1}$ exists and $\phi_{\mathrm{c},k+1}$ is bounded. Then* $\lim_{k\to\infty} \tilde{u}_k(\theta_{\mathrm{c},k+1}) = 0$.

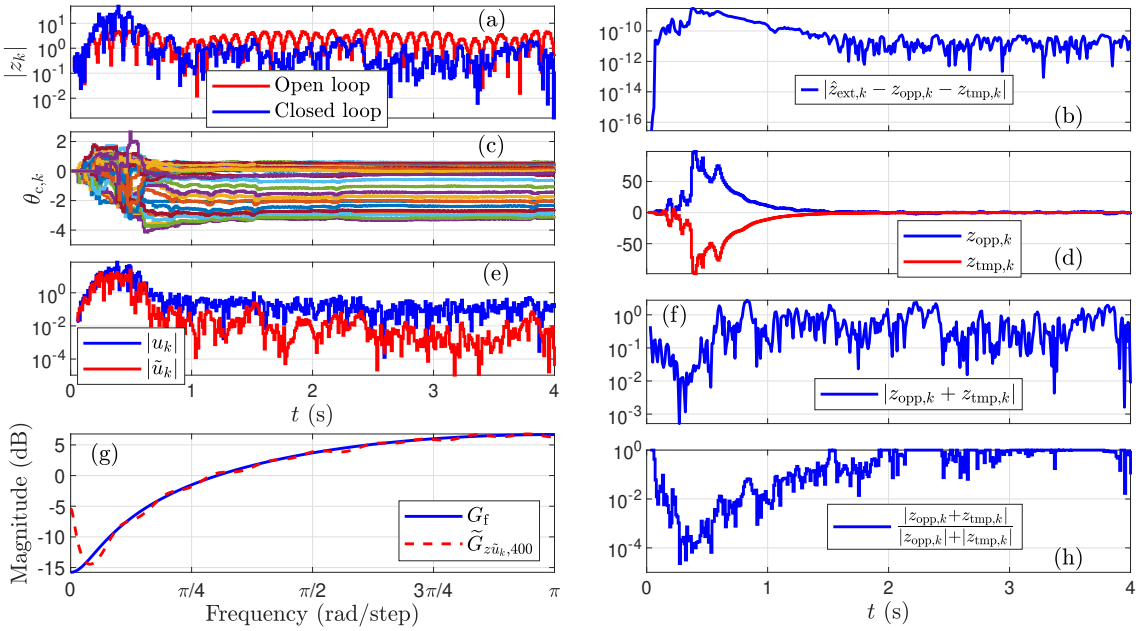

Figure 5.1: Example 4: (a) open- and closed-loop responses; (b) that $|\hat{z}_k - z_{\mathrm{opp},k} - z_{\mathrm{tmp},k}| < 3.01 \times 10^{-9}$ for all t, which confirms (5.7); (c) the evolution of $\theta_{\mathrm{c},k}$; (d) $z_{\mathrm{opp},k}$ and $z_{\mathrm{tmp},k}$; (e) $|u_k|$ and $|\tilde{u}_k|$; (f) $|\hat{z}_{\mathrm{ext},k}|$; (g) the frequency response of $G_{\mathrm{f}}(\mathbf{q})$ and $\widetilde{G}_{z\tilde{u},400}(\mathbf{q})$; (h) $\frac{|z_{\mathrm{opp},k}+z_{\mathrm{tmp},k}|}{|z_{\mathrm{opp},k}|+|z_{\mathrm{tmp},k}|}$.

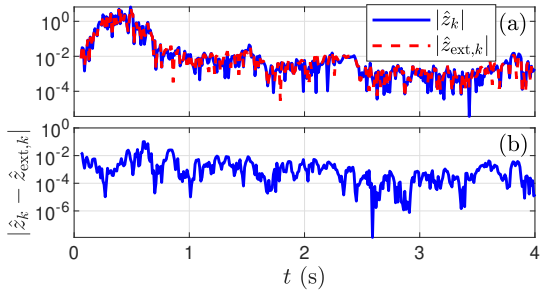

Figure 5.2: Example 4: For an IIR $G_{\mathrm{f}}(\mathbf{q})$, (a) shows the absolute value of the retrospective cost variable and its extension, and (b) shows the absolute error between the retrospective cost variable and its extension.

Proof. Equations (4.3) and (5.1) imply that

$$\tilde{u}_k(\theta_{\mathrm{c},k+1}) = \phi_{\mathrm{c},k}(\theta_{\mathrm{c},k} - \theta_{\mathrm{c},k+1}).$$

Defining $\alpha = \sup_{k \geq 0} \sigma_{\max}(\phi_{\mathrm{c},k})$, it follows that

$$\begin{aligned} \|\tilde{u}_k(\theta_{\mathrm{c},k+1})\| &\leq \sigma_{\max}(\phi_{\mathrm{c},k}) \|\theta_{\mathrm{c},k} - \theta_{\mathrm{c},k+1}\| \\ &\leq \alpha \|\theta_{\mathrm{c},k} - \theta_{\mathrm{c},k+1}\|, \end{aligned}$$

where $\sigma_{\max}$ denotes the maximum singular value. Hence,

$$\lim_{k \to \infty} \|\tilde{u}_k(\theta_{\mathrm{c},k+1})\| \leq \alpha \lim_{k \to \infty} \|\theta_{\mathrm{c},k} - \theta_{\mathrm{c},k+1}\| = 0. \qquad \square$$

Proposition 5.2 and (5.9) suggest that the convergence of $\theta_{\mathrm{c},k}$ implies that $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ converges to zero, as illustrated in Figure 5.1(g). Therefore, (5.30) implies that $|z_{\mathrm{opp},k}(\overline{\theta}_{\mathrm{c}})| \approx 0$, and thus, if $\theta_{\mathrm{c},k}$ converges, then the one-step predicted performance $|z_{\mathrm{opp},k}(\overline{\theta}_{\mathrm{c}})|$ is small. This mechanism underlies the convergence of RCAC in Figure 5.1 to a stabilizing controller that rejects the unknown disturbance. Note, however, that the convergence of $\theta_{\mathrm{c},k}$ and the consequent convergence of $\tilde{u}_k(\theta_{\mathrm{c},k+1})$ to zero do not imply that $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ converges to zero. In fact, Example 8 demonstrates that a poor choice of $G_{\mathrm{f}}(\mathbf{q})$ may cause $z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$ to diverge while $\theta_{\mathrm{c},k}$ converges.

5.2 Feasibility of $G_{\mathrm{f}}(\mathbf{q})$

The following definition concerns the case where there exists a controller parameter vector that exactly matches the transfer function $\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})$ to $G_{\mathrm{f}}(\mathbf{q})$.

Definition 5.1. *Assume that, for all $k \geq 0$, $\tilde{y}_k = z_k \in \mathbb{R}^q$. Then, $G_{\mathrm{f}}(\mathbf{q}) \in \mathbb{R}(\mathbf{q})^{q \times m}_{\mathrm{prop}}$ is* feasible *if there exists $\theta_{\mathrm{c}} = \mathrm{vec}\left[\, P_1 \cdots P_{n_{\mathrm{c}}} \; Q_1 \cdots Q_{n_{\mathrm{c}}} \,\right] \in \mathbb{R}^{l_{\theta_{\mathrm{c}}}}$ such that*

$$\widetilde{G}_{z\tilde{u}}(\mathbf{q}) = G_{\mathrm{f}}(\mathbf{q}), \tag{5.32}$$

where

$$\widetilde{G}_{z\tilde{u}}(\mathbf{q}) \triangleq -\mathbf{q}^{n_{\mathrm{c}}}[I_q + EG_{\mathrm{d}}(\mathbf{q})G_{\mathrm{c}}(\mathbf{q})]^{-1}EG_{\mathrm{d}}(\mathbf{q})D_{\mathrm{c}}(\mathbf{q})^{-1}, \tag{5.33}$$

with

$$D_{\mathrm{c}}(\mathbf{q}) \triangleq I_m\mathbf{q}^{n_{\mathrm{c}}} - P_1\mathbf{q}^{n_{\mathrm{c}}-1} - \cdots - P_{n_{\mathrm{c}}}, \tag{5.34}$$

$$N_{\mathrm{c}}(\mathbf{q}) \triangleq Q_1\mathbf{q}^{n_{\mathrm{c}}-1} + \cdots + Q_{n_{\mathrm{c}}}, \tag{5.35}$$

$$G_{\mathrm{c}}(\mathbf{q}) \triangleq D_{\mathrm{c}}^{-1}(\mathbf{q})N_{\mathrm{c}}(\mathbf{q}). \tag{5.36}$$

Definition 5.2. *Let $\theta_{\mathrm{c},k}$ be given by* (4.29)*, and $\widetilde{G}_{z\tilde{u},k}(\mathbf{q})$ be given by* (5.24)*. Then the* asymptotic feasibility distance *is*

$$\mathrm{f}_\infty \triangleq \limsup_{k\to\infty} \|\widetilde{G}_{z\tilde{u},k}(\mathbf{q}) - G_{\mathrm{f}}(\mathbf{q})\|_\infty. \tag{5.37}$$

For the SISO case, the following result identifies several features of $\widetilde{G}_{z\tilde{u}}(\mathbf{q})$ that are determined by $G_{\mathrm{d}}(\mathbf{q})$.

Proposition 5.3. *For all $k \geq 0$, assume that $\tilde{y}_k = z_k$, y_k, and u_k are scalar. Furthermore, let $\theta_{\mathrm{c}} \in \mathbb{R}^{l_{\theta_{\mathrm{c}}}}$ and $G_{\mathrm{f}}(\mathbf{q}) \in \mathbb{R}(\mathbf{q})_{\mathrm{prop}}$. Then the following statements hold:*

i) The leading numerator coefficient of $\widetilde{G}_{z\tilde{u}}(\mathbf{q})$ is equal to the leading numerator coefficient of $-EG_{\mathrm{d}}(\mathbf{q})$.

ii) The relative degree of $\widetilde{G}_{z\tilde{u}}(\mathbf{q})$ is equal to the relative degree of $G_{\mathrm{d}}(\mathbf{q})$.

iii) The zeros of $\widetilde{G}_{z\tilde{u}}(\mathbf{q})$ consist of the zeros of $G_{\mathrm{d}}(\mathbf{q})$ as well as n_{c} zeros at zero.

Proof. Since $\tilde{y}_k = z_k$ and u_k are scalar, it follows that E is scalar and the closed-loop

transfer function (5.33) specializes to

$$\widetilde{G}_{z\tilde{u}}(\mathbf{q}) = \frac{-\mathbf{q}^{n_\mathrm{c}} E N_\mathrm{d}(\mathbf{q})}{D_\mathrm{d}(\mathbf{q}) D_\mathrm{c}(\mathbf{q}) + E N_\mathrm{d}(\mathbf{q}) N_\mathrm{c}(\mathbf{q})}, \tag{5.38}$$

which implies *i*). To prove *ii*), let $\mathrm{d_d}$ denote the degree of $D_\mathrm{d}(\mathbf{q})$, and let $\xi \geq 0$ denote the relative degree of $G_\mathrm{d}(\mathbf{q})$, so that the degree of $N_\mathrm{d}(\mathbf{q})$ is $\mathrm{d_d} - \xi$. Since the degree of $\mathbf{q}^{n_\mathrm{c}} E N_\mathrm{d}(\mathbf{q})$ is $n_\mathrm{c} + \mathrm{d_d} - \xi$ and the degree of $D_\mathrm{d}(\mathbf{q}) D_\mathrm{c}(\mathbf{q}) + E N_\mathrm{d}(\mathbf{q}) N_\mathrm{c}(\mathbf{q})$ is $n_\mathrm{c} + \mathrm{d_d}$, it follows that the relative degree of $\widetilde{G}_{z\tilde{u}}(\mathbf{q})$ is ξ. Finally, *iii*) follows from the fact that the numerator of (5.38) is the numerator of $E G_\mathrm{d}(\mathbf{q})$ multiplied by $\mathbf{q}^{n_\mathrm{c}}$. □

The following result, which is an immediate consequence of Proposition 5.3, provides necessary conditions for feasibility in the SISO case.

Proposition 5.4. *For all $k \geq 0$, assume that $\tilde{y}_k = z_k$, y_k, and u_k are scalar. Furthermore, let $\theta_\mathrm{c} \in \mathbb{R}^{l_{\theta_\mathrm{c}}}$, let $G_\mathrm{f}(\mathbf{q}) \in \mathbb{R}(\mathbf{q})_\mathrm{prop}$, and assume that $G_\mathrm{f}(\mathbf{q})$ is feasible. Then the following statements hold:*

$i)$ *The leading numerator coefficient of $G_\mathrm{f}(\mathbf{q})$ is equal to the leading numerator coefficient of $-E G_\mathrm{d}(\mathbf{q})$.*

$ii)$ *The relative degree of $G_\mathrm{f}(\mathbf{q})$ is equal to the relative degree of $G_\mathrm{d}(\mathbf{q})$.*

$iii)$ *The zeros of $G_\mathrm{f}(\mathbf{q})$ consist of the zeros of $G_\mathrm{d}(\mathbf{q})$, as well as n_c zeros at zero.*

5.3 RCAC with Feasible and Infeasible $G_\mathrm{f}(\mathbf{q})$ for SISO Systems

This subsection investigates the effect of feasible and infeasible target models on the convergence of $\theta_{\mathrm{c},k}$ given by (4.29). For all of the examples in this and the following subsection, let $G_u(s)$ be given by (5.31), and, for $T_\mathrm{s} = 0.01$ s/step, let $G_\mathrm{d}(\mathbf{q})$ denote the

ZOH discretization of $G_u(s)$. In particular,

$$G_{\mathrm{d}}(\mathbf{q}) = \frac{0.9988(\mathbf{q} - 1.1628)(\mathbf{q} - 0.7393)}{(\mathbf{q} - 0.9048)(\mathbf{q}^2 - 1.905\mathbf{q} + 0.994)}. \tag{5.39}$$

Assume that w is matched, that is, $G_u(s) = G_w(s)$, and let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 1 and 0.01, respectively. For various choices of the target model $G_{\mathrm{f}}(\mathbf{q})$, the following examples consider disturbance rejection with noisy measurements with $r_k = 0$, $E_z = 1$, $E_u = 0$, and $E = 1$.

Example 5. *Feasible* $G_{\mathrm{f}}(\mathbf{q})$. A linear-quadratic-Gaussian (LQG) controller $G_{\mathrm{LQG}}(\mathbf{q})$ is designed for $G_{\mathrm{d}}(\mathbf{q})$ given by (5.39) using the MATLAB command lqg with $Q_{xu} = I_4$ and $Q_{wv} = I_4$. The LQG controller

$$G_{\mathrm{LQG}}(\mathbf{q}) \triangleq \frac{N_{\mathrm{LQG}}(\mathbf{q})}{D_{\mathrm{LQG}}(\mathbf{q})}, \tag{5.40}$$

is used to construct

$$G_{\mathrm{f,LQG}}(\mathbf{q}) = \frac{-\mathbf{q}^n N_{\mathrm{d}}(\mathbf{q})}{D_{\mathrm{d}}(\mathbf{q}) D_{\mathrm{LQG}}(\mathbf{q}) + N_{\mathrm{d}}(\mathbf{q}) N_{\mathrm{LQG}}(\mathbf{q})}. \tag{5.41}$$

The corresponding closed-loop target model is given by

$$G_{\mathrm{f,LQG}}(\mathbf{q}) = \frac{-0.9988\mathbf{q}^3(\mathbf{q} - 1.1628)(\mathbf{q} - 0.7393)}{(\mathbf{q} - 0.888)(\mathbf{q} - 0.212)(\mathbf{q}^2 - 1.199\mathbf{q} + 0.3738)(\mathbf{q}^2 - 0.0926\mathbf{q} + 0.1148)}, \tag{5.42}$$

Note that (5.42) is feasible by construction. Since $G_{\mathrm{f,LQG}}(\mathbf{q})$ is feasible, Proposition 5.4 implies that its leading numerator coefficient -0.9988 and relative degree 1 are the same as those of $-EG_{\mathrm{d}}(\mathbf{q})$ and that its zeros 0, 0.7393 and 1.1628 are the zeros of $G_{\mathrm{d}}(\mathbf{q})$ as well as $n = 3$ zeros at zero. Next, adaptive control is applied with $G_{\mathrm{f}}(\mathbf{q}) = G_{\mathrm{f,LQG}}(\mathbf{q})$, $p_{\mathrm{c},0} = 10^7$, and $n_{\mathrm{c}} = n = 3$. Figure 5.3(d) shows that $\widetilde{G}_{z\tilde{u},1000}(\mathbf{q})$ and $G_{\mathrm{f}}(\mathbf{q})$ have similar

frequency responses, which is consistent with the fact that $G_{\mathrm{f,LQG}}(\mathbf{q})$ is feasible. Moreover, Figure 5.3(c) shows that $G_{\mathrm{c},1000}(\mathbf{q})$ and $G_{\mathrm{LQG}}(\mathbf{q})$ have similar frequency responses, which suggests that the adaptive controller approximately converges to the LQG controller. $\diamond$

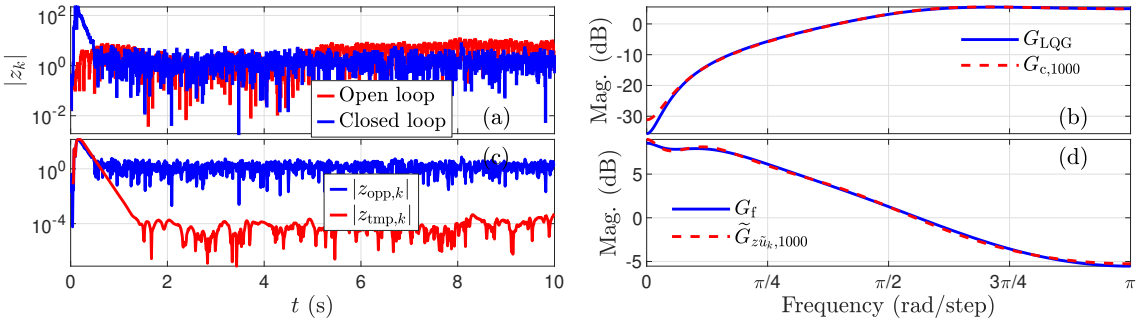

Figure 5.3: Example 5: (a) open- and closed-loop responses; (b) frequency response of $G_{\mathrm{LQG}}(\mathbf{q})$ and $G_{\mathrm{c},1000}(\mathbf{q})$; (c) $|z_{\mathrm{opp},k}|$ and $|z_{\mathrm{tmp},k}|$; (d) frequency response of $G_{\mathrm{f}}(\mathbf{q})$ and $\widetilde{G}_{z\tilde{u},1000}(\mathbf{q})$.

Example 6. *Robustness to infeasible* $G_{\mathrm{f}}(\mathbf{q})$. To investigate the robustness of the feasible target model (5.42), the target model is chosen to be various infeasible perturbations of the feasible target model given by

$$G_{\mathrm{f}}(\mathbf{q}) = \alpha_{\mathrm{LNC}} G_{\mathrm{f,LQG}}(\mathbf{q}), \tag{5.43}$$

$$G_{\mathrm{f}}(\mathbf{q}) = \frac{1}{\mathbf{q}^{\alpha_{\mathrm{RD}}}} G_{\mathrm{f,LQG}}(\mathbf{q}), \tag{5.44}$$

$$G_{\mathrm{f}}(\mathbf{q}) = \frac{-0.9988\mathbf{q}^3(\mathbf{q}-1.1628)(\mathbf{q}-\alpha_{\mathrm{MP}})}{(\mathbf{q}-0.8878)(\mathbf{q}-0.2118)(\mathbf{q}^2-1.199\mathbf{q}+0.3738)(\mathbf{q}^2-0.0926\mathbf{q}+0.1148)}, \tag{5.45}$$

$$G_{\mathrm{f}}(\mathbf{q}) = \frac{-0.9988\mathbf{q}^3(\mathbf{q}-\alpha_{\mathrm{NMP}})(\mathbf{q}-0.7393)}{(\mathbf{q}-0.8878)(\mathbf{q}-0.2118)(\mathbf{q}^2-1.199\mathbf{q}+0.3738)(\mathbf{q}^2-0.0926\mathbf{q}+0.1148)}, \tag{5.46}$$

which reflect uncertainty in α_{LNC}, α_{RD}, α_{MP}, and α_{NMP}, respectively. Note that (5.43), (5.44), (5.45), and (5.46) are equal to (5.42) for the nominal values $\alpha_{\mathrm{LNC}} = 1$, $\alpha_{\mathrm{RD}} = 0$, $\alpha_{\mathrm{MP}} = 0.7393$, and $\alpha_{\mathrm{NMP}} = 1.1628$, respectively.

The suppression metric g_{s} is defined as the ratio of the root-mean-square of the last 1000

subinterval steps of the open-loop response and the closed-loop response in dB. The case $g_{\rm s} > 0$ corresponds to disturbance suppression relative to the response of the open-loop system. Simulations where either $g_{\rm s} \leq 0$ or the output of the closed-loop system diverges are indicated as failures.

To investigate the closed-loop performance with an off-nominal target model, $\alpha_{\rm LNC}$, $\alpha_{\rm RD}$, $\alpha_{\rm MP}$, and $\alpha_{\rm NMP}$ are varied from their nominal values, and RCAC is applied with $n_{\rm c} = n = 3$, $p_{\rm c,0} = 1000$, for $0 \leq t \leq 20$ s. Figure 5.4 shows that the adaptive controller can be applied with the target models (5.43)–(5.46), where $\alpha_{\rm LNC}$, $\alpha_{\rm MP}$, and $\alpha_{\rm NMP}$ are off-nominal. In particular, Figure 5.4 shows the suppression metric $g_{\rm s}$ and asymptotic feasibility distance f_∞ for target models with various sources of infeasibility. Figures 5.4(a) and 5.4(e) show $g_{\rm s}$ and f_∞, respectively, for (5.43), where $\alpha_{\rm LNC} \in [-0.5, 6]$, which shows that infeasibility due to the sign of the leading numerator coefficient of the target model causes failure. However, the adaptive controller is robust to infeasibility due to the magnitude of the leading numerator coefficient of the target model. Figures 5.4(b) and 5.4(f) show $g_{\rm s}$ and f_∞, respectively, for (5.44), where $\alpha_{\rm RD} \in \{0, 1, 2, 3\}$, which shows that infeasibility due to the relative degree of target model causes failure. Figures 5.4(c) and 5.4(g) show $g_{\rm s}$ and f_∞, respectively, for (5.45), where $\alpha_{\rm MP} \in [-1.2, 1.2]$, which shows that the adaptive controller is robust to infeasibility due to an incorrectly modeled MP zero in the target model. However, note that the adaptive controller fails when a MP zero of $G_{\rm d}(\mathbf{q})$ is replaced with a positive NMP zero in the target model. Figures 5.4(d) and 5.4(h) show $g_{\rm s}$ and f_∞, respectively, for (5.46), where $\alpha_{\rm NMP} \in [0.9, 1.5]$, which shows that the adaptive controller is robust to infeasibility due to an incorrectly modeled NMP zero in the target model. Note that the adaptive controller fails when $\alpha_{\rm NMP} < 1$ in the target model (5.46), that is, when the NMP zero in the feasible target model (5.42) is replaced with a MP zero. $\diamond$

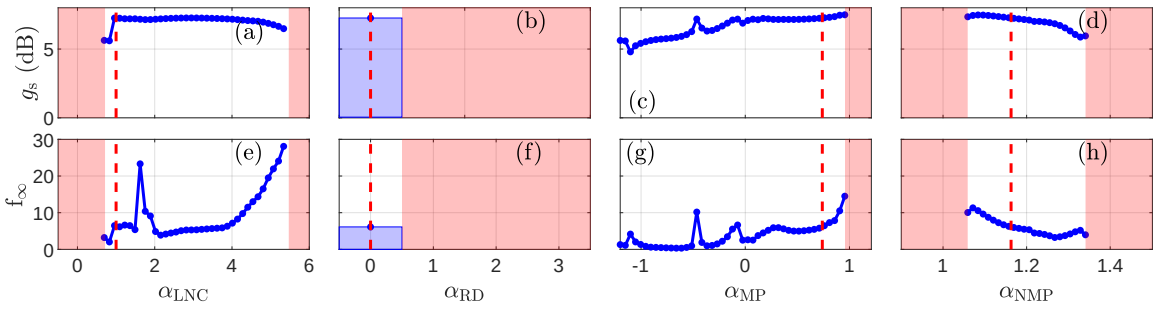

Figure 5.4: Example 6: For $G_f(\mathbf{q})$ given by (5.43)–(5.46), (a)–(d) show g_s, and (e)–(h) show f_∞. The dashed lines indicate nominal values of α_{LNC}, α_{RD}, α_{MP}, and α_{NMP}; the shaded regions indicate values for which $g_s \leq 0$.

5.4 Construction of $G_f(\mathbf{q})$ for SISO Systems

Example 6 shows that RCAC can reject disturbances with an infeasible $G_f(\mathbf{q})$ as long as $G_f(\mathbf{q})$ shares certain properties with $-EG_d(\mathbf{q})$, as described by the following definition.

Definition 5.3. *Assume that $EG_d(\mathbf{q})$ is SISO, and let $G_f(\mathbf{q})$ be a proper SISO transfer function. Then $G_f(\mathbf{q})$ is quasi-feasible if the following statements hold:*

i) The leading numerator coefficients of $G_f(\mathbf{q})$ and $-EG_d(\mathbf{q})$ have the same sign.

ii) $G_f(\mathbf{q})$ and $-EG_d(\mathbf{q})$ have the same relative degree.

iii) $G_f(\mathbf{q})$ and $-EG_d(\mathbf{q})$ have the same NMP zeros.

Note that a quasi-feasible target model may be feasible; however, most quasi-feasible target model are infeasible

Definition 5.4. *The nominal target model is the minimal-order, quasi-feasible FIR target model whose leading numerator coefficient is equal to the leading numerator coefficient of $-EG_d(\mathbf{q})$.*

Note that the nominal target model is uniquely defined. Furthermore, the nominal target model may be feasible; however, in most cases, the nominal target model is infeasible The rationale for choosing the nominal target model to be FIR is the fact that the target location

for each closed-loop pole is the center of the open unit disk. For details, see [24]. Note that the nominal target model for $-EG_{\mathrm{d}}(\mathbf{q})$, with $G_{\mathrm{d}}(\mathbf{q})$ given by (5.39), is

$$G_{\mathrm{f,n}}(\mathbf{q}) = -0.9988\frac{\mathbf{q} - 1.1628}{\mathbf{q}^2}. \tag{5.47}$$

The following example investigates the efficacy of the nominal target model when the required modeling information is uncertain.

Example 7. *Robustness to perturbations from the nominal target model.* To investigate the robustness of the nominal target model, first consider the case where $G_{\mathrm{f}}(\mathbf{q})$ given by (5.47). Figure 5.5 shows the suppression metric g_{s} and the asymptotic feasibility distance f_∞ for this choice of target model, marked with the vertical red dashed lines.

Next, the target model is chosen to be a perturbation of the nominal target model given by the off-nominal target models

$$G_{\mathrm{f}}(\mathbf{q}) = \alpha_{\mathrm{LNC}} G_{\mathrm{f,n}}(\mathbf{q}), \tag{5.48}$$

$$G_{\mathrm{f}}(\mathbf{q}) = -0.9988\frac{\mathbf{q} - 1.1628}{\mathbf{q}^{2+\alpha_{\mathrm{RD}}}}, \tag{5.49}$$

$$G_{\mathrm{f}}(\mathbf{q}) = -0.9988\frac{\mathbf{q} - \alpha_{\mathrm{NMP}}}{\mathbf{q}^2}. \tag{5.50}$$

which reflect uncertainty in α_{LNC}, α_{RD}, and α_{NMP}, respectively. Note that (5.48), (5.49), and (5.50) are equal to $G_{\mathrm{f,n}}(\mathbf{q})$ for the nominal values $\alpha_{\mathrm{LNC}} = 1$, $\alpha_{\mathrm{RD}} = 0$, and $\alpha_{\mathrm{NMP}} = 1.1628$, respectively. To investigate the closed-loop performance with an off-nominal target model, α_{LNC}, α_{RD}, and α_{NMP} are varied from their nominal values, and adaptive control is applied with $n_{\mathrm{c}} = 10$, $p_{\mathrm{c},0} = 1000$, for $0 \leq t \leq 20$ s. Figure 5.5 shows that the adaptive controller can be applied with the target models $G_{\mathrm{f,LNC}}(\mathbf{q})$ and $G_{\mathrm{f,NMP}}(\mathbf{q})$, where α_{LNC} and α_{NMP} are off-nominal. $\diamond$

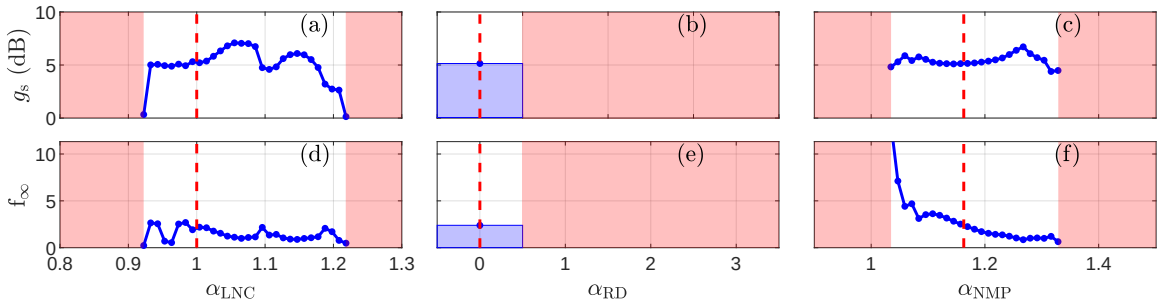

Figure 5.5: Example 7: For $G_f(\mathbf{q})$ given by (5.48)–(5.50), (a)–(c) show g_s, and (d)–(f) show f_∞. The dashed lines indicate nominal values of α_{LNC}, α_{RD}, and α_{NMP}; the shaded regions indicate values for which $g_s \leq 0$.

Example 7 suggests that $G_f(\mathbf{q})$ can be constructed as

$$G_f(\mathbf{q}) = -G_\xi \frac{\prod_{i=1}^{N_z}(\mathbf{q} - \alpha_{z,i})}{\mathbf{q}^{N_z+\xi}}, \tag{5.51}$$

where $G_\xi, \alpha_{z,i}, N_z, \xi$, are the leading numerator coefficient, all NMP zeros, number of NMP zeros, and relative degree of $EG_d(\mathbf{q})$, respectively. Note that the minus sign in (5.51) is due to the minus sign in (3.7).

Example 8. *Unmodeled NMP zeros and the retrospective performance-variable decomposition.* Let $G_f(\mathbf{q}) = -\frac{0.9988}{\mathbf{q}}$, which has the same leading numerator coefficient and relative degree as $-EG_d(\mathbf{q})$, however, it does not have the NMP zero of $G_d(\mathbf{q})$. Adaptive control is applied with $E_z = 1$, $E_u = 0$, $E = 1$, $n_c = 16$, and $p_{c,0} = 1000$.

As shown by Examples 4 and 5, the minimization of the retrospective performance variable $\hat{z}_k(\theta_{c,k+1})$ leads to matching between $\widetilde{G}_{z\tilde{u},k+1}(\theta_{c,k+1})$ and $G_f(\mathbf{q})$. Figure 5.6(h) shows that this is what happens for this example as well. Since (5.24) has a NMP zero at 1.1628 rad/step and $G_f(\mathbf{q})$ does not, the optimization attempts to cancel this NMP zero using the denominator of (5.24). This results in a controller pole at the NMP zero as shown in Figure 5.6(g), which results in a hidden instability, demonstrated by the lack of divergence of $|z_k|$ and the exponential divergence of $|u_k|$, as shown in Figures 5.6(e) and (a), respectively.

Additionally, as shown in Figure 5.6(b), the spectral radius of

$$D_u(\mathbf{q})D_\mathrm{c}(\mathbf{q}) + N_u(\mathbf{q})N_\mathrm{c}(\mathbf{q}),$$

which is the denominator polynomial of all closed-loop transfer functions, converges to a value greater than 1, which shows that all the closed-loop transfer functions are unstable. However, since $G_\mathrm{f}(\mathbf{q})$ is asymptotically stable, and $|z_k|$ and $\tilde{u}_k(\theta_{\mathrm{c},k+1})$ remain small, it follows from (5.5) that $\hat{z}_{\mathrm{ext},k}(\theta_{\mathrm{c},k+1})$ remains small, as shown in Figure 5.6(d). This in turn implies that $z_{\mathrm{opp},k}(\theta_{\mathrm{c},k+1}) \approx -z_{\mathrm{tmp},k}(\theta_{\mathrm{c},k+1})$, which can be seen in Figure 5.6(f). ◇

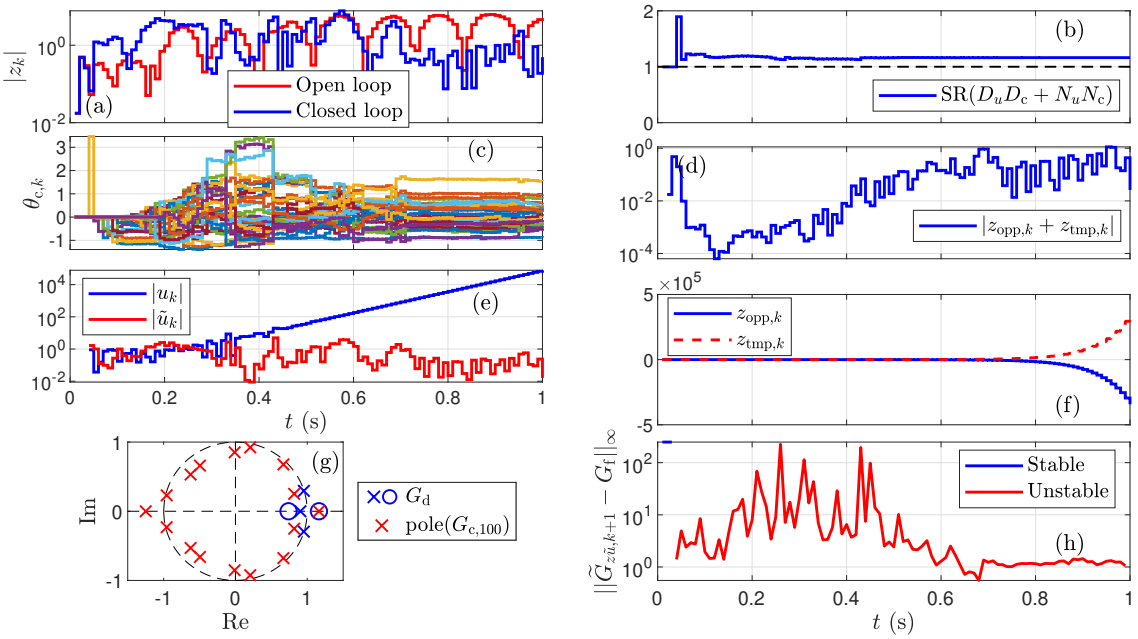

Figure 5.6: Example 8: (a) the open- and closed-loop responses; (b) the spectral radius of $D_uD_\mathrm{c} + N_uN_\mathrm{c}$; (c) the evolution of $\theta_{\mathrm{c},k}$; (d) $|\hat{z}_{\mathrm{ext},k}|$; (e) $|u_k|$ and $|\tilde{u}_k|$; (f) $z_{\mathrm{opp},k}$ and $z_{\mathrm{tmp},k}$; (g) the poles and zeros of $G_\mathrm{d}(\mathbf{q})$ and the poles of $G_\mathrm{c}(\mathbf{q})$; (h) $\|\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q}) - G_\mathrm{f}(\mathbf{q})\|_\infty$ coded by color for the stability of $\widetilde{G}_{z\tilde{u},k+1}(\mathbf{q})$.

The next example demonstrates a feature of SISO $G_\mathrm{f}(\mathbf{q})$ involving its zeros. In particular, it is numerically observed that controller poles do not converge to the locations of zeros of $G_\mathrm{f}(\mathbf{q})$.

Example 9. *Repulsion of controller poles by zeros of* $G_\mathrm{f}(\mathbf{q})$. Let the ZOH discretiza-

tion of $G_u(s)$ for $T_\mathrm{s} = 1$ s/step, be given by

$$G_\mathrm{d}(\mathbf{q}) = \frac{0.9988(\mathbf{q} - 0.1628)(\mathbf{q} - 0.7393)}{(\mathbf{q} - 0.9048)(\mathbf{q}^2 - 1.6\mathbf{q} + 0.9)}. \tag{5.52}$$

Assume that w is matched, that is, $G_u(s) = G_w(s)$, and let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 0.01 and 0.001, respectively. Consider disturbance rejection with noisy measurements with $r_k = 0$, $E_z = 1$, $E_u = 0$, and $E = 1$.

First, adaptive control is applied with $G_\mathrm{f}(\mathbf{q}) = -\frac{0.9988}{\mathbf{q}}$, as shown in Figures 5.7(a)–(d). Next, adaptive control is applied with $G_\mathrm{f}(\mathbf{q}) = -\frac{0.9988(\mathbf{q}-0.5994+0.2593\jmath)(\mathbf{q}-0.5994-0.2593\jmath)}{\mathbf{q}^3}$, as shown in Figures 5.7(e)–(h). Note that the zeros of $G_\mathrm{f}(\mathbf{q})$ are at the locations of two poles of the asymptotic controller from the first simulation. As shown in Figure 5.7(h), no controller poles converge to the location of the zeros of $G_\mathrm{f}(\mathbf{q})$. ◇

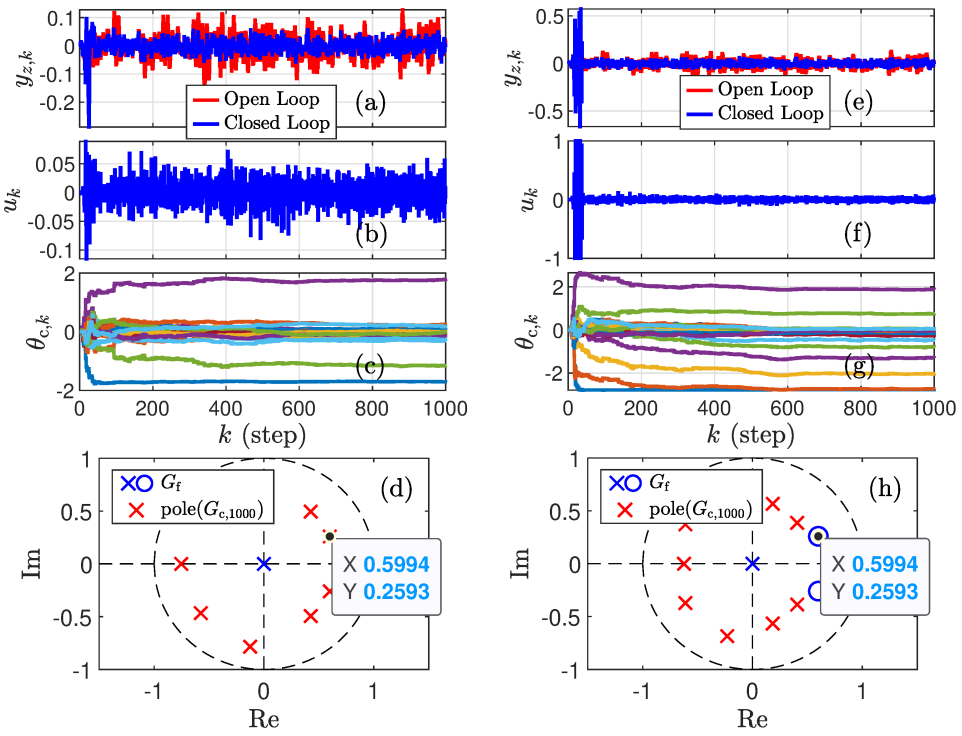

Figure 5.7: Example 9: (a) and (e) show the open- and closed-loop responses; (b) and (f) show the control; (c) and (g) show the evolution of $\theta_{\mathrm{c},k}$; (d) and (h) show the poles of the controller at $k = 1000$ and the poles and zeros of $G_\mathrm{f}(\mathbf{q})$. Note that in (h) there are no controller poles at the locations of zeros of $G_\mathrm{f}(\mathbf{q})$.

CHAPTER 6

RCAC for MIMO Control

To investigate the role of the target model $G_{\rm f}(\mathbf{q})$ in MIMO case, note that the closed-loop transfer function from r_k to y_k is given by

$$\tilde{G}_{yr}(\mathbf{q}) = [I_p + G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q})]^{-1}G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q}) \tag{6.1}$$

$$= G_{\rm d}(\mathbf{q})[I_m + G_{\rm c}(\mathbf{q})G_{\rm d}(\mathbf{q})]^{-1}G_{\rm c}(\mathbf{q}) \tag{6.2}$$

$$= G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q})[I_p + G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q})]^{-1}, \tag{6.3}$$

asssume that $G_{\rm d}(\mathbf{q})$ and $G_{\rm c}(\mathbf{q})$ have full normal rank, and consider the definitions and propositions in Appendix A. Note that, if $G_{\rm d}(\mathbf{q})$ is square, then Proposition 2.1 implies that $\mathrm{CZ}(G_{\rm d}, G_{\rm c})$ and $\mathrm{CZ}(G_{\rm c}, G_{\rm d})$ are both empty. Alternatively, consider the case where $p \neq m$, and thus $G_{\rm d}(\mathbf{q})$ in Figure 3.2 is rectangular. Note that both products $G_{\rm d}G_{\rm c} \in \mathbb{R}(\mathbf{q})^{p\times p}_{\rm prop}$ and $G_{\rm c}G_{\rm d} \in \mathbb{R}(\mathbf{q})^{m\times m}_{\rm prop}$ appear in (6.1)–(6.3). In particular, in the case where $m > p$, $G_{\rm c}(\mathbf{q})G_{\rm d}(\mathbf{q})$ is up-squared, and thus $\mathrm{CZ}(G_{\rm c}, G_{\rm d})$ is empty, whereas $G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q})$ is down-squared, and thus $\mathrm{CZ}(G_{\rm d}, G_{\rm c})$ may be nonempty. On the other hand, in the case $m < p$, $G_{\rm d}(\mathbf{q})G_{\rm c}(\mathbf{q})$ is up-squared, and thus $\mathrm{CZ}(G_{\rm d}, G_{\rm c})$ is empty, whereas $G_{\rm c}(\mathbf{q})G_{\rm d}(\mathbf{q})$ is down-squared, and thus $\mathrm{CZ}(G_{\rm c}, G_{\rm d})$ may be nonempty. As shown in the next example, cascade zeros of the down-squared loop transfer function may be cancelled by RCAC.

Example 10. *Cancellation of a NMP cascade zero for a wide system.* Consider $G_u(s)$

and $G_w(s)$ given by (3.3) and (3.4) with

$$A = \begin{bmatrix} -80 & 0 & 0 & 0 \\ 0 & -20 & 0 & 0 \\ -80 & 0 & -10 & -40 \\ -80 & 0 & 40 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} -1.8 & 1.35 & -0.85 \\ 1.02 & -0.22 & -1.12 \\ 0.13 & -0.59 & 2.53 \\ 0.71 & -0.29 & 1.66 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \tag{6.4}$$

$$C = \begin{bmatrix} 1.31 & -0.87 & 0.79 & -8.33 \\ -1.26 & -2.18 & -1.33 & -6.45 \end{bmatrix}, \quad D = 0_{2\times 3}, \tag{6.5}$$

and $T_\mathrm{s} = 0.01$ s/step. Note that A is asymptotically stable. Let $(A_\mathrm{d}, B_\mathrm{d}, C_\mathrm{d}, D_\mathrm{d})$ be a minimal realization of $G_\mathrm{d}(\mathbf{q})$. The objective is to reject the effect of a white, zero-mean, Gaussian disturbance on both components of $y_k = [y_{1,k} \;\; y_{2,k}]^\mathrm{T}$, and thus $E = I_2$. For (6.4), (6.5), $EG_\mathrm{d}(\mathbf{q})$ has no transmission zeros and no NMP channel zeros. Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 1 and 0.001, respectively. Using the Markov parameters $H_1 = C_\mathrm{d}B_\mathrm{d}$ and $H_2 = C_\mathrm{d}A_\mathrm{d}B_\mathrm{d}$ of $G_\mathrm{d}(\mathbf{q})$, let

$$G_\mathrm{f}(\mathbf{q}) = -\frac{H_1}{\mathbf{q}} - \frac{H_2}{\mathbf{q}^2}. \tag{6.6}$$

This choice of $G_\mathrm{f}(\mathbf{q})$ ensures that u_k is not restricted to a subspace of $\mathbb{R}^m$, where $m = 3$, as shown in [52]. With $G_\mathrm{f}(\mathbf{q})$ given by (6.6) and $p_{\mathrm{c},0} = 10^3$, $E_z = I_2$, $E_u = 0$, $n_\mathrm{c} = 20$, Figure 6.1 shows that a controller pole cancels a NMP cascade zero of $(G_\mathrm{d}, G_{\mathrm{c},509})$ at 1.168 rad/step, which causes the control u_k to diverge. Note that $G_\mathrm{d}(\mathbf{q})G_{\mathrm{c},509}(\mathbf{q})$ does not have a transmission zero at 1.168 rad/step due to pole-zero cancellation, and thus the zero at 1.168 rad/step is an evanescent NMP zero of $(G_\mathrm{d}, G_{\mathrm{c},509})$. $\diamond$

Example 11. *Cancellation of a NMP cascade zero for a tall system.* Consider $G_u(s)$

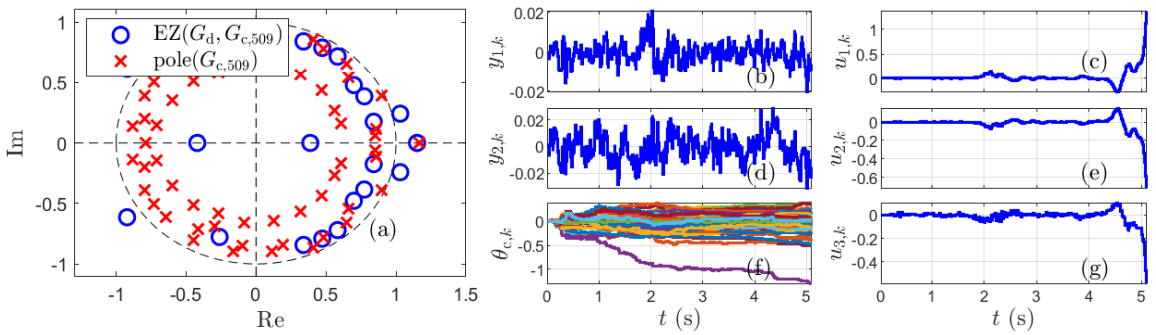

Figure 6.1: Example 10: (a) EZ($G_{\rm d}, G_{\rm c,509}$) and controller poles, where a NMP element of EZ($G_{\rm d}, G_{\rm c,509}$) is cancelled by a controller pole. (b),(d) closed-loop response; (c),(e),(g) all components of u_k diverge; (f) $\theta_{{\rm c},k}$.

and $G_w(s)$ given by (3.3) and (3.4) with

$$A = \begin{bmatrix} -120 & 0 & 0 & 0 \\ 0 & -30 & 0 & 0 \\ -80 & 0 & -15 & -60 \\ -80 & 0 & 40 & -15 \end{bmatrix}, \quad B = \begin{bmatrix} 0.231 & -0.126 \\ -0.087 & -0.318 \\ 0.079 & -0.133 \\ -1.233 & -0.645 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \tag{6.7}$$

$$C = \begin{bmatrix} -1.8 & 1.02 & 0.13 & 0.71 \\ 1.35 & -0.22 & 0.59 & -1.29 \\ -0.85 & 1.12 & -4.53 & 3.6 \end{bmatrix}, \quad D = 0_{3\times 2}, \tag{6.8}$$

and $T_{\rm s} = 0.01$ s/step. Note that A is asymptotically stable. Let $(A_{\rm d}, B_{\rm d}, C_{\rm d}, D_{\rm d})$ be a minimal realization of $G_{\rm d}(\mathbf{q})$. The objective is to reject the effect of a white, zero-mean, Gaussian disturbance on both components of $y_k = [y_{1,k} \;\; y_{2,k} \;\; y_{3,k}]^{\rm T}$, and thus $E = I_3$. For (6.7), (6.8), $EG_{\rm d}(\mathbf{q})$ has no transmission zeros and no NMP channel zeros. Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 1 and 0.001, respectively. Using the Markov parameters $H_1 = C_{\rm d}B_{\rm d}$ and $H_2 = C_{\rm d}A_{\rm d}B_{\rm d}$ of $G_{\rm d}(\mathbf{q})$, let

$$G_{\rm f}(\mathbf{q}) = -\frac{H_1}{\mathbf{q}} - \frac{H_2}{\mathbf{q}^2}. \tag{6.9}$$

This choice of $G_{\rm f}(\mathbf{q})$ ensures that u_k is not restricted to a subspace of $\mathbb{R}^m$, where $m = 2$, as shown in [52]. With $G_{\rm f}(\mathbf{q})$ given by (6.9) and $p_{{\rm c},0} = 10^5$, $E_z = I_3$, $E_u = 0$, $n_{\rm c} = 20$,

Figure 6.2 shows that controller poles cancel NMP cascade zeros of $(G_{\mathrm{c},97}, G_{\mathrm{d}})$ at 1.035 rad/step, which causes the control u_k to diverge. Note that $G_{\mathrm{c,c},97}(\mathbf{q})G_{\mathrm{d}}(\mathbf{q})$ does not have transmission zeros at the locations of the NMP cascade zeros of $(G_{\mathrm{c},97}, G_{\mathrm{d}})$ due to pole-zero cancellation, and thus these zeros are evanescent NMP zero of $(G_{\mathrm{c},97}, G_{\mathrm{d}})$. $\diamond$

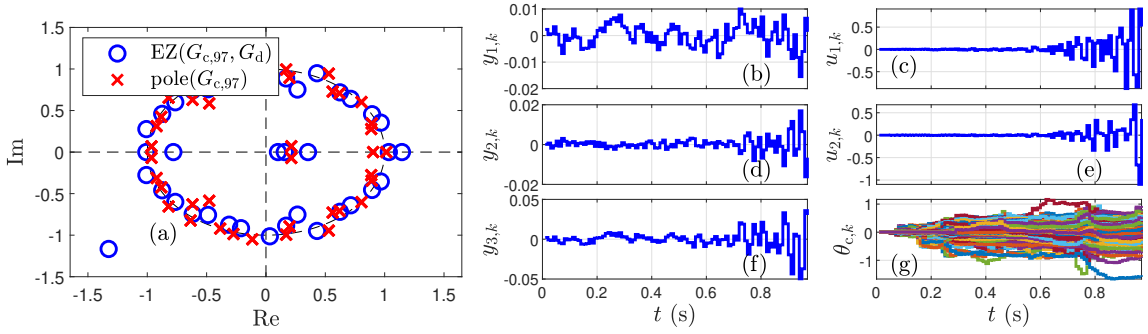

Figure 6.2: Example 11: (a) EZ$(G_{\mathrm{c},97}, G_{\mathrm{d}})$ and controller poles, where a NMP element of CZ$(G_{\mathrm{c},130}, G_{\mathrm{d}})$ is cancelled by a controller pole. (b),(d),(f) closed-loop response; (c),(e), all components of u_k diverge; (g) $\theta_{\mathrm{c},k}$.

Note that, the pole-zero cancellation in Example 8 is between an adaptive controller $G_{\mathrm{c},k}(\mathbf{q})$ and a fixed system $G_{\mathrm{d}}(\mathbf{q})$, whose zero locations are fixed. On the other hand, the pole-zero cancellations in Examples 10 and 11 are between an adaptive controller $G_{\mathrm{c},k}(\mathbf{q})$ and cascade zeroes of $(G_{\mathrm{d}}, G_{\mathrm{c},k})$ or $(G_{\mathrm{c},k}, G_{\mathrm{d}})$, respectively. Note that, the locations of cascade zeros of $(G_{\mathrm{d}}, G_{\mathrm{c},k})$ and $(G_{\mathrm{c},k}, G_{\mathrm{d}})$ depend on $G_{\mathrm{c},k}$, and thus, are not fixed. This allows for the possibility of "transient" cancellations of cascade zeros where the cancelled NMP cascade zeros transition to MP cascade zeros, or the controller poles move away from the locations of the NMP cascade zeros.

CHAPTER 7

Phasor-Based Adaptive Control of a Test-Feeder Distribution Network

This chapter presents work adapted from [53] on the adaptive MIMO control of a model of the electrical grid. An electrical grid consists of multiple interconnected components that provide and use energy. A key concern for these systems is the ability to maintain desired voltage magnitudes and angles throughout the grid in the presence of time-dependent disturbances in the form of renewable energy sources. This is a decentralized feedback control problem, where modeling information is limited by the changing topology and parameters of the grid. The authors consider command following and disturbance rejection problems for a standard grid model called the IEEE 13-node test feeder model. The model is quasi-static due to the assumption of fast transient response. Retrospective cost adaptive control (RCAC) is applied in both centralized and decentralized control architectures, where each MIMO controller has six inputs and six outputs due to the need to regulate the magnitude and phase of three-phase power at each power node. The performance of RCAC is evaluated under conditions of extremely limited model information.

7.1 IEEE13NTF Model

We consider the nonlinear, time-varying discrete-time plant

$$x_{k+1} = f(x_k, u_k, d_k, k), \tag{7.1}$$

$$y_{0,k} = g(x_k, u_k, d_k, k), \tag{7.2}$$

$$y_{\mathrm{n},k} = y_{0,k} + v_k, \tag{7.3}$$

$$z_k \stackrel{\triangle}{=} r_k - y_{\mathrm{n},k}, \tag{7.4}$$

where $k \geq 0$ is the step, $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control input, $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and $g\colon \mathbb{R}^n \to \mathbb{R}^n$ are nonlinear functions that represent the 13-Node Test Feeder (IEEE13NTF) model, $d_k \in \mathbb{R}^l$ is the disturbance, $y_{0,k} \in \mathbb{R}^p$ is plant output, $y_{\mathrm{n},k} \in \mathbb{R}^p$ is the measurement, $r_k \in \mathbb{R}^p$ is the command, $v_k \in \mathbb{R}^p$ is the sensor noise, and $z_k \in \mathbb{R}^p$ is the measured error, which is also the performance variable. m, l, and p are the dimensions of the control, disturbance, and measurement vectors, respectively. The model (7.1)–(7.4) can be viewed as a synchronously sampled, sampled-data version of IEEE13NTF shown in Figure 7.1, where the sensing, actuation, and load/generation disturbances are on physically separated nodes. IEEE13NTF is a quasi-static model that requires the iterative solution of a set of nonlinear algebraic equations. Therefore, the algebraic solution of (7.1)–(7.4) thus determine the state update and the effective impulse response.

7.2 Adaptive Control Algorithm

Despite the fact that (7.1)–(7.4) is a nonlinear, time-varying plant, RCAC uses extremely limited modeling information to update the controller coefficients. Furthermore, no explicit information about the nonlinearity or time variation is used to select the adaptive tuning weights. The goal of this study is thus to determine the extent to which RCAC can accommodate the unmodeled features of (7.1)–(7.4). For SISO linear plants RCAC

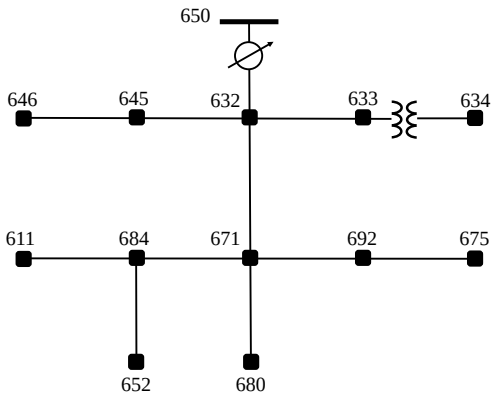

Figure 7.1: The IEEE13NTF model. The model used for this chapter is provided by the OPAL-RT RT-Lab ePHASORSIM package.

requires limited modeling data, namely, the sign of the leading numerator coefficient, the relative degree, and the locations of nonminimum-phase (NMP) zeros, if any. This modeling information is used to construct the filter $G_{\mathrm{f}}(\mathbf{q})$, which is used to compute the retrospective performance variable. For application to IEEE13NTF, the RCAC tuning weights are chosen based on a combination of nominal simulation tuning and perturbed simulation testing.

The input of $G_{\mathrm{c},k}(\mathbf{q})$ is the three voltage magnitude errors and three voltage angle errors at the performance node, and its output is the three active powers and three reactive powers extracted at the actuation node. The controller order $n_{\mathrm{c}}(\mathbf{q})$ and structure are chosen by the user, as are the adaptation weight $p_{\mathrm{c},0}$, control weight E_u, and cost weight E_z. The coefficients of $G_{\mathrm{c},k}(\mathbf{q})$ are entries of the controller coefficient vector θ_k. The initial value θ_0 is set to zero for all simulations; this assumption reflects the absence of additional modeling information. In practice, θ_0 can be chosen to be nonzero based on modeling or prior adaptation.

7.3 Problem Setup

We consider IEEE13NTF from Figure 7.1, which is simulated using OpalRT RT-LABv2017.0.4.59, and use fixed-structure RCAC described in [54] with a sparse target model G_{f}. RCAC is updated at 100 Hz, which is sufficiently fast for the time-varying load and PV generation dynamics considered in this study. In this chapter, real customer PV generation data is obtained from Pecan Street [55] and is integrated into IEEE13NTF. In what follows, continuous time t and the sampled time step k are related by $t = \frac{k}{100}$.

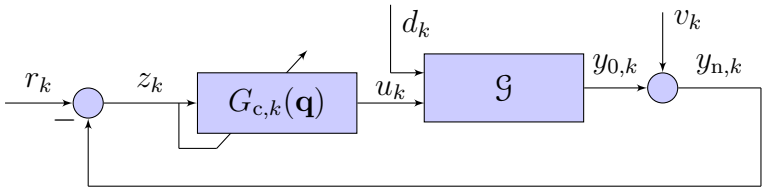

Figure 7.2: Block diagram representation of the adaptive servo problem with the adaptive controller $G_{\mathrm{c},k}$ and IEEE 13-Node Test Feeder $\mathcal{G}$.

In this chapter we consider only three-phase performance nodes for command-following. In particular, for each performance node we define

$$y_{0,k} \triangleq \begin{bmatrix} y_{0,\mathrm{mag},k} \\ y_{0,\mathrm{ang},k} \end{bmatrix}, \quad z_k \triangleq \begin{bmatrix} z_{\mathrm{mag},k} \\ z_{\mathrm{ang},k} \end{bmatrix}, \tag{7.5}$$

where

$$y_{0,\mathrm{mag},k} \triangleq \begin{bmatrix} \mathrm{V}_{\mathrm{mag,A},k} \\ \mathrm{V}_{\mathrm{mag,B},k} \\ \mathrm{V}_{\mathrm{mag,C},k} \end{bmatrix}, \quad y_{0,\mathrm{ang},k} \triangleq \begin{bmatrix} \mathrm{V}_{\mathrm{ang,A},k} \\ \mathrm{V}_{\mathrm{ang,B},k} \\ \mathrm{V}_{\mathrm{ang,C},k} \end{bmatrix}, \tag{7.6}$$

$$z_{\mathrm{mag},k} \triangleq \begin{bmatrix} z_{\mathrm{mag,A},k} \\ z_{\mathrm{mag,B},k} \\ z_{\mathrm{mag,C},k} \end{bmatrix}, \quad z_{\mathrm{ang},k} \triangleq \begin{bmatrix} z_{\mathrm{ang,A},k} \\ z_{\mathrm{ang,B},k} \\ z_{\mathrm{ang,C},k} \end{bmatrix}. \tag{7.7}$$

We set up phasor-based control as a servo problem as shown in Figure 7.2 with the objective

being phase balancing. Consequently, r_k has the same dimensions as $y_{0,k}$, and we define

$$r_k \triangleq \begin{bmatrix} r_{\text{mag}} \\ r_{\text{mag}} \\ r_{\text{mag}} \\ r_{\text{ang}} \\ r_{\text{ang}} - 120^\circ \\ r_{\text{ang}} + 120^\circ \end{bmatrix}, \tag{7.8}$$

which allows us to specify the command for each node as a pair of 2 scalars $r_{\text{mag}},\ r_{\text{ang}}$ instead of 6 scalars. This reference is designed to balance the three phases.

We model actuators as idealized three-phase, four-quadrant operation, DC/AC inverters, connected to infinitely large battery packs, at the actuation nodes. To avoid the case where an actuation node goes offline, we assume that each battery pack has infinite capacity. Each RCAC controller commands a single inverter, which can add or extract active and reactive power at each of the 3 phases. The output u_k of the controller $G_{\text{c},k}(\mathbf{q})$ at step k is defined to be u_k scaled by a fixed scaling. An effective scaling is determined from the simulated closed-loop response. In particular, the power extracted at an actuation node is given by

$$P_{\text{act},k} \triangleq \begin{bmatrix} P_{\text{act},A,k} \\ Q_{\text{act},A,k} \\ P_{\text{act},B,k} \\ Q_{\text{act},B,k} \\ P_{\text{act},C,k} \\ Q_{\text{act},C,k} \end{bmatrix} = 200u_k. \tag{7.9}$$

Since we consider only three-phase performance and actuation nodes, it follows that $m = p = 6$ for all simulations. Additionally, we define

$$v_k \triangleq \begin{bmatrix} v_{\text{mag},k} \\ v_{\text{ang},k} \end{bmatrix}, \tag{7.10}$$

$$v_{\mathrm{mag},k} \triangleq \begin{bmatrix} v_{\mathrm{mag},A,k} \\ v_{\mathrm{mag},B,k} \\ v_{\mathrm{mag},C,k} \end{bmatrix}, \quad v_{\mathrm{ang},k} \triangleq \begin{bmatrix} v_{\mathrm{ang},A,k} \\ v_{\mathrm{ang},B,k} \\ v_{\mathrm{ang},C,k} \end{bmatrix}, \tag{7.11}$$

where each component of $v_{\mathrm{mag},k}$ is Gaussian white noise with standard deviation 0.05 V, and each component of $v_{\mathrm{ang},k}$ is Gaussian white noise with standard deviation 0.0005°.

7.4 Nominal Simulation Tuning

In this section we determine the controller structure, weightings, and target model by running multiple simulations of a *single* scenario. The scenario selected for nominal simulation tuning is command following at node 671 with load disturbances at nodes 611 and 634 and with actuation at node 675.

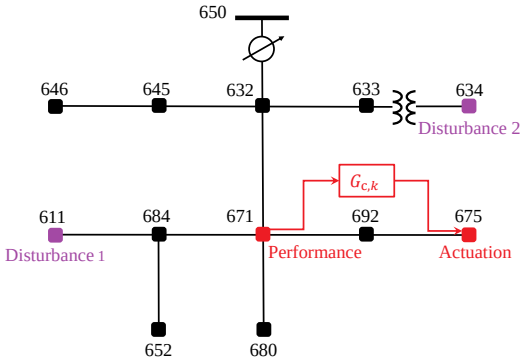

Figure 7.3: Nominal Simulation Tuning: At node 675 the controller $G_{c,k}$ can add or remove active and reactive power; at node 671 the controller $G_{c,k}$ has access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; at nodes 611 and 634, the active and reactive power are varied, which represents an unmodeled load disturbance.

We define the active and reactive power extracted at node 634 as

$$\begin{bmatrix} P_{634,A,k} \\ Q_{634,A,k} \\ P_{634,B,k} \\ Q_{634,B,k} \\ P_{634,C,k} \\ Q_{634,C,k} \end{bmatrix} = \Big(1 + d_k\Big) \begin{bmatrix} 160 \text{ kW} \\ 110 \text{ kvar} \\ 120 \text{ kW} \\ 90 \text{ kvar} \\ 120 \text{ kW} \\ 90 \text{ kvar} \end{bmatrix}, \tag{7.12}$$

the active and reactive power extracted at node 611 as

$$\begin{bmatrix} P_{611,k} \\ Q_{611,k} \end{bmatrix} = \Big(1 + d_k\Big) \begin{bmatrix} 170 \text{ kW} \\ 80 \text{ kvar} \end{bmatrix}, \tag{7.13}$$

the time-varying load disturbance d_k as

$$d_k = \begin{cases} 0, & k < 500, \\ 1, & 500 \leq k < 10000, \\ 1 + \frac{\sin 0.00025k}{4}, & k \geq 10000, \end{cases} \tag{7.14}$$

and set

$$r_{\text{mag}} = 2267 \text{ V}, \tag{7.15}$$

$$r_{\text{ang}} = -5.3^{\circ}. \tag{7.16}$$

To simplify the MIMO controller, each SISO entry is chosen to be either PI or 2nd-order IIR. Based on simulation, the 6×6 controller structure with the least complexity that

follows phase-balancing commands is given by

$$\begin{bmatrix} \text{PI} & \text{IIR} & \text{PI} & \text{PI} & \text{PI} & \text{PI} \\ \text{PI} & \text{PI} & \text{PI} & \text{IIR} & \text{PI} & \text{PI} \\ \text{PI} & \text{PI} & \text{PI} & \text{PI} & \text{PI} & \text{IIR} \\ \text{IIR} & \text{PI} & \text{PI} & \text{PI} & \text{PI} & \text{PI} \\ \text{PI} & \text{PI} & \text{IIR} & \text{PI} & \text{PI} & \text{PI} \\ \text{PI} & \text{PI} & \text{PI} & \text{PI} & \text{IIR} & \text{PI} \end{bmatrix}. \tag{7.17}$$

Fixed-structure RCAC described in [54] is used to adapt the controller coefficients. For (7.17), RCAC adapts 84 coefficients; if each entry of (7.17) were chosen to be 2nd-order IIR, then RCAC would need to adapt 144 coefficients.

Additionally, we set $p_{\mathrm{c},0} = 10^9$, $E_u = 0$, $E_z = \mathrm{diag}(1, 1, 1, 100, 100, 100)$, and use $G_{\mathrm{f}}(\mathbf{q}) = \frac{1}{\mathbf{q}}\hat{H}_1$, where $\hat{H}_1$ is constructed by the following procedure:

i) First, we obtain H_1 by impulsing each input at the actuation node in IEEE13NTF at the initial conditions, and recording the first value of each measurement at the performance node. We separately replace each entry of H_1 with 0, and record the entries whose replacement by 0 does not cause RCAC to fail; these entries are highlighted in green:

$$\begin{bmatrix} 92.54 & 67.93 & 6.71 & -8.39 & -48.21 & -3.84 \\ -89.25 & -3.81 & 9.81 & 20.64 & 14.38 & -28.56 \\ 24.75 & -26.95 & -9.35 & -0.12 & 58.68 & 83.23 \\ 4.03 & -0.50 & -0.18 & -0.15 & 0.07 & 0.87 \\ 0.12 & 0.77 & 0.48 & -0.20 & -0.97 & -0.47 \\ -1.74 & -0.52 & -0.07 & 0.15 & 2.60 & -0.66 \end{bmatrix}.$$

ii) Next, we separately replace each entry of H_1 with its additive inverse, and record the entries whose replacement by the additive inverse does not cause RCAC to fail; these

entries are highlighted in green:

$$\begin{bmatrix} 92.54 & 67.93 & 6.71 & -8.39 & -48.21 & -3.84 \\ -89.25 & -3.81 & 9.81 & 20.64 & 14.38 & -28.56 \\ 24.75 & -26.95 & -9.35 & -0.12 & 58.68 & 83.23 \\ 4.03 & -0.50 & -0.18 & -0.15 & 0.07 & 0.87 \\ 0.12 & 0.77 & 0.48 & -0.20 & -0.97 & -0.47 \\ -1.74 & -0.52 & -0.07 & 0.15 & 2.60 & -0.66 \end{bmatrix}.$$

iii) We now replace all 29 of the green entries in steps 1 and 2 with 0, which yields

$$\begin{bmatrix} 0 & 67.93 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 83.23 \\ 4.03 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.60 & -0.66 \end{bmatrix}.$$

iv) Next, we separately replace each nonzero entry in $\hat{H}_1$ constructed in the previous step with 0 in order to determine the modified matrix $\hat{H}_1$ that has the smallest number of nonzero entries for which RCAC does not fail, which is given by

$$\begin{bmatrix} 0 & 67.93 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 83.23 \\ 4.03 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.60 & 0 \end{bmatrix}. \tag{7.18}$$

v) Finally, the numerical values in (7.18) suggest that it may be possible to parameterize (7.18) using only two numbers. Numerical testing suggests 200 V for magnitude and

5° for angle, which yields

$$\hat{H}_1 = \begin{bmatrix} 0 & 200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 200 \\ 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \end{bmatrix}. \tag{7.19}$$

Note that the rows of (7.19) can be rearranged to construct a 6×6 diagonal matrix. This rearrangement yields a specific pairing of the input and output variables in (7.5)–(7.7). Figures 7.4–7.6 illustrate RCAC using (7.19), the controller structure (7.17), and the chosen weightings, with voltage magnitude and angle commands at node 671, actuation at node 675, and the time-varying load disturbance (7.14) at nodes 611 and 634 in the presence of the sensor noise (7.34). This completes the nominal simulation tuning.

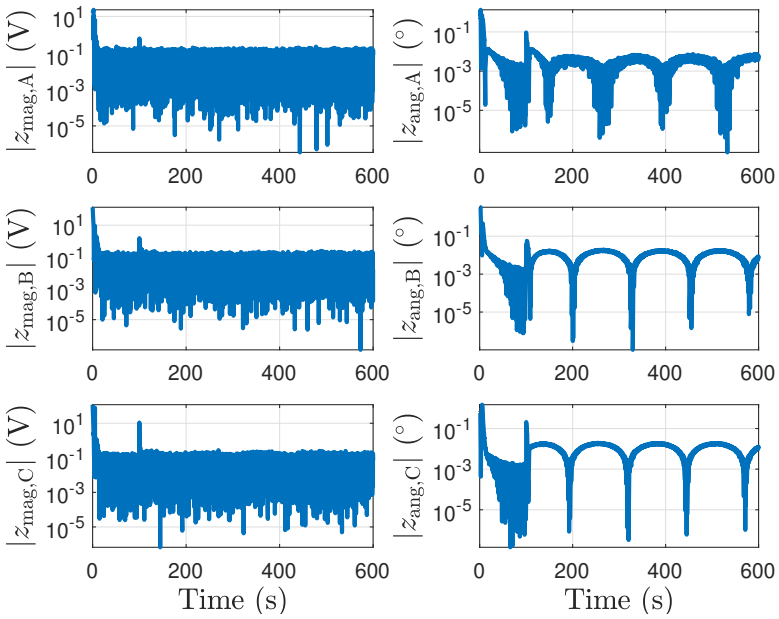

Figure 7.4: Nominal Simulation Tuning: The voltage magnitude and angle errors $|z_{\text{mag}}|$ and $|z_{\text{ang}}|$ at node 671 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 0.1 V and 0.01°, respectively.

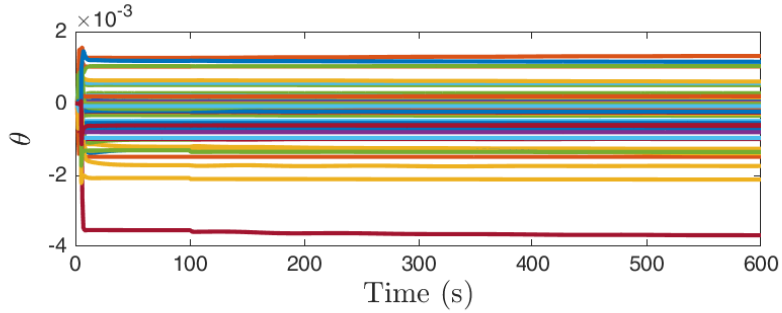

Figure 7.5: Nominal Simulation Tuning: The RCAC controller $G_{\mathrm{c},k}(\mathbf{q})$ adapts to follow the setpoint commands. RCAC readapts at $t = 5$ s to account for the step load disturbance, and again at $t = 100$ s to account for the additional harmonic component of the load disturbance.

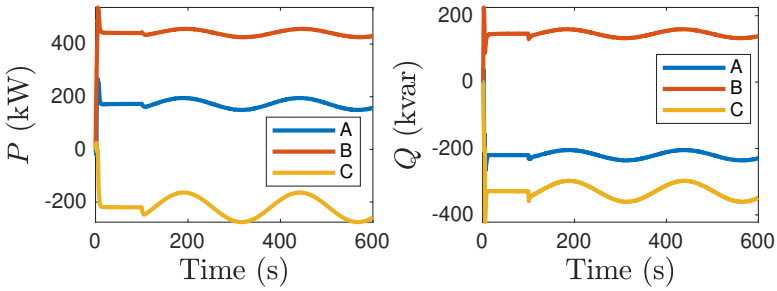

Figure 7.6: Nominal Simulation Tuning: The active and reactive power extracted from each phase by the RCAC controller $G_{\mathrm{c},k}$ at node 675 is shown.

7.5 Perturbed Simulation Testing

In this section we apply the controller structure, weightings, and target model obtained by nominal simulation tuning in the previous section to multiple scenarios involving perturbed simulations. Each simulation represents a perturbation of the nominal simulation in terms of the disturbance signals, actuator saturation levels, and choice of disturbance, actuation, and performance nodes. For all simulations, we use the controller structure given by (7.17) with the weightings $p_{\mathrm{c},0} = 10^9$, $E_u = 0$, $E_z = \mathrm{diag}(1, 1, 1, 100, 100, 100)$, and target model $G_{\mathrm{f}}(\mathbf{q}) = \frac{1}{\mathbf{q}}\hat{H}_1$, where $\hat{H}_1$ is given by (7.19). Each case is simulated once, and no attempt is made to re-tune RCAC based on the response of the closed-loop system.

Example 12. *Command following at node 633 with load disturbance at nodes 611 and 634, actuation at node 632.*

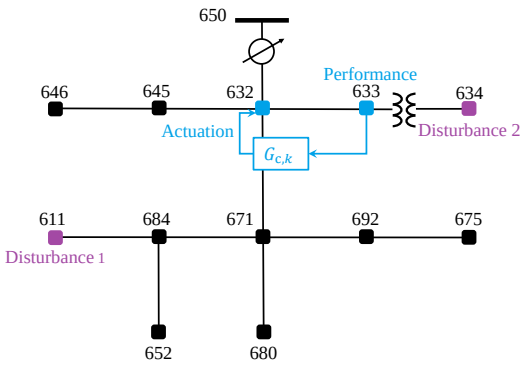

Figure 7.7: Perturbed Simulation Testing 12: At node 632 the controller $G_{\mathrm{c},k}$ can add or remove active and reactive power; at node 633 the controller $G_{\mathrm{c},k}$ has access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; at nodes 611 and 634, the active and reactive power are varied, which represents an unmodeled load disturbance.

We define the active and reactive power at nodes 634 and 611 using (7.12) and (7.13), respectively, and set the load disturbance d_k using (7.14). For node 633 we set the command

$$r_{\mathrm{mag}} = 2312 \text{ V}, \tag{7.20}$$

$$r_{\mathrm{ang}} = -2.56^\circ. \tag{7.21}$$

For the controller $G_{\mathrm{c},k}$, we use the tuning weights described at the beginning of this section. Feedback control with adaptation begins at $t = 2$ s. Figures 7.8–7.10 illustrate RCAC with voltage magnitude and angle commands at node 633, actuation at node 632, and the time-varying load disturbance (7.14) at nodes 611 and 634 in the presence of the sensor noise (7.34). ◇

Example 13. *Command following at nodes 671 and 633 with load disturbances at nodes 611 and 634, and actuation at nodes 675 and 632, using decentralized control.*

We define the active and reactive power at nodes 634 and 611 using (7.12) and (7.13),

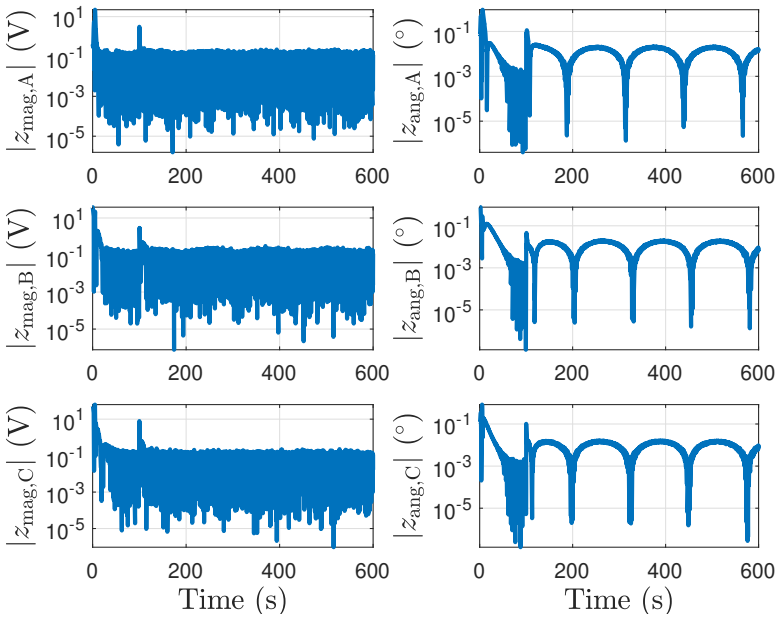

Figure 7.8: Perturbed Simulation Testing 12: The voltage magnitude and angle errors $|z_{\mathrm{mag}}|$ and $|z_{\mathrm{ang}}|$ node 633 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 0.1 V and 0.01°, respectively.

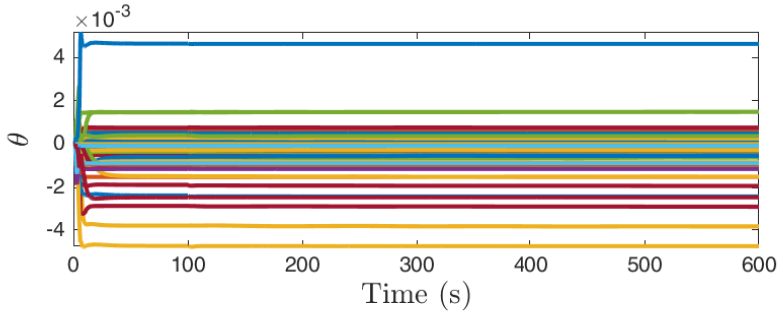

Figure 7.9: Perturbed Simulation Testing 12: RCAC controller $G_{\mathrm{c},k}(\mathbf{q})$ adapts to follow the setpoint commands. RCAC readapts at $t = 5$ s to account for the step load disturbance, and again at $t = 100$ s to account for the additional harmonic component of the load disturbance.

respectively, and set the load disturbance d_k

$$d_k = \begin{cases} 0, & k < 1000, \\ 1, & \text{otherwise.} \end{cases} \tag{7.22}$$

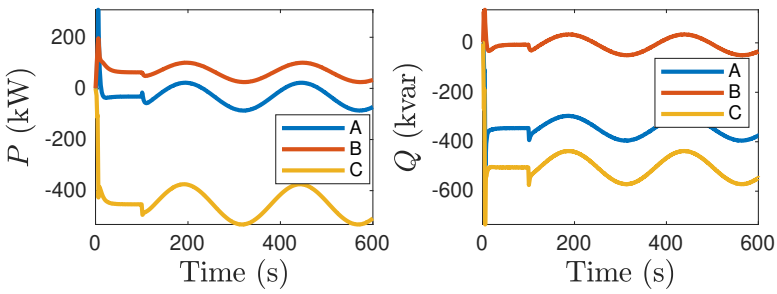

Figure 7.10: Perturbed Simulation Testing 12: The active and reactive power extracted from each phase by the RCAC controller $G_{c,k}$ at node 632 is shown.

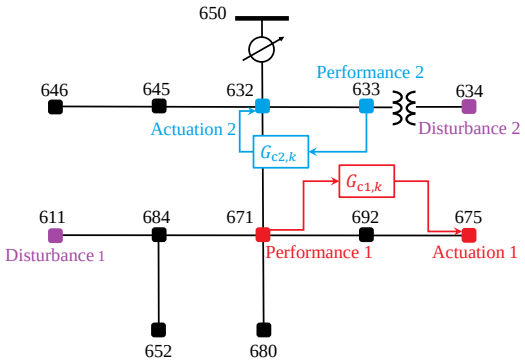

Figure 7.11: Perturbed Simulation Testing 13: At node 675, subcontroller $G_{c1,k}$ can add or remove active and reactive power; at node 671, subcontroller $G_{c1,k}$ has access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; at node 632, subcontroller $G_{c2,k}$ can add or remove active and reactive power; at node 633, subcontroller $G_{c2,k}$ has access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; at nodes 611 and 634, the active and reactive power is varied to represent an unmodeled load disturbance.

For node 671 we set the command

$$r_{671,\text{mag}} = 2300 \text{ V}, \tag{7.23}$$

$$r_{671,\text{ang}} = 1.23^\circ, \tag{7.24}$$

and for node 633 we set the command

$$r_{633,\text{mag}} = 2290 \text{ V}, \tag{7.25}$$

$$r_{633,\text{ang}} = -0.34^\circ. \tag{7.26}$$

For each subcontroller $G_{\text{c1},k}(\mathbf{q})$ and $G_{\text{c2},k}(\mathbf{q})$ we use the tuning weights described at the beginning of this section. Feedback control with adaptation using subcontroller $G_{\text{c1},k}(\mathbf{q})$ and subcontroller $G_{\text{c2},k}(\mathbf{q})$ begins at $t = 2$ s, and $t = 30$ s, respectively. Figures 7.12–7.16 illustrate a pair of decentralized RCAC subcontrollers with voltage magnitude and angle commands at nodes 671 and 633, actuation at nodes 675 and 632, and the time-varying load disturbance (7.22) at nodes 611 and 634 in the presence of the sensor noise (7.34). There is no direct communication between subcontrollers $G_{\text{c1},k}(\mathbf{q})$ and $G_{\text{c2},k}(\mathbf{q})$. ◇

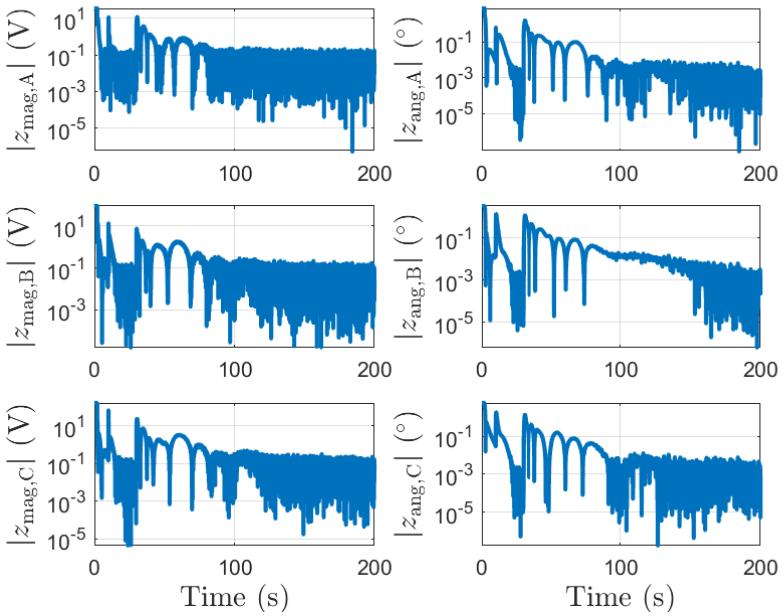

Figure 7.12: Perturbed Simulation Testing 13: Voltage magnitude and angle errors $|z_{\text{mag}}|$ and $|z_{\text{ang}}|$ for node 671 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 0.1 V and 0.01°, respectively.

Example 14. *Command following at node 671 with load disturbances at nodes*

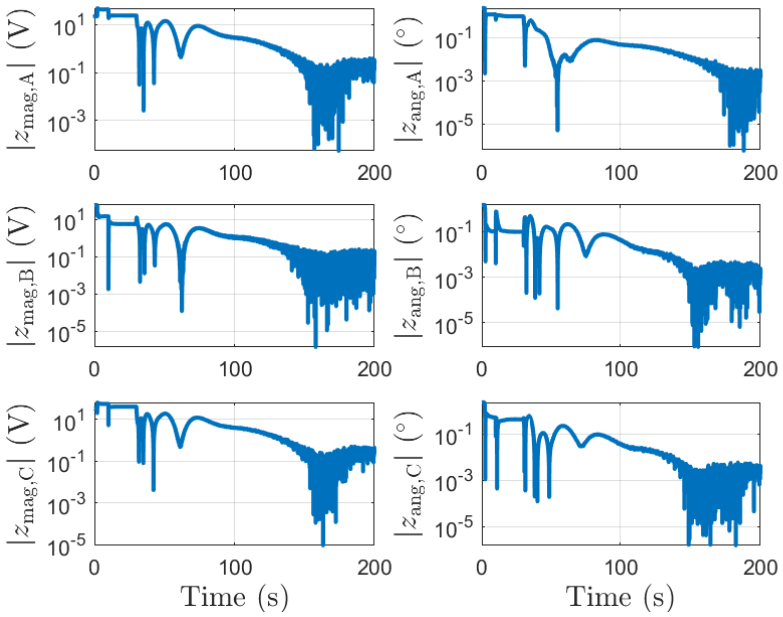

Figure 7.13: Perturbed Simulation Testing 13: Voltage magnitude and angle errors $|z_{\mathrm{mag}}|$ and $|z_{\mathrm{ang}}|$ for node 633 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 0.1 V and 0.01°, respectively.

611 and 634, and actuation at nodes 675 and 632 with unknown multivariable actuator saturation, using decentralized control.

We define the active and reactive power at nodes 634 and 611 using (7.12) and (7.13), respectively, and set the load disturbance d_k as

$$d_k = \begin{cases} 0, & k < 1000, \\ 1, & \text{otherwise.} \end{cases} \tag{7.27}$$

For node 671 we set the command

$$r_{671,\mathrm{mag}} = 2267\ \mathrm{V}, \tag{7.28}$$

$$r_{671,\mathrm{ang}} = -5.3°. \tag{7.29}$$

To simulate an unknown, multivariable actuator saturation at nodes 632 and 675, on

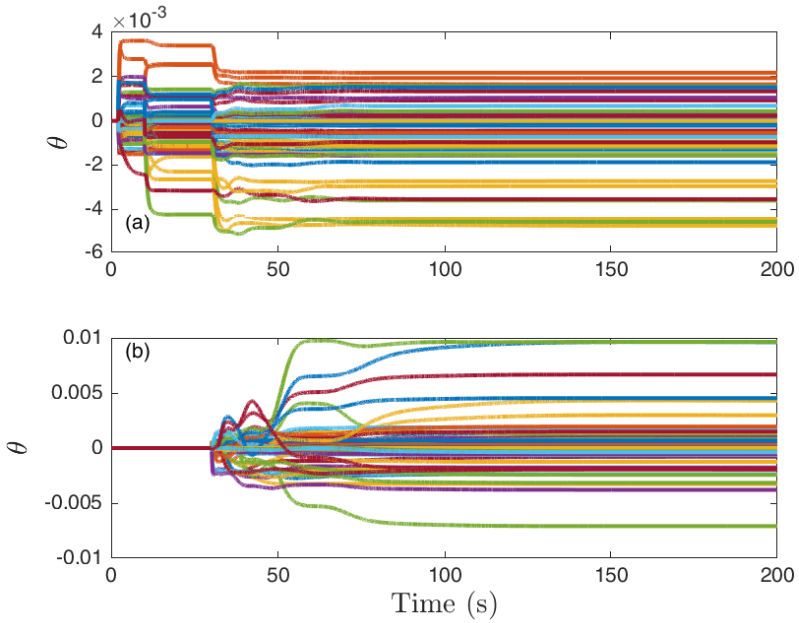

Figure 7.14: Perturbed Simulation Testing 13: (a) θ_k for subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ is zero for $t < 2$ s, after which adaptation of subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ begins in order to follow the setpoint commands. Subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ readapts at $t = 10$ in order to account for the step disturbance, and again at $t = 30$ s in order to account for subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ starting adaptation; (b) θ_k for subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ is zero for $t < 30$ s, after which adaptation of subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ begins to follow the setpoint commands.

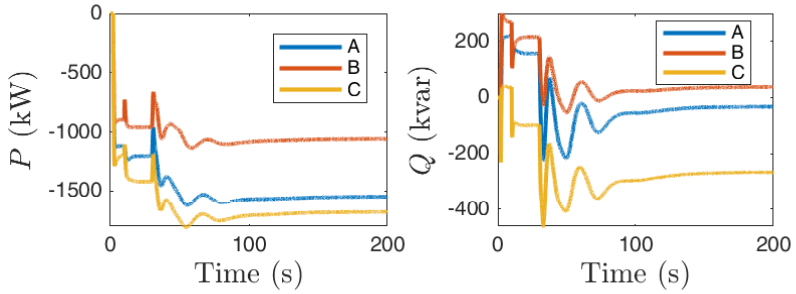

Figure 7.15: Perturbed Simulation Testing 13: The active and reactive power extracted by subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ from each phase at node 675 is shown.

each phase $i \in \{A, B, C\}$, we set

$$P_{\mathrm{act,i}} = \begin{cases} P_{\mathrm{act,i,req}}, & P^2_{\mathrm{act,i,req}} + Q^2_{\mathrm{act,i,req}} \leq S^2_{\mathrm{max}} \\ S_{\mathrm{max}} \cos\alpha, & \text{otherwise,} \end{cases} \tag{7.30}$$

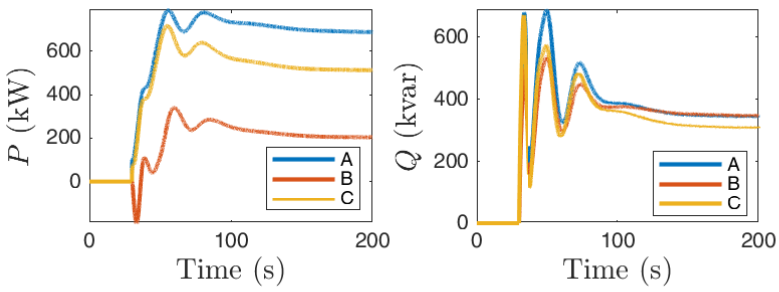

Figure 7.16: Perturbed Simulation Testing 13: The active and reactive power extracted by subcontroller $G_{c2,k}(\mathbf{q})$ from each phase at node 632 is shown.

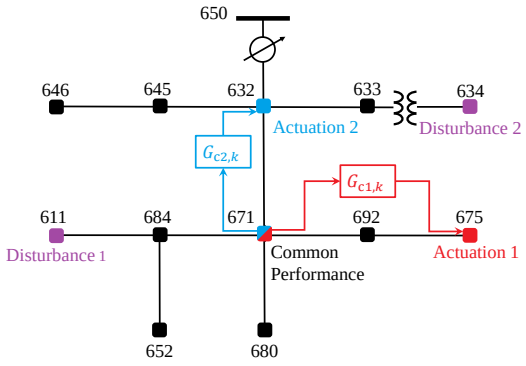

Figure 7.17: Perturbed Simulation Testing 14: At node 675 and node 632, subcontroller $G_{c1,k}(\mathbf{q})$ and subcontroller $G_{c2,k}(\mathbf{q})$ can add or remove active and reactive power, respectively; at node 671, subcontrollers $G_{c1,k}(\mathbf{q})$ and $G_{c2,k}(\mathbf{q})$ have access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; at nodes 611 and 634, the active and reactive power is varied to represent an unmodeled load disturbance.

$$Q_{\text{act,i}} = \begin{cases} Q_{\text{act,i,req}}, & P^2_{\text{act,i,req}} + Q^2_{\text{act,i,req}} \leq S^2_{\max} \\ S_{\max} \sin \alpha, & \text{otherwise,} \end{cases} \tag{7.31}$$

where

$$\alpha \triangleq \text{atan2}(Q_{\text{act,i,req}}, P_{\text{act,i,req}}), \tag{7.32}$$

$P_{\text{act,i}}, Q_{\text{act,i}}$ are the active and reactive power extracted at phase i in kW and kvar respectively, $P_{\text{act,i,req}}, Q_{\text{act,i,req}}$ are the requested active and reactive power extraction at phase i

in kW and kvar respectively, $S_{\max} = 350$ kVA is the maximum apparent power rating for each phase of the inverter, arguments of k are omitted for clarity. This saturation represents an upper limit on the apparent power extracted or injected at each phase: $S_{\max}$.

For each subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ and $G_{\mathrm{c2},k}(\mathbf{q})$ we use the tuning weights described at the beginning of this section. Feedback control with adaptation using subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ and subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ begins at $t = 2$ s, and at $t = 30$ s, respectively. Figures 7.18–7.21 illustrate a pair of decentralized RCAC subcontrollers with voltage magnitude and angle commands at node 671, actuation at nodes 675 and 632, and the time-varying load disturbance (7.27) at nodes 611 and 634 in the presence of the sensor noise (7.34). There is no direct communication between subcontrollers $G_{\mathrm{c1},k}(\mathbf{q})$ and $G_{\mathrm{c2},k}(\mathbf{q})$.

In Figure 7.22, the requested and extracted reactive power is plotted versus the requested and extracted active power for $0 \leq k \leq 60000$, for each phase A, B, C, at each actuation node. This shows that the unknown, multivariable actuator saturation affects five of the six phases of the two actuation nodes. RCAC has no knowledge of either the saturation form or level. ◇

Example 15. *Command following at node 671 with PV generation at nodes 611, 634, 645, 646, 652, and 692, and actuation at nodes 675 and 632, using decentralized control.*

We include PV penetration in IEEE13NTF by the following procedure. From the Pecan Street online repository [55], we obtain generation and consumption data spanning 24-h (midnight to midnight) of 200 customers with PV generation capability. Since this data is per minute, we use linear interpolation to obtain sub-minute data for simulation. Next, for nodes 611, 634, 645, 646, 652, and 692, we aggregate this data using combinations of the 200 customers. Then we replace the static loads in IEEE13NTF at each of these nodes with the aggregated data. The aggregated data represents a collection of loads with a significant proportion of PV generation. The time-varying load is a disturbance to be

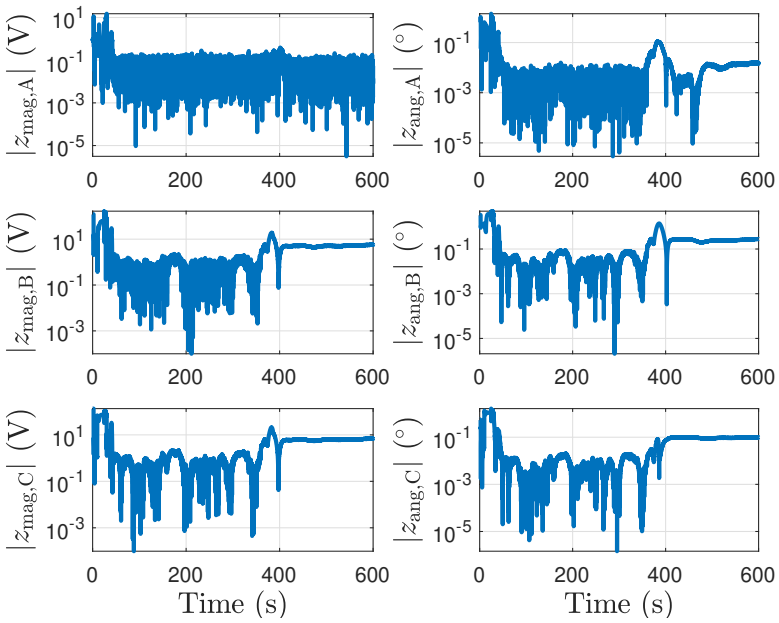

Figure 7.18: Perturbed Simulation Testing 14: Voltage magnitude and angle errors $|z_{\mathrm{mag}}|$ and $|z_{\mathrm{ang}}|$ for node 671 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 1 V and 0.1°, respectively.

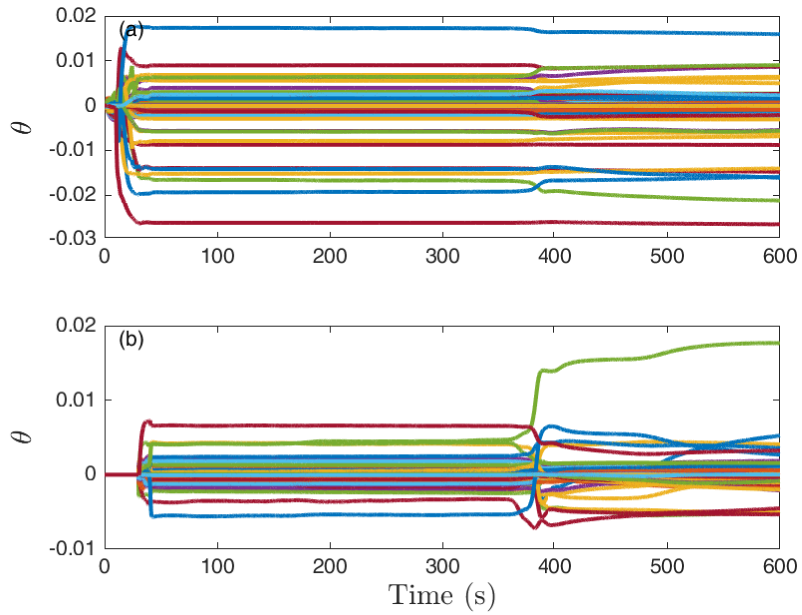

Figure 7.19: Perturbed Simulation Testing 14: (a) at $t = 2$ s, subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ begins adapting. At $t = 30$ s, subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ begins adapting. RCAC has no knowledge of the saturation.

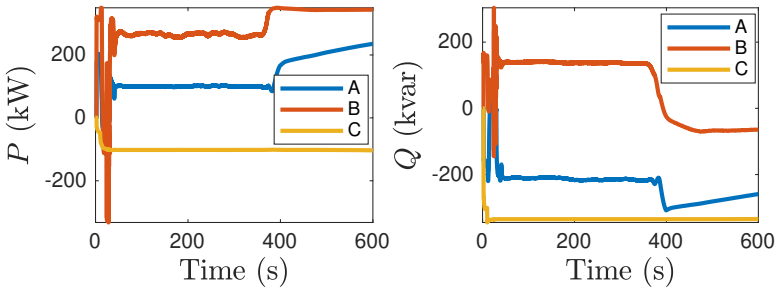

Figure 7.20: Perturbed Simulation Testing 14: The active and reactive power extracted by subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ from each phase at node 675 is shown.

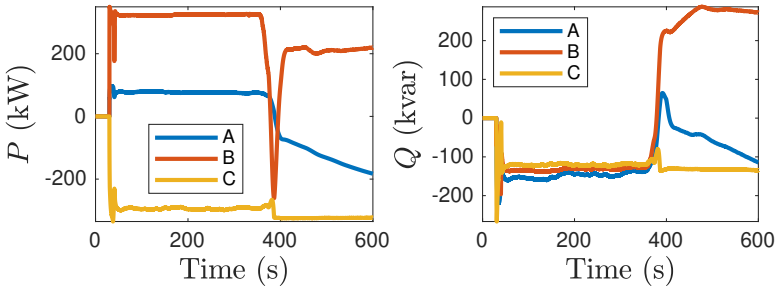

Figure 7.21: Perturbed Simulation Testing 14: The active and reactive power extracted by subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ from each phase at node 632 is shown.

rejected by RCAC. For node 671 we set the command

$$r_{671,\mathrm{mag}} = 2267\ \mathrm{V}, \tag{7.33}$$

$$r_{671,\mathrm{ang}} = -5.3^{\circ}. \tag{7.34}$$

For each subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ and $G_{\mathrm{c2},k}(\mathbf{q})$, we use the tuning weights described at the beginning of this section. There is no communication between subcontrollers $G_{\mathrm{c1},k}(\mathbf{q})$ and $G_{\mathrm{c2},k}(\mathbf{q})$. Feedback control with adaptation using subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ and subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ begins at $t = 2$ s and $t = 30$ s, respectively. Figures 7.24–7.27 illustrate a pair of decentralized RCAC subcontrollers with voltage magnitude and angle commands at node 671, actuation at nodes 675 and 632, and PV generation at nodes 611, 634, 645, 646, 652,

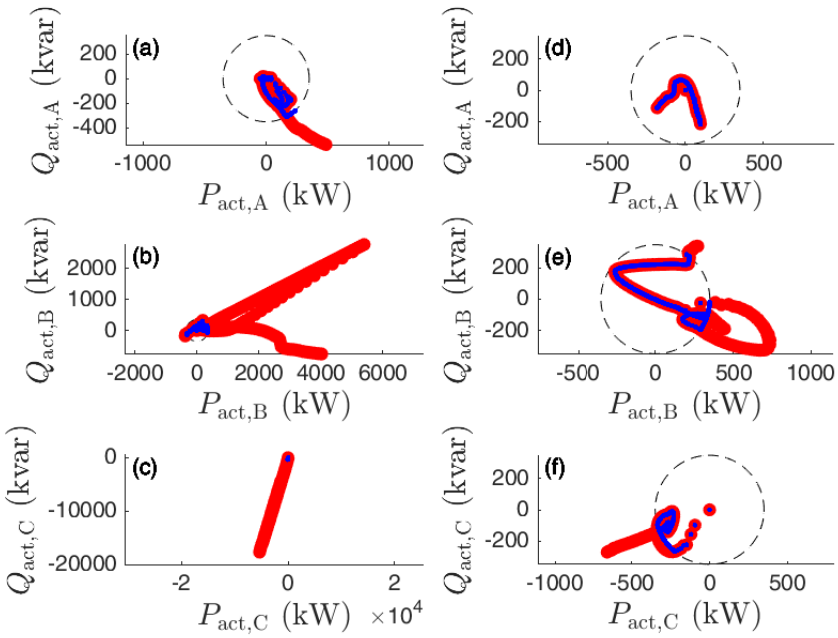

Figure 7.22: Perturbed Simulation Testing 14: The requested (red) and applied (blue) power of phases A, B, C at node 675 is shown in (a), (b), (c), respectively; The requested (red) and applied (blue) power of phases A, B, C at node 632 is shown in (d), (e), (f), respectively. The black, dashed circles represent upper limit of apparent power flow in the actuator $S_{\max}$ defined in (7.30),(7.31).

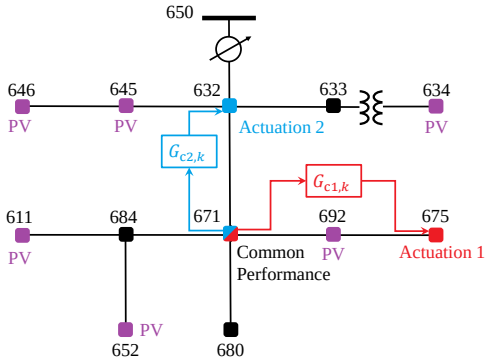

Figure 7.23: Perturbed Simulation Testing 15: At node 675 and node 632, subcontroller $G_{\text{c1},k}(\mathbf{q})$ and subcontroller $G_{\text{c2},k}(\mathbf{q})$ can add or remove active and reactive power, respectively; at node 671, subcontrollers $G_{\text{c1},k}(\mathbf{q})$ and $G_{\text{c2},k}(\mathbf{q})$ have access to the voltage magnitude and angle measurements, where the voltage angles are relative to the slack bus at node 650; nodes 611, 634, 645, 646, 652, and 692 emulate diurnal PV power generation and consumption to represent unmodeled load disturbances.

692 in the presence of the sensor noise (7.34). We define

$$\text{PV penetration} \triangleq \frac{\max\limits_t G(t)}{\max\limits_t L(t)} \times 100\%, \tag{7.35}$$

where $G(t)$ is the total PV generation across all nodes, and $L(t)$ is the total load across all nodes. For this simulation PV penetration $= 70.61\%$. $\diamond$

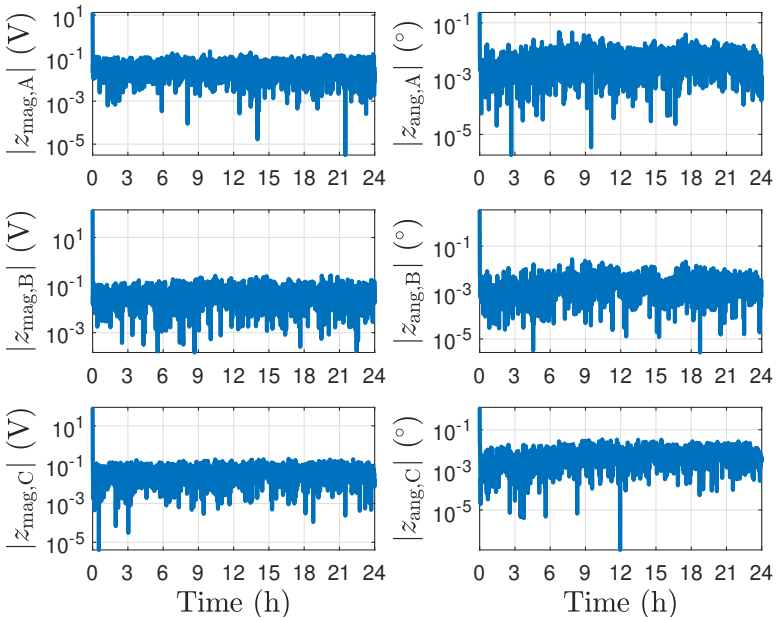

Figure 7.24: Perturbed Simulation Testing 15: Voltage magnitude and angle errors $|z_{\text{mag}}|$ and $|z_{\text{ang}}|$ for node 671 are shown on a logarithmic scale. Asymptotically, the voltage-magnitude and voltage-angle errors are approximately less than 0.1 V and 0.01°, respectively.

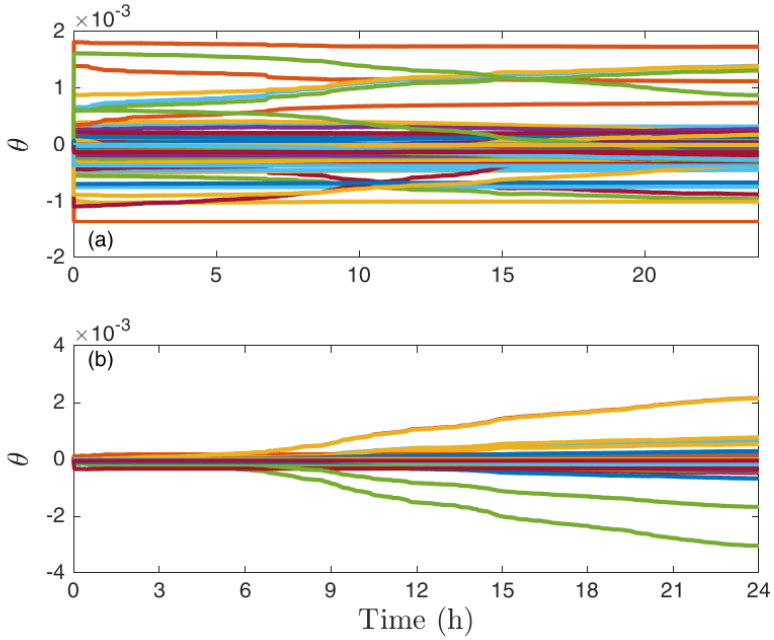

Figure 7.25: Perturbed Simulation Testing 15: (a) θ_k for subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ is shown; (b) θ_k for subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ is shown.

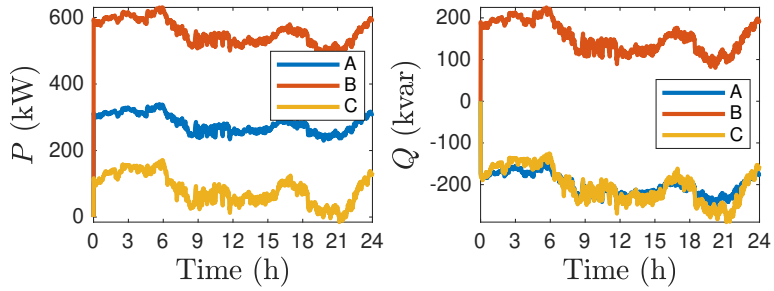

Figure 7.26: Perturbed Simulation Testing 15: The active and reactive power extracted by subcontroller $G_{\mathrm{c1},k}(\mathbf{q})$ from each phase at node 675 is shown.

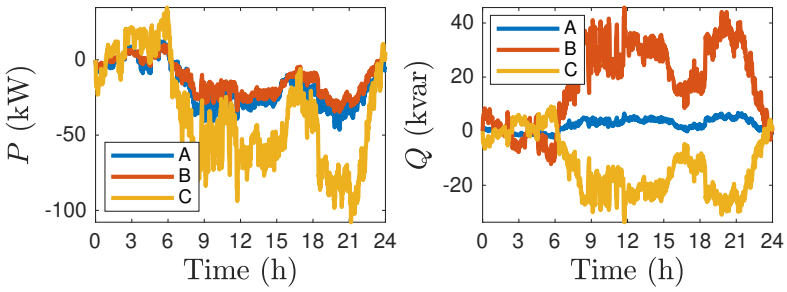

Figure 7.27: Perturbed Simulation Testing 15: The active and reactive power extracted by subcontroller $G_{\mathrm{c2},k}(\mathbf{q})$ from each phase at node 632 is shown.

CHAPTER 8

Online Identification Using Recursive Least Squares

This section investigates the performance of RLS for online, closed-loop identification (RLSID). The goal is to estimate key features of the open-loop transfer function $-EG_{\mathrm{d}}(\mathbf{q})$ from u_k to z_k needed to construct $G_{\mathrm{f}}(\mathbf{q})$, which, as shown in Section 4, serves as the target model for $\widetilde{G}_{z\tilde{u},k}(\mathbf{q})$. Since closed-loop identification may lead to biased estimates, open-loop identification is also considered in order to provide a baseline comparison.

8.1 RLSID

In this subsection, RLSID is used to identify $EG_{\mathrm{d}}(\mathbf{q})$. The transfer function $EG_{\mathrm{d}}(\mathbf{q})$ from u_k to $y_{z,k}$ is given by

$$EG_{\mathrm{d}}(\mathbf{q}) = (I_q\mathbf{q}^n + F_1\mathbf{q}^{n-1} + \cdots + F_n)^{-1}(G_0\mathbf{q}^n + G_1\mathbf{q}^{n-1} + \cdots + G_n), \tag{8.1}$$

where $G_0, \ldots, G_n \in \mathbb{R}^{q\times m}$, and $F_1, \ldots, F_n \in \mathbb{R}^{q\times q}$ are the numerator and denominator coefficients of the transfer function, respectively.

Consider the sampled-data identification architecture shown in Figure 8.1, which is based on Figure 3.2. Since E is known, $y_{z,k} = Ey_k$ can be computed internally by RLSID. Furthermore, at each step k, the requested control input u_k and the measurement y_k are

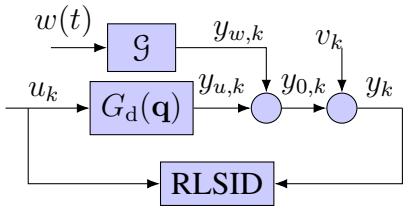

Figure 8.1: Online identification using RLSID.

assumed to be available. In order to identify $EG_{\rm d}(\mathbf{q})$, a model of the form

$$y_{z,k} = -\sum_{i=1}^{\eta} F_{i,k} y_{z,k-i} + \sum_{i=0}^{\eta} G_{i,k} u_{k-i}, \tag{8.2}$$

is fit to data where η is the RLSID window length, and $G_{0,k}, \ldots, G_{\eta,k} \in \mathbb{R}^{q\times m}$, and $F_{1,k}, \ldots, F_{\eta,k} \in \mathbb{R}^{q\times q}$ are numerator and denominator coefficient matrices that are to be estimated.

Next, note that (8.2) can be written as

$$y_{z,k} = \phi_{{\rm m},k}\theta_{{\rm m},k}, \tag{8.3}$$

where

$$\phi_{{\rm m},k} \triangleq \begin{bmatrix} -y_{z,k-1} \\ \vdots \\ -y_{z,k-\eta} \\ u_k \\ \vdots \\ u_{k-\eta} \end{bmatrix}^{\rm T} \otimes I_q \in \mathbb{R}^{q\times l_{\theta_{\rm m}}}, \tag{8.4}$$

$$\theta_{{\rm m},k} \triangleq \text{vec}\,[\, F_{1,k} \;\cdots\; F_{\eta,k}\, G_{0,k} \;\cdots\; G_{\eta,k}\,] \in \mathbb{R}^{l_{\theta_{\rm m}}}, \tag{8.5}$$

is the *model coefficient vector*, and $l_{\theta_{\rm m}} = \eta q^2 + (\eta+1)qm$. The *model-output error* is defined by

$$z_{\mathrm{m},k}(\theta_{\mathrm{m}}) \triangleq y_{z,k} - \phi_{\mathrm{m},k}\theta_{\mathrm{m}}, \tag{8.6}$$

where θ_{m} is an argument for optimization of the form

$$\theta_{\mathrm{m}} \triangleq \mathrm{vec}\,[\, F_1 \ \cdots \ F_\eta \ G_0 \ \cdots \ G_\eta \,] \in \mathbb{R}^{l_{\theta_{\mathrm{m}}}}. \tag{8.7}$$

Next, to apply RLSID, note that the minimizer $\theta_{\mathrm{m},k+1}$ of the quadratic cost function

$$J_k(\theta_{\mathrm{m}}) \triangleq \sum_{i=0}^{k} z_{\mathrm{m},i}(\theta_{\mathrm{m}})^{\mathrm{T}} z_{\mathrm{m},i}(\theta_{\mathrm{m}}) + (\theta_{\mathrm{m}} - \theta_{\mathrm{m},0})^{\mathrm{T}} P_{\mathrm{m},0}^{-1}(\theta_{\mathrm{m}} - \theta_{\mathrm{m},0}) \tag{8.8}$$

is given recursively by

$$P_{\mathrm{m},k+1} = P_{\mathrm{m},k} - P_{\mathrm{m},k}\phi_{\mathrm{m},k}^{\mathrm{T}}(I_q + \phi_{\mathrm{m},k}P_{\mathrm{m},k}\phi_{\mathrm{m},k}^{\mathrm{T}})^{-1}\phi_{\mathrm{m},k}P_{\mathrm{m},k}, \tag{8.9}$$

$$\theta_{\mathrm{m},k+1} = \theta_{\mathrm{m},k} + P_{\mathrm{m},k+1}\phi_{\mathrm{m},k}^{\mathrm{T}}(y_{z,k} - \phi_{\mathrm{m},k}\theta_{\mathrm{m},k}). \tag{8.10}$$

Note that $\theta_{\mathrm{m},0} = 0$ is chosen to reflect the absence of additional modeling information, and $P_{\mathrm{m},0} = p_{\mathrm{m},0}I_{l_{\theta_{\mathrm{m}}}}$, where $p_{\mathrm{m},0} \in (0, \infty)$ is a tuning parameter. As shown by Example 16, the regularization term $(\theta_{\mathrm{m}} - \theta_{\mathrm{m},0})^{\mathrm{T}} P_{\mathrm{m},0}^{-1}(\theta_{\mathrm{m}} - \theta_{\mathrm{m},0})$ in (8.8), which is a required feature of RLS [56–59], causes the estimates to be biased. Although the regularization-induced bias can be minimized by choosing $p_{\mathrm{m},0}$ to be large, it cannot be entirely avoided. The RLSID model at step k is given by

$$EG_{\mathrm{d},k}(\mathbf{q}) \triangleq (I_q\mathbf{q}^\eta + F_{1,k}\mathbf{q}^{\eta-1} + \cdots + F_{\eta,k})^{-1}(G_{0,k}\mathbf{q}^\eta + \cdots + G_{\eta,k}). \tag{8.11}$$

Unless stated otherwise, for all of the examples in this dissertation RLSID is applied with a strictly proper model, which is enforced by removing u_k and $G_{0,k}$ from the definitions (8.4) and (8.5), respectively, and redefining $l_{\theta_{\mathrm{m}}} = \eta q(q + m)$.

8.2 Relative Degree and Leading Numerator Coefficient of SISO Systems

In the case where u_k and $y_{z,k}$ are scalar, the transfer function $EG_{\mathrm{d}}(\mathbf{q})$ from u_k to $y_{z,k}$ can be expressed as

$$EG_{\mathrm{d}}(\mathbf{q}) = \frac{EN_{\mathrm{d}}(\mathbf{q})}{D_{\mathrm{d}}(\mathbf{q})} = \frac{G_0\mathbf{q}^n + \cdots + G_n}{\mathbf{q}^n + F_1\mathbf{q}^{n-1} + \cdots + F_n}, \tag{8.12}$$

where n is the order of $EG_{\mathrm{d}}(\mathbf{q})$, and $G_0, \ldots, G_n \in \mathbb{R}$ and $F_1, \ldots, F_n \in \mathbb{R}$ are numerator and denominator coefficients, respectively. The *leading numerator coefficient* of (8.12) is the leftmost nonzero coefficient of $EN_{\mathrm{d}}(\mathbf{q})$, and the *relative degree* of (8.12) is $\xi \stackrel{\triangle}{=} \deg D_{\mathrm{d}}(\mathbf{q}) - \deg EN_{\mathrm{d}}(\mathbf{q})$. Note that G_ξ is leading numerator coefficient of $EG_{\mathrm{d}}(\mathbf{q})$, and, in the case where $\xi \geq 1$, $G_0 = \cdots = G_{\xi-1} = 0$.

8.3 Numerical Examples

For all of the examples in this section, let $G_u(s)$ be given by Case 1 in Table 3.1, and let $G_{\mathrm{d}}(\mathbf{q})$ denote the ZOH discretization of $G(s)$ with $T_{\mathrm{s}} = 0.03$ s/step, $EG_{\mathrm{d}}(\mathbf{q})$ is a SISO 12th-order transfer function with a NMP zero at 1.4901 rad/step. Furthermore, $G_0 = G_1 = G_2 = 0$ and $G_3 = 0.2972$, and thus the relative degree of $EG_{\mathrm{d}}(\mathbf{q})$ is 3 and G_3 is its leading numerator coefficient. To assess the ability of RLSID to estimate the relative degree and leading numerator coefficient of $EG_{\mathrm{d}}(\mathbf{q})$, $G_{i,k}$ and G_i are compared for $i = 1, 2, 3$. Furthermore, to assess the accuracy of the estimate of the NMP zero of $G_{\mathrm{d}}(\mathbf{q})$, the smallest distance $\mathrm{d}_{\mathrm{z},k}$ between the zeros of the RLSID model and the NMP zero of $EG_{\mathrm{d}}(\mathbf{q})$ is computed at each step. In order to assess the accuracy of open- and closed-loop identification, let $\eta = 12$, which is the order of $EG_{\mathrm{d}}(\mathbf{q})$. Each example in this section

involves 100 trials for $0 \leq t \leq 1000$ s.

Example 16. *Open-loop RLSID with no disturbance, no sensor noise, showing regularization-induced bias.* Let the input u_k of $G_{\mathrm{d}}(\mathbf{q})$ be zero-mean, Gaussian white noise with standard deviation 1, and let $\overline{w}_{k,i} = 0$ and $v_k = 0$. To demonstrate the effect of regularization, RLSID is applied to the input-output data with two choices of $p_{\mathrm{m},0}$, namely, $p_{\mathrm{m},0} = 10^{-3}$ and $p_{\mathrm{m},0} = 10^{4}$. The averaged results from 100 trials are shown in Figure 8.2. As shown in Figure 8.2, the errors in the estimates of the first three numerator coefficients and the NMP zero are larger for trials with a larger regularization. ◇

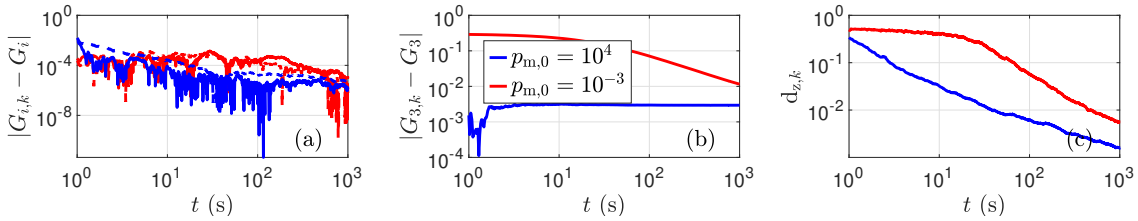

Figure 8.2: Example 16: Regularization in RLSID. Averaged (a) estimation errors for G_1, G_2, (b) estimation error for G_3, (c) $\mathrm{d}_{\mathrm{z},k}$. The accuracy of the identification is poor when the regularization is large.

Example 17. *Open-loop RLSID with disturbance and sensor noise.* Let the input u_k of $G_{\mathrm{d}}(\mathbf{q})$ be zero-mean, Gaussian white noise with standard deviation 1, let and $p_{\mathrm{m},0} = 10^4$. To demonstrate the effect of disturbance and sensor noise, RLSID is applied to the input-output data with $\overline{w}_k = 0, v_k = 0$, and with $\overline{w}_{k,i}, v_k$ being zero-mean, Gaussian white noise with standard deviations $10, 1$, respectively. The averaged results from 100 trials are shown in Figure 8.3. As shown in Figure 8.3, the errors in the estimates of the first three numerator coefficients and the NMP zero are larger for the trials with disturbance and sensor noise present. ◇

Example 18. *Closed-loop RLSID with LQG Control.* To demonstrate the effect of closed-loop control, RLSID is applied to the input-output data for open- and closed-loop scenarios. In particular, for open-loop simulations, u_k is zero-mean, Gaussian white

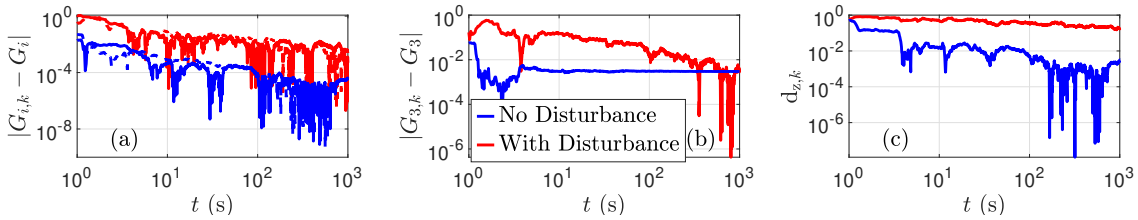

Figure 8.3: Example 17: Disturbance and sensor noise in RLSID. Averaged (a) estimation errors for G_1, G_2, (b) estimation error for G_3, (c) $\mathrm{d}_{z,k}$. Disturbance and sensor noise degrade identification accuracy.

noise with standard deviation 1, and for closed-loop simulations u_k is given by an LQG feedback controller designed using the MATLAB command `lqg` with $Q_{xu} = Q_{wv} = I_{13}$. Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 0.05 and 0.005, respectively. For RLSID set $p_{\mathrm{m},0} = 10^4$. The averaged results from 100 trials are shown in Figure 8.4. As shown in Figure 8.4, the errors in the estimates of the first three numerator coefficients and the NMP zero are larger for closed-loop input-output data relative to open-loop input-output data. ◇

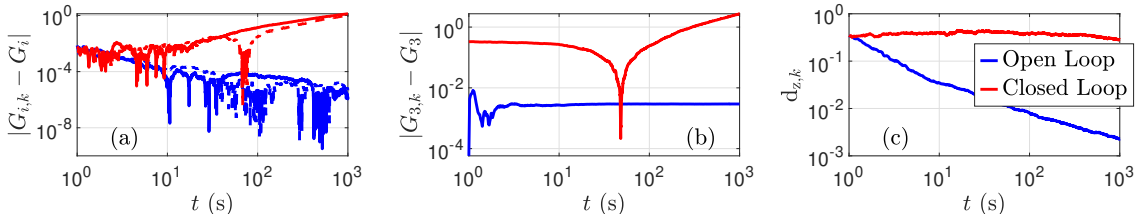

Figure 8.4: Example 18: Closed-loop RLSID. Averaged (a) estimation errors for G_1 and G_2, (b) estimation error for G_3, (c) $\mathrm{d}_{z,k}$. The closed-loop identification accuracy is poor compared to open-loop identification.

CHAPTER 9

Data-Driven Retrospective Cost Adaptive Control

This section describes DDRCAC [33], which combines RLSID with RLS-based adaptive control (RLSAC). The online identification uses RLS to fit an infinite-impulse-response (IIR) model based on data $y_{z,k}$ and u_k collected during closed-loop operation. At each step, the identified IIR model is used to construct a time-dependent target model $G_{\mathrm{f},k}(\mathbf{q})$. In particular, $G_{\mathrm{f},k}(\mathbf{q})$ is constructed as an FIR filter whose numerator is chosen to be the numerator of the latest identified IIR model. Note that this online technique for constructing $G_{\mathrm{f},k}(\mathbf{q})$ is a variation of the offline technique described in Section 4, where $G_{\mathrm{f}}(\mathbf{q})$ was constructed using only the NMP zeros of $EG_{\mathrm{d}}(\mathbf{q})$. This approach avoids the need to compute NMP zeros during online operation and can be used in the MIMO case, where the numerator of the RLSID model is a $q \times m$ polynomial matrix. This target model is then used by RLSAC to update the coefficients of an IIR controller. For DDRCAC, both RLS implementations use variable-rate forgetting (VRF), as given by the following result [17].

Proposition 9.1. *For all $k \geq 0$, let $\bar{y}_k \in \mathbb{R}^{l_{\bar{y}}}$, $\phi_k \in \mathbb{R}^{l_{\bar{y}} \times l_{\bar{\theta}}}$, $\lambda_k \in (0,1]$, and define $\rho_k \triangleq \prod_{j=0}^{k} \lambda_j$. Let $\bar{\theta}_0 \in \mathbb{R}^{l_{\bar{\theta}}}$, and let $\bar{P}_0 \in \mathbb{R}^{l_{\bar{\theta}} \times l_{\bar{\theta}}}$ be positive definite. Furthermore, for all $k \geq 0$, denote the minimizer of*

$$J_k(\bar{\theta}) \triangleq \sum_{i=0}^{k} \frac{\rho_k}{\rho_i} (\bar{y}_i - \phi_i \bar{\theta})^{\mathrm{T}} (\bar{y}_i - \phi_i \bar{\theta}) + \rho_k (\bar{\theta} - \bar{\theta}_0)^{\mathrm{T}} \bar{P}_0^{-1} (\bar{\theta} - \bar{\theta}_0). \tag{9.1}$$

where $\bar{\theta} \in \mathbb{R}^{l_{\bar{\theta}}}$, *by* $\bar{\theta}_{k+1} \triangleq \underset{\bar{\theta} \in \mathbb{R}^{l_{\bar{\theta}}}}{\operatorname{argmin}} J_k(\bar{\theta})$. *Then, for all* $k \geq 0$, $\bar{\theta}_{k+1}$ *is given by*

$$\bar{P}_{k+1} = \tfrac{1}{\lambda_k}\bar{P}_k - \tfrac{1}{\lambda_k}\bar{P}_k\phi_k^{\mathrm{T}}(\lambda_k I_{l_{\bar{y}}} + \phi_k\bar{P}_k\phi_k^{\mathrm{T}})^{-1}\phi_k\bar{P}_k, \tag{9.2}$$

$$\bar{\theta}_{k+1} = \bar{\theta}_k + \bar{P}_{k+1}\phi_k^{\mathrm{T}}(\bar{y}_k - \phi_k\bar{\theta}_k). \tag{9.3}$$

For RLSID and RLSAC, a technique for specifying λ_k is given later in this section.

9.1 RLSID

In order to identify $EG_{\mathrm{d}}(\mathbf{q})$, an IIR model of the form (8.2) is fit to data. Since E is known, $y_{z,k} = Ey_k$ can be computed internally by RLSID. Using Proposition 9.1, for all $k \geq 0$ the model coefficient vector $\theta_{\mathrm{m},k}$ is updated recursively using

$$P_{\mathrm{m},k+1} = \tfrac{1}{\lambda_{\mathrm{m},k}}P_{\mathrm{m},k} - \tfrac{1}{\lambda_{\mathrm{m},k}}P_{\mathrm{m},k}\phi_{\mathrm{m},k}^{\mathrm{T}}(\lambda_{\mathrm{m},k}I_q + \phi_{\mathrm{m},k}P_{\mathrm{m},k}\phi_{\mathrm{m},k}^{\mathrm{T}})^{-1}\phi_{\mathrm{m},k}P_{\mathrm{m},k}, \tag{9.4}$$

$$\theta_{\mathrm{m},k+1} = \theta_{\mathrm{m},k} + P_{\mathrm{m},k+1}\phi_{\mathrm{m},k}^{\mathrm{T}}(y_{z,k} - \phi_{\mathrm{m},k}\theta_{\mathrm{m},k}), \tag{9.5}$$

where $\phi_{\mathrm{m},k}$ and $\theta_{\mathrm{m},k}$ are given by (8.4) and (8.5), respectively, and $P_{\mathrm{m},0} \in \mathbb{R}^{l_{\theta_{\mathrm{m}}} \times l_{\theta_{\mathrm{m}}}}$ is positive definite. The RLSID model at step k is given by

$$EG_{\mathrm{d},k}(\mathbf{q}) = (I_q\mathbf{q}^{\eta} + F_{1,k}\mathbf{q}^{\eta-1} + \cdots + F_{\eta,k})^{-1}(G_{0,k}\mathbf{q}^{\eta} + \cdots + G_{\eta,k}). \tag{9.6}$$

9.2 RLSAC

Define the strictly proper dynamic compensator

$$u_k \triangleq \mathrm{sat}_{\bar{u}}(\phi_{\mathrm{c},k}\theta_{\mathrm{c},k}), \tag{9.7}$$

where $\phi_{\mathrm{c},k}$ and $\theta_{\mathrm{c},k}$ are given by (4.3) and (4.4), respectively. The definition (9.7) represents an IIR controller whose output is saturated component-wise by the scalar saturation function $\mathrm{sat}_{\bar{u}}$ defined by

$$\mathrm{sat}_{\bar{u}_i}(x_i) \triangleq \begin{cases} x_i, & |x_i| < \bar{u}_i, \\ \mathrm{sign}(x_i)\bar{u}_i, & |x_i| \geq \bar{u}_i. \end{cases} \tag{9.8}$$

Next, define the filtered signals

$$u_{\mathrm{f},k} \triangleq G_{\mathrm{f},k}(\mathbf{q})u_k, \tag{9.9}$$

$$\phi_{\mathrm{f},k} \triangleq G_{\mathrm{f},k}(\mathbf{q})\phi_{\mathrm{c},k}, \tag{9.10}$$

where, for startup, $u_{\mathrm{f},k}$ and $\phi_{\mathrm{f},k}$ are initialized at zero and thus are computed as the forced responses of (9.9) and (9.10), respectively, and where $G_{\mathrm{f},k}(\mathbf{q})$ is the time-dependent target model constructed using the updated numerator coefficients $G_{0,k+1}, \ldots, G_{\eta,k+1}$ of the model (8.2). In particular,

$$G_{\mathrm{f},k}(\mathbf{q}) \triangleq -\sum_{i=0}^{\eta} G_{i,k+1}\frac{1}{\mathbf{q}^i}, \tag{9.11}$$

which has the same form as (5.51) except that (9.11) is time varying, generalizes to MIMO systems, and includes all of the zeros of $EG_{\mathrm{d},k}(\mathbf{q})$. In the case where $q = m = 1$, it follows from $G_{0,k} = \cdots = G_{\xi-1,k} = 0$ and $G_{\xi,k} = G_\xi$ that (9.11) and $-EG_{\mathrm{d}}(\mathbf{q})$ have the same leading numerator coefficient and relative degree. Note that, at each step k, the numerator of (9.11) is chosen to be the numerator of (9.6). If there exists $k \geq 0$ such that $G_{0,k} = \cdots = G_{\eta,k} = 0_{q\times m}$, then $G_{\mathrm{f},k}(\mathbf{q})$ is chosen to be

$$G_{\mathrm{f},k}(\mathbf{q}) \triangleq -\mathbf{1}_{q\times m}. \tag{9.12}$$

The retrospective performance variable is defined to be

$$\hat{z}_k(\theta_{\mathrm{c}}) \triangleq z_k - u_{\mathrm{f},k} + \phi_{\mathrm{f},k}\theta_{\mathrm{c}}. \tag{9.13}$$

Using (9.11) and (9.12), (9.13) can be expressed as

$$\hat{z}_k(\theta_{\mathrm{c}}) \triangleq z_k - N_k\bar{u}_k + N_k\bar{\phi}_{\mathrm{c},k}\theta_{\mathrm{c}}. \tag{9.14}$$

where

$$N_k \triangleq \begin{cases} [\,-\mathbf{1}_{q\times m}\;0\;\cdots\;0\,], & G_{0,k+1} = \cdots = G_{\eta,k} = 0, \\ [\,-G_{0,k+1}\;\cdots\;-G_{\eta,k+1}\,], & \text{otherwise}, \end{cases} \tag{9.15}$$

$N_k \in \mathbb{R}^{q\times(\eta+1)m}$, $\bar{u}_k$ and $\bar{\phi}_{\mathrm{c},k}$ are given by (4.16) and (4.17) with $n_{\mathrm{f}} = \eta$, respectively, and $G_{0,k+1}, \ldots, G_{\eta,k+1} \in \mathbb{R}^{q\times m}$ are the numerator coefficients of the RLSID model. Note that, by performing the RLSID update at step k before the RLSAC update, it follows thus the estimated numerator coefficients $G_{0,k+1}, \ldots, G_{\eta,k+1}$ are available for constructing N_k at step k.

Next, define the *controller cost variable*

$$z_{\mathrm{c},k}(\theta_{\mathrm{c}}) \triangleq \begin{bmatrix} E_z\hat{z}_k(\theta_{\mathrm{c}}) \\ E_u\phi_{\mathrm{c},k}\theta_{\mathrm{c}} \\ E_{\Delta u}(\phi_{\mathrm{c},k}\theta_{\mathrm{c}} - u_k) \end{bmatrix} \in \mathbb{R}^{q+r_1+r_2}, \tag{9.16}$$

where the performance weighting $E_z \in \mathbb{R}^{q\times q}$ is nonsingular and $E_u \in \mathbb{R}^{r_1\times m}$, $E_{\Delta u} \in \mathbb{R}^{r_2\times m}$ are the control weighting and control-move weighting, respectively. If $E_u = 0$ and $E_{\Delta u} = 0$, then $r_1 = 0$ and $r_2 = 0$, respectively, and all expressions involving E_u and $E_{\Delta u}$

are omitted from (9.16), as well as from all subsequent expressions. Note that

$$z_{\mathrm{c},k}(\theta_{\mathrm{c}})^{\mathrm{T}} z_{\mathrm{c},k}(\theta_{\mathrm{c}}) = \hat{z}_k(\theta_{\mathrm{c}})^{\mathrm{T}} R_z \hat{z}_k(\theta_{\mathrm{c}}) + \theta_{\mathrm{c}}^{\mathrm{T}} \phi_{\mathrm{c},k}^{\mathrm{T}} R_u \phi_{\mathrm{c},k} \theta_{\mathrm{c}} + (\phi_{\mathrm{c},k}\theta_{\mathrm{c}} - u_k)^{\mathrm{T}} \phi_{\mathrm{c},k}^{\mathrm{T}} R_{\Delta u} \phi_{\mathrm{c},k} (\phi_{\mathrm{c},k}\theta_{\mathrm{c}} - u_k), \tag{9.17}$$

where $R_z \triangleq E_z^{\mathrm{T}} E_z \in \mathbb{R}^{q \times q}$ is positive definite, and $R_u \triangleq E_u^{\mathrm{T}} E_u \in \mathbb{R}^{m \times m}$, $R_{\Delta u} \triangleq E_{\Delta u}^{\mathrm{T}} E_{\Delta u} \in \mathbb{R}^{m \times m}$ are positive semidefinite.

Using Proposition 9.1, for all $k \geq 0$ the controller coefficient vector $\theta_{\mathrm{c},k}$ is updated recursively using

$$P_{\mathrm{c},k+1} = \tfrac{1}{\lambda_{\mathrm{c},k}} P_{\mathrm{c},k} - \tfrac{1}{\lambda_{\mathrm{c},k}} P_{\mathrm{c},k} \phi_{\mathrm{fc},k}^{\mathrm{T}} (\lambda_{\mathrm{c},k} I_{q+r_1+r_2} + \phi_{\mathrm{fc},k} P_{\mathrm{c},k} \phi_{\mathrm{fc},k}^{\mathrm{T}})^{-1} \phi_{\mathrm{fc},k} P_{\mathrm{c},k}, \tag{9.18}$$

$$\theta_{\mathrm{c},k+1} = \theta_{\mathrm{c},k} + P_{\mathrm{c},k+1} \phi_{\mathrm{fc},k}^{\mathrm{T}} (y_{\mathrm{c},k} - \phi_{\mathrm{fc},k} \theta_{\mathrm{c},k}), \tag{9.19}$$

where

$$y_{\mathrm{c},k} \triangleq \begin{bmatrix} E_z z_k - E_z N_k \bar{u}_k \\ 0 \\ -E_{\Delta u} u_k \end{bmatrix} \in \mathbb{R}^{q+r_1+r_2}, \quad \phi_{\mathrm{fc},k} \triangleq \begin{bmatrix} -E_z N_k \bar{\phi}_{\mathrm{c},k} \\ -E_u \phi_{\mathrm{c},k} \\ -E_{\Delta u} \phi_{\mathrm{c},k} \end{bmatrix} \in \mathbb{R}^{(q+r_1+r_2) \times l_{\theta_{\mathrm{c}}}}. \tag{9.20}$$

and $P_{\mathrm{c},0} \in \mathbb{R}^{l_{\theta_{\mathrm{c}}} \times l_{\theta_{\mathrm{c}}}}$ is positive definite.

For all of the examples in this dissertation, $\theta_{\mathrm{m},k}$ and $\theta_{\mathrm{c},k}$ are initialized as 0, and thus (9.12) is invoked at startup. This assumption reflects the absence of additional prior modeling information; however, $\theta_{\mathrm{m},k}$ and $\theta_{\mathrm{c},k}$ can be initialized based on any available modeling information. To initialize RLSAC and RLSID, $P_{\mathrm{c},0} = p_{\mathrm{c},0} I_{l_{\theta_{\mathrm{c}}}}$ and $P_{\mathrm{m},0} = p_{\mathrm{c},0} I_{l_{\theta_{\mathrm{m}}}}$ are chosen, where, for convenience, $p_{\mathrm{c},0} > 0$ is a common tuning parameter.

9.3 Data-Dependent Variable Rate Forgetting

For data-dependent variable-rate forgetting, set

$$\lambda_{\mathrm{m},k} = \frac{1}{1 + \varepsilon e(z_{\mathrm{m},k-\tau_\mathrm{d}}, \ldots, z_{\mathrm{m},k})\mathbf{1}[e(z_{\mathrm{m},k-\tau_\mathrm{d}}, \ldots, z_{\mathrm{m},k})]}, \tag{9.21}$$

$$\lambda_{\mathrm{c},k} = \frac{1}{1 + \varepsilon e(z_{k-\tau_\mathrm{d}}, \ldots, z_k)\mathbf{1}[e(z_{k-\tau_\mathrm{d}}, \ldots, z_k)]}, \tag{9.22}$$

where

$$e(x_{k-\tau_\mathrm{d}}, \ldots, x_k) \triangleq \frac{\sqrt{\frac{1}{\tau_\mathrm{n}} \sum_{i=k-\tau_\mathrm{n}}^{k} x_i^{\mathrm{T}} x_i}}{\sqrt{\frac{1}{\tau_\mathrm{d}} \sum_{i=k-\tau_\mathrm{d}}^{k} x_i^{\mathrm{T}} x_i}} - 1.2, \tag{9.23}$$

"$\mathbf{1}$" is the step function that is 0 for negative arguments and 1 for nonnegative arguments, and $e(0, \ldots, 0) \triangleq 0$. In (9.21)–(9.23), $\varepsilon \geq 0$, $0 < \tau_\mathrm{n} < \tau_\mathrm{d}$ are numerator and denominator window lengths, respectively. If the sequence $x_{k-\tau_\mathrm{d}}, \ldots, x_k$ is zero-mean noise, then the numerator and denominator of (9.23) approximate the average standard deviation of the noise over the intervals $[k - \tau_\mathrm{n}, k]$ and $[k - \tau_\mathrm{d}, k]$, respectively. In particular, by choosing $\tau_\mathrm{d} >> \tau_\mathrm{n}$, it follows that the denominator of (9.23) approximates the long-term-average standard deviation of x_k, whereas the numerator of (9.23) approximates the short-term-average standard deviation of x_k. Consequently, the case $e(x_{k-\tau_\mathrm{d}}, \ldots, x_k) > 0$ implies that the short-term-average standard deviation of x_k is greater than the long-term-average standard deviation of x_k plus a threshold of 0.2. The function $e(x_{k-\tau_\mathrm{d}}, \ldots, x_k)$ used in VRF suspends forgetting when the short-term-average standard deviation of x_k drops below 1.2 times the long-term-average standard deviation of x_k. This technique thus prevents forgetting in RLSID and RCAC due to zero-mean sensor noise with constant standard deviation rather than due to the magnitude of the noise-free identification error and command-following error.

A list of parameters to be selected for DDRCAC is presented in Table 9.1.

Table 9.1: Tuning parameters that need to be selected for DDRCAC.

Parameter	Description	Selection
η	Model window length	Integer ≥ 1 (1–10)
n_{c}	Controller window length	Integer ≥ 1 (2–40)
E_u	Control weighting	scaled $m \times m$ identity
$E_{\Delta u}$	Control move weighting	scaled $m \times m$ identity
$\bar{u}$	Control saturation-limit vector	95% actuator saturation limit
$p_{\mathrm{c},0}$	Initial RLS covariance scaling for RLSAC and RLSID	$p_{\mathrm{c},0} > 0$
ε	Forgetting parameter	$0 \leq \varepsilon < 1$ (0.001 – 0.2)
$\tau_{\mathrm{n}}, \tau_{\mathrm{d}}$	Forgetting window lengths	Integers $\tau_{\mathrm{d}} > \tau_{\mathrm{n}}$ ($\tau_{\mathrm{n}} \in$ [1–400], $\tau_{\mathrm{d}} \sim 3\tau_{\mathrm{n}}$)

9.4 Numerical Examples

This subsection demonstrates DDRCAC, which uses no prior knowledge of $EG_{\mathrm{d}}(\mathbf{q})$ and thus, in particular, no prior knowledge of the leading numerator coefficient, NMP zeros, or relative degree of $EG_{\mathrm{d}}(\mathbf{q})$. Unless stated otherwise, all of the examples in this subsection use the same tuning parameters, namely, $p_{\mathrm{c},0} = 10^3$, $\eta = 4$, $n_{\mathrm{c}} = 20$, $E = 1$, $E_z = 1$, $E_u = 0.1$, $E_{\Delta u} = 0$, $\varepsilon = 0.001$, $\tau_{\mathrm{n}} = 200$, $\tau_{\mathrm{d}} = 600$, and $\bar{u} = 1$. Furthermore, for all of the examples in this section $\tilde{y}_k \triangleq z_k$. As in Section 8.3, the ability of RLSID to estimate the leading numerator coefficient and relative degree of $EG_{\mathrm{d}}(\mathbf{q})$ is investigated by comparing the first ξ numerator coefficients of the RLSID model and $EG_{\mathrm{d}}(\mathbf{q})$. For all of the examples in this subsection RLSID and RLSAC are applied with a strictly proper RLSID model and target model, respectively, which is enforced by removing u_k and $G_{0,k}$ from the definitions (8.4) and (8.5), respectively, redefining $l_{\theta_{\mathrm{m}}} = \eta q(q+m)$ and

$$N_k \triangleq \begin{cases} [\, -\mathbf{1}_{q\times m}\ 0\ \cdots\ 0\,], & G_{0,k+1} = \cdots = G_{\eta,k} = 0, \\ [\, -G_{1,k+1}\ \cdots\ -G_{\eta,k+1}\,], & \text{otherwise}, \end{cases} \tag{9.24}$$

where $N_k \in \mathbb{R}^{q\times \eta m}$.

Example 19. *Interaction between RLSID and RLSAC.* Let

$$G_u(s) = \frac{100(s-10)(s+30)}{(s+10)(s^2-10s+1000)}, \tag{9.25}$$

which is unstable and NMP, and, for $T_\mathrm{s} = 0.01$ s/step, let $G_\mathrm{d}(\mathbf{q})$ denote the ZOH discretization of $G_u(s)$. Then the NMP zero, leading numerator coefficient, and relative degree of $G_\mathrm{d}(\mathbf{q})$ are 1.1056 rad/step, $G_\xi = G_1 = 1.079$, and $\xi = 1$, respectively. Let $\overline{w}_{k,i} = 0$, and let v_k be zero-mean, Gaussian white noise with standard deviation 0.001.

For command following with $r_k = \sin 0.23 T_\mathrm{s} k$, control is applied using an LQG controller designed for $(A_\mathrm{d}, B_\mathrm{d}, C_\mathrm{d}, D_\mathrm{d})$ augmented with a model of the harmonic command, using the MATLAB command lqg, with weights $Q_{xu} = Q_{wv} = I_6$. Figures 9.1(a) and 9.1(c) show the response and control u_k for the LQG controller, respectively. RLSID with VRF given by (9.4), (9.5) is used for closed-loop identification with the time-invariant LQG controller, as shown in Figures 9.1(e) and 9.1(h). In this case the leading numerator coefficient and NMP zero of $G_\mathrm{d}(\mathbf{q})$ are estimated poorly, as shown by Figures 9.1(j),(l).

Next, adaptive control is applied with $\eta = 10$, where Figures 9.1(k),(m) show that, at $t \approx 0.1$ s, the leading numerator coefficient is correctly estimated but the estimate of the NMP zero of $G_\mathrm{d}(\mathbf{q})$ is erroneous. The initially poor RLSID model at $t \approx 0.1$ s results in a poor, infeasible target model, which induces a large transient response in $y_{z,k}$ and u_k for $0 \leq t \leq 1$ s. The additional persistency of this transient response, however, facilitates subsequent identification of the NMP zero of $G_\mathrm{d}(\mathbf{q})$ at $t \approx 0.85$ s, as shown in Figure 9.1(k). Note that $\theta_{\mathrm{m},k}$ is converged for $t > 0.41$ s, and thus the time-dependent target model is also converged. With the converged time-dependent target model, Figure 9.1(g) shows that RLS with VRF facilitates further adaptation of $\theta_{\mathrm{c},k}$ for $t > 0.41$ s, and $\theta_{\mathrm{c},k}$ is converged for $t > 1$ s. This example thus illustrates mutually beneficial interaction between RLSID and RLSAC. $\diamond$

Example 20. *RCAC, DDRCAC, and $\hat{z}_k(\theta_{\mathrm{c},k+1})$ decomposition.* Let $G_u(s)$ be given by Case 2 in Table 3.1 with $T_\mathrm{s} = 0.01$ s/step. In order to avoid numerical issues arising from the need for multiple discretized systems, the disturbance w_k is assumed to be constant within each sampling interval $[kT_\mathrm{s}, (k+1)T_\mathrm{s})$. Because $G_u(s)$ is lightly damped, high-precision arithmetic is used to compare the left- and right-hand sides of (5.7).

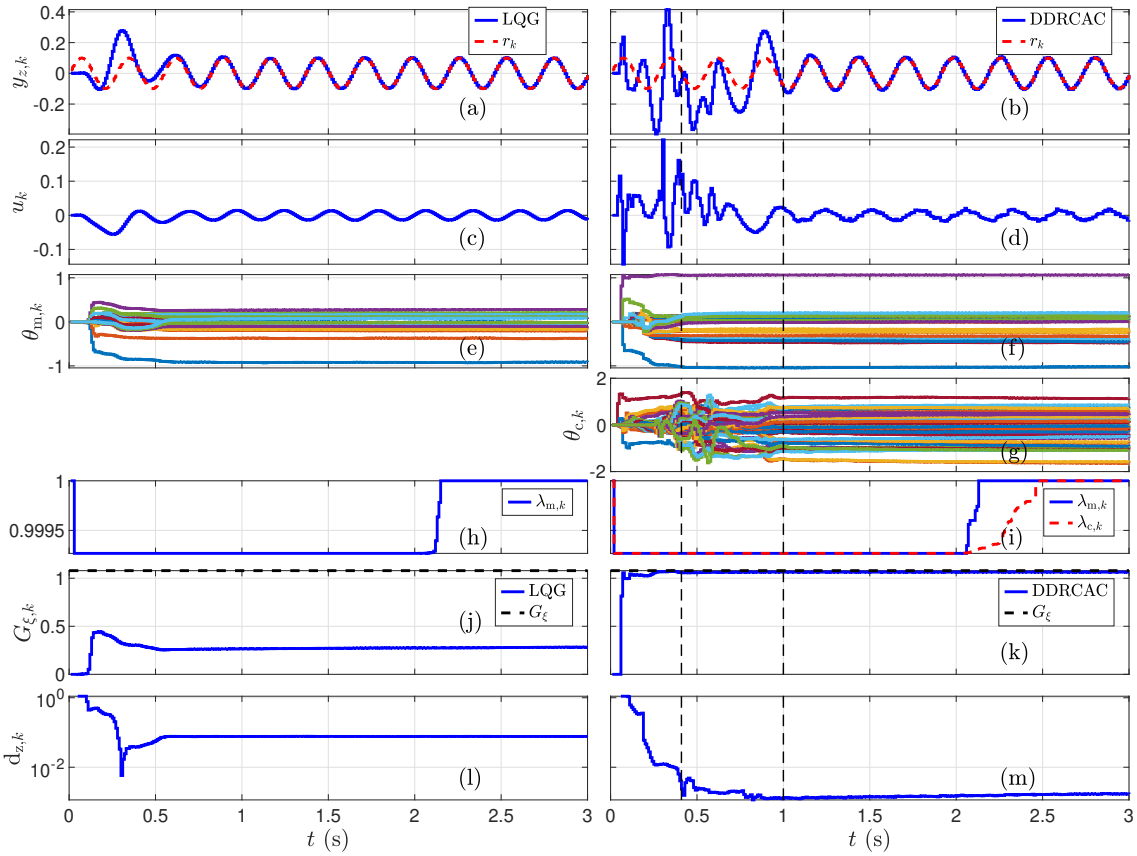

Figure 9.1: Example 19: RLSID with LQG yields biased estimates of G_ξ and the NMP zero of $G_{\mathrm{d}}(\mathbf{q})$; for adaptive control, the biases in (k) and (m) are smaller. The vertical dashed lines denote the settling times of $\theta_{\mathrm{m},k}$ and $\theta_{\mathrm{c},k}$.

For disturbance rejection, let $r_k = 0$, and let w_k and v_k be zero-mean, Gaussian white noise with standard deviations 0.1 and 0.001, respectively. Three scenarios are considered, namely, (1) RCAC with the nominal target model $G_{\mathrm{f}}(\mathbf{q}) = -0.153\frac{(\mathbf{q}-1.1078)}{\mathbf{q}^2}$, which assumes knowledge of the true leading numerator coefficient, NMP zeros, and relative degree of $EG_{\mathrm{d}}(\mathbf{q})$ (2) RCAC with the off-nominal target model $G_{\mathrm{f}}(\mathbf{q}) = -0.35\frac{(\mathbf{q}-1.2)}{\mathbf{q}^2}$, where the leading numerator coefficient is erroneous by a factor of 2.29 and the NMP zero is erroneous by a factor of 1.08, and (3) DDRCAC. RCAC is applied with $n_{\mathrm{c}} = 20$, $E_u = 0.1$, $E_z = 1$, and $p_{\mathrm{c},0} = 10^3$, which are identical to the tuning parameters for DDRCAC specified above.

The first, second, and third columns of Figure 9.2 correspond to scenarios (1), (2), and (3), respectively. Note that the closed-loop performance degrades significantly due to the

use of the off-nominal target model. However, with no prior knowledge of the system dynamics, DDRCAC achieves closed-loop performance similar to RCAC with the nominal target model.

Figure 9.3 shows the RLSID coefficients $\theta_{\mathrm{m},k}$, the true and estimated leading numerator coefficients G_ξ and $G_{\xi,k}$, respectively, the variable-rate forgetting factors $\lambda_{\mathrm{m},k}, \lambda_{\mathrm{c},k}$, and the closest distance $\mathrm{d}_{\mathrm{z},k}$ between the zeros of the RLSID model and the NMP zero of $EG_{\mathrm{d}}(\mathbf{q})$. Note that RLSID approximates the leading numerator coefficient, NMP zero, and relative degree of $EG_{\mathrm{d}}(\mathbf{q})$, and thus the time-dependent target model (9.11) approximates the nominal target model. ◇

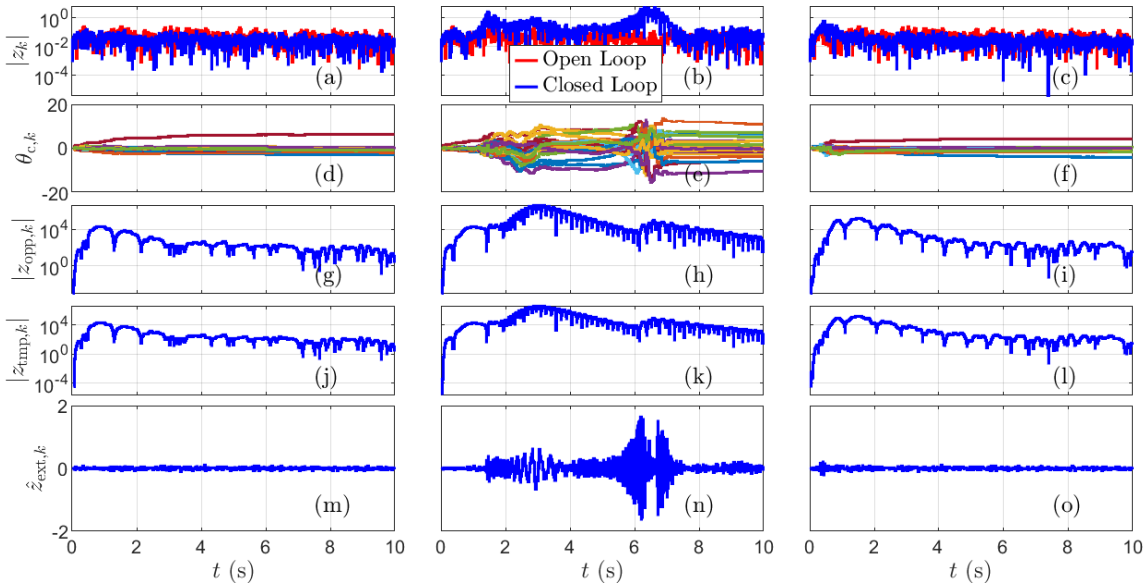

Figure 9.2: Example 20: Columns 1–3 correspond to RCAC with the nominal target model, RCAC with an off-nominal target model, and DDRCAC. The performance of **DDRCAC** is similar to the performance RCAC in column 1.

Example 21. *Effect of sensor noise and* $p_{\mathrm{c},0}$. Let $G_u(s)$ be given by Case 3 in Table 3.1 with $T_{\mathrm{s}} = 0.01$ s/step. Then the NMP zeros, leading numerator coefficient, and relative degree of $G_{\mathrm{d}}(\mathbf{q})$ are $\{1.106 \pm 0.106\jmath\}$ rad/step, $G_\xi = 0.128$, and $\xi = 3$, respectively. Hence, $G_1 = 0$, $G_2 = 0$, and $G_{\xi,k} = G_3 = 0.128$. The time-dependent target model (9.11) has the same leading numerator coefficient and relative degree as $-EG_{\mathrm{d}}(\mathbf{q})$, and is thus equal to the nominal target model, if $G_{0,k} = \cdots = G_{\xi-1,k} = 0$ and $G_{\xi,k} = G_\xi$.

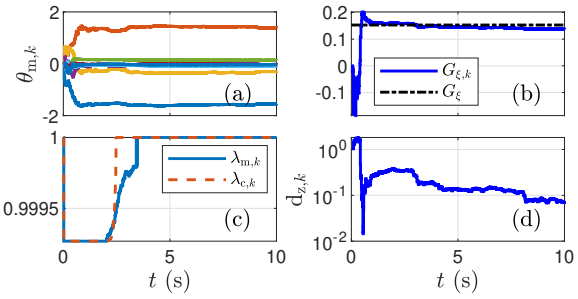

Figure 9.3: Example 20: (a) RLSID coefficients $\theta_{\mathrm{m},k}$; (b) identified and true leading numerator coefficients, $G_{\xi,k}$, and G_ξ, respectively; (c) forgetting factors $\lambda_{\mathrm{m},k}$ and $\lambda_{\mathrm{c},k}$ for RLSID and RLSAC, respectively; (d) $\mathrm{d}_{\mathrm{z},k}$.

Let $r_k = 0$, let $\overline{w}_{k,i}$ be Gaussian white noise with standard deviation 0.1 and mean 0.5, and consider three scenarios, where v_k is zero-mean, Gaussian white noise with standard deviations 0.001, 0.01, and 0.1; these scenarios correspond to the first, second, and third columns of Figure 9.4, respectively. The measurement signal-to-noise ratio (SNR) is defined to be the ratio of the root-mean-square of the last 1000 subinterval steps of y_k to the root-mean-square of the last 1000 subinterval steps of v_k. Note that the suppression metric g_{s} decreases as SNR increases.

Next, to investigate the effect of $p_{\mathrm{c},0}$, three disturbance rejection scenarios with $r_k = 0$ are considered, where $p_{\mathrm{c},0}$ is 10, 10^2, and 10^3; these scenarios correspond to the first, second, and third columns of Figure 9.5, respectively. Note that, although the transient response of identified numerator coefficients increases with $p_{\mathrm{c},0}$, the level of asymptotic disturbance suppression is largely insensitive to the choice of $p_{\mathrm{c},0}$. ◇

Example 22. *Example 10 revisited using DDRCAC.* As shown in Example 10, the control of non-square MIMO systems using RCAC can cause the creation of NMP cascade zeros of $(G_{\mathrm{d}}, G_{\mathrm{c},k})$ that are cancelled by poles of $G_{\mathrm{c},k}$, leading to the divergence of u_k. DDRCAC is applied with $E_u = 0$, and thus the tuning parameters are identical to the RCAC tuning parameters in Example 10. As in Example 10, Figure 9.6 shows that the controller gives rise to NMP cascade zeros. However, unlike Example 10, these NMP zeros are not cancelled by the controller, and thus u_k does not diverge. ◇

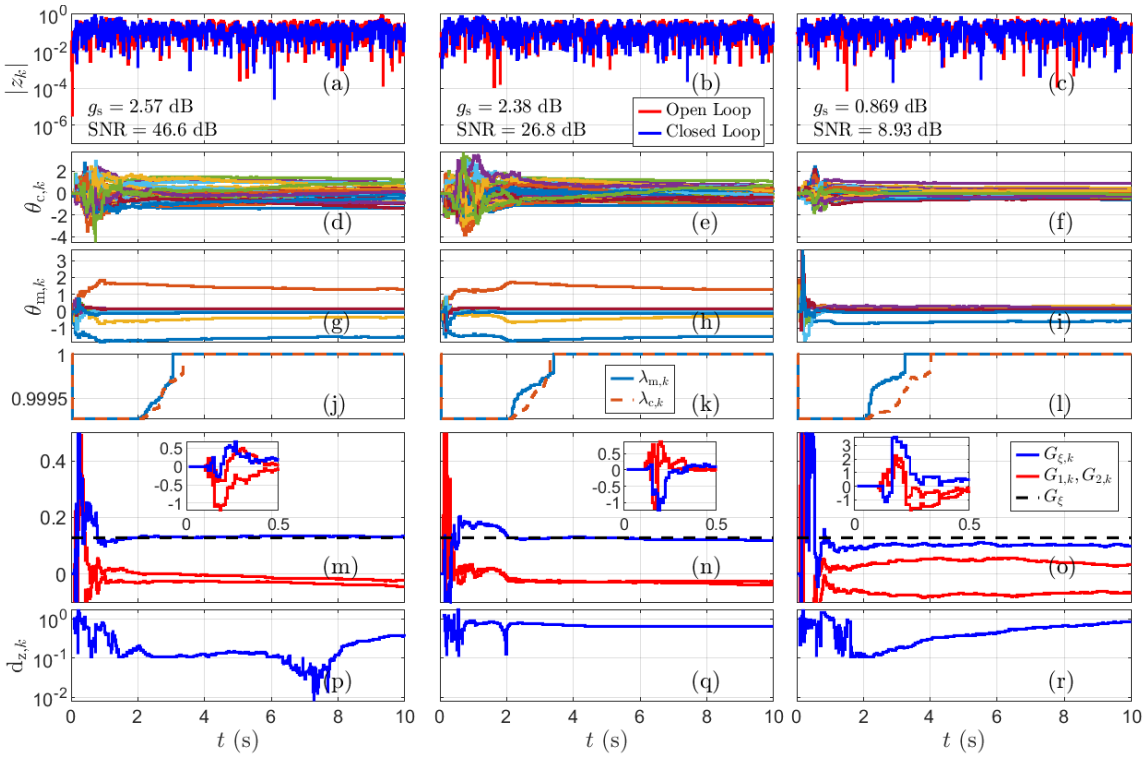

Figure 9.4: Example 21: Columns 1–3 correspond to v_k with standard deviations 0.001, 0.01, and 0.1. The insets in (m), (n), (o) show the full range of the transient response.

Example 23. *Example 11 revisited using DDRCAC.* As shown in Example 11, the control of non-square MIMO systems using RCAC can cause the creation of NMP cascade zeros of $(G_{c,k}, G_d)$ that are cancelled by poles of $G_{c,k}$, leading to the divergence of u_k. DDRCAC is applied with $E_z = I_3, p_{c,0} = 10^5, E_u = 0$, and thus the tuning parameters are identical to the RCAC tuning parameters in Example 11. As in Example 11, Figure 9.7 shows that the controller gives rise to NMP cascade zeros. However, unlike Example 11, these NMP zeros are not cancelled by the controller, and thus u_k does not diverge. ◇

Example 24. *Time-varying relative degree and NMP zeros with abrupt and smooth transitions.* Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 0.1 and 0.01, respectively, and $r_k = 0$. Let $G_1(s)$, $G_2(s)$, and $G_3(s)$ be given by Case 1, Case 2, and Case 3 in Table 3.1, respectively, with minimal realizations (A_1, B_1, C_1, D_1), (A_2, B_2, C_2, D_2), and (A_3, B_3, C_3, D_3), respectively. Furthermore, at each intersample time

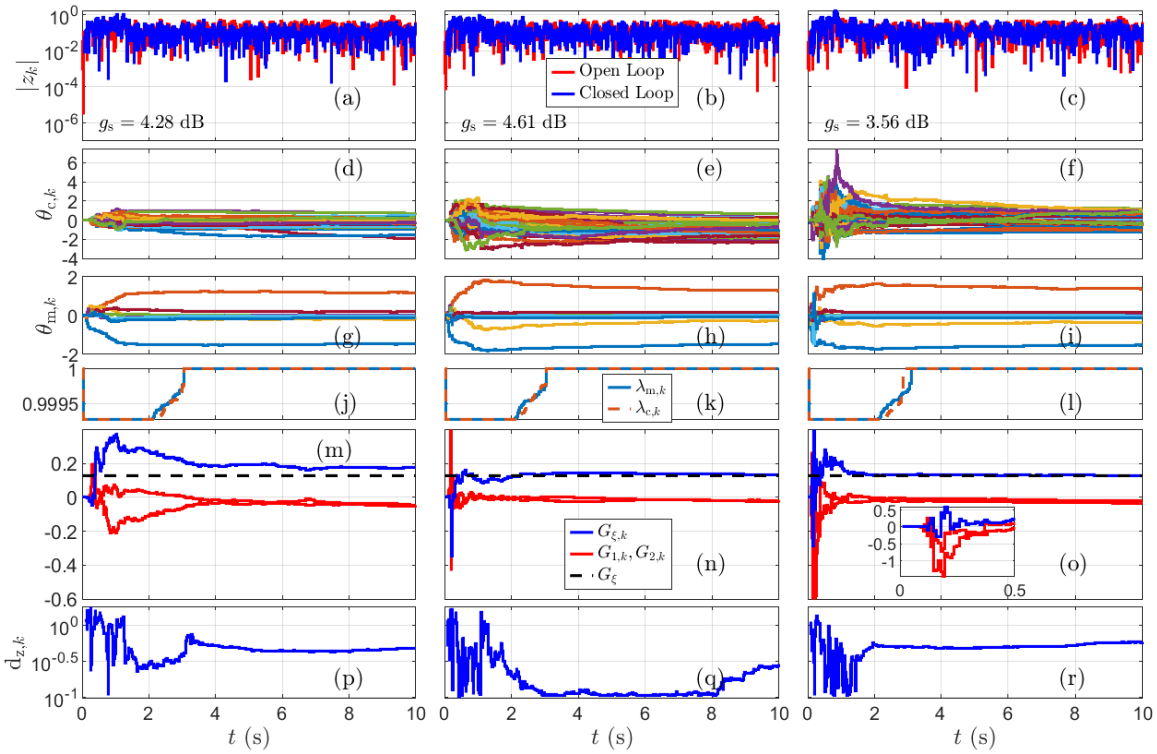

Figure 9.5: Example 21: Columns 1–3 correspond to $p_{c,0} = 10$, $p_{c,0} = 10^2$, $p_{c,0} = 10^3$. The inset in (o) shows the full range of the transient response.

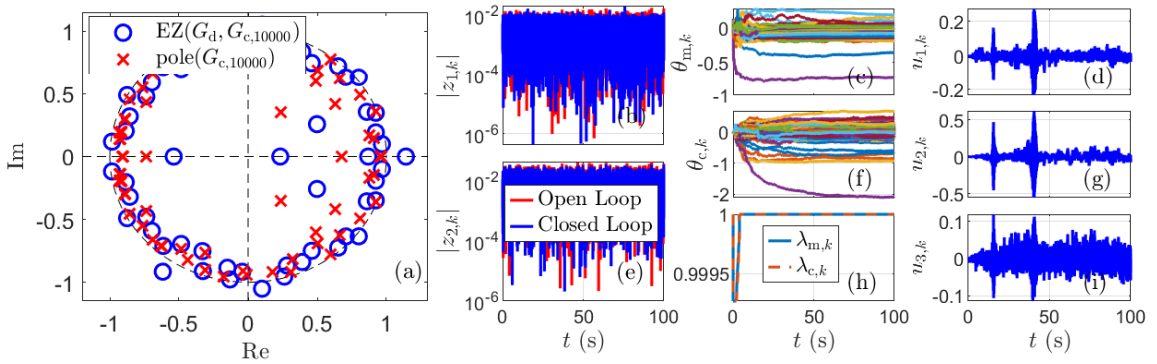

Figure 9.6: Example 22: Example 10 revisited using DDRCAC. Unlike Example 10, no NMP cascade zeros are cancelled by the controller.

step $t = \frac{k}{10}T_s$, let $G_u(s)$ be given by (3.1) and (3.2) with

$$A(t) \triangleq f(A_2, A_1, A_3, t), \quad B_w(t) = B(t) \triangleq f(B_2, B_1, B_3, t), \tag{9.26}$$

$$C(t) \triangleq f(C_2, C_1, C_3, t), \quad D(t) \triangleq f(D_2, D_1, D_3, t), \tag{9.27}$$

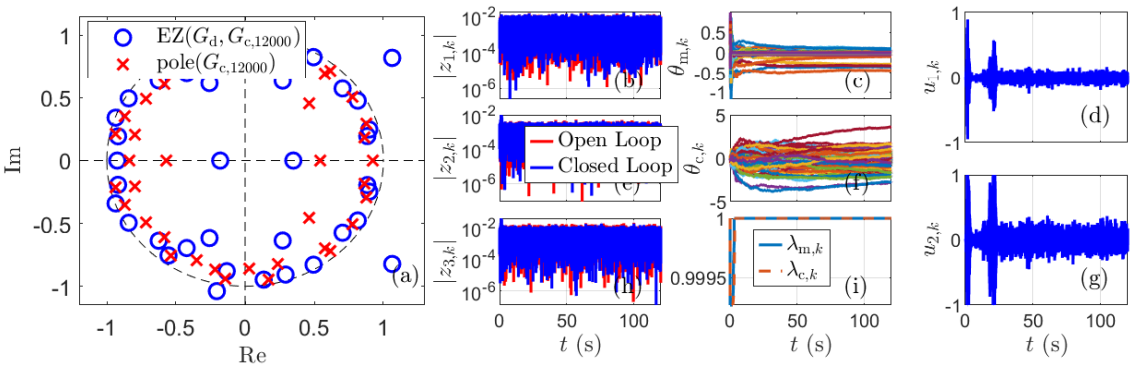

Figure 9.7: Example 23: Example 11 revisited using DDRCAC. Unlike Example 11, no NMP cascade zeros are cancelled by the controller.

$$f(M_1, M_2, M_3, t) \triangleq \begin{cases} M_1, & t \leq 10 \text{ s}, \\ M_2, & 10 < t \leq 15 \text{ s} \\ M_2 + (M_2 - M_1)\frac{t-10}{5}, & 15 < t \leq 20 \text{ s} \\ M_3, & t > 20 \text{ s}. \end{cases} \tag{9.28}$$

Note that, at $t = 10$ s the relative degree of the discretization of (9.26) and (9.27) changes from 1 to 3, and during $15 \leq t < 20$ s, the dynamics of of the discretization of (9.26) and (9.27) smoothly transition from a single real NMP zero at 1.1078 rad/step to a pair of complex NMP zeros at $\{1.106 \pm 0.106\jmath\}$ rad/step.

Figure 9.8 shows that the adaptive controller rejects the disturbance despite the unknown, abrupt and smooth transitions in the dynamics (9.26) and (9.27). Note that Figure 9.8(f), $G_{\xi,k}$ is equal to $G_{1,k}$ for $t \leq 10$ s and equal to $G_{3,k}$ for $t > 10$ s. Furthermore, note that $G_{\xi-1,k}, G_{\xi-2,k}$ are undefined for $t \leq 10$ s, and are thus plotted for $t > 10$ s in Figure 9.8(f). $\diamond$

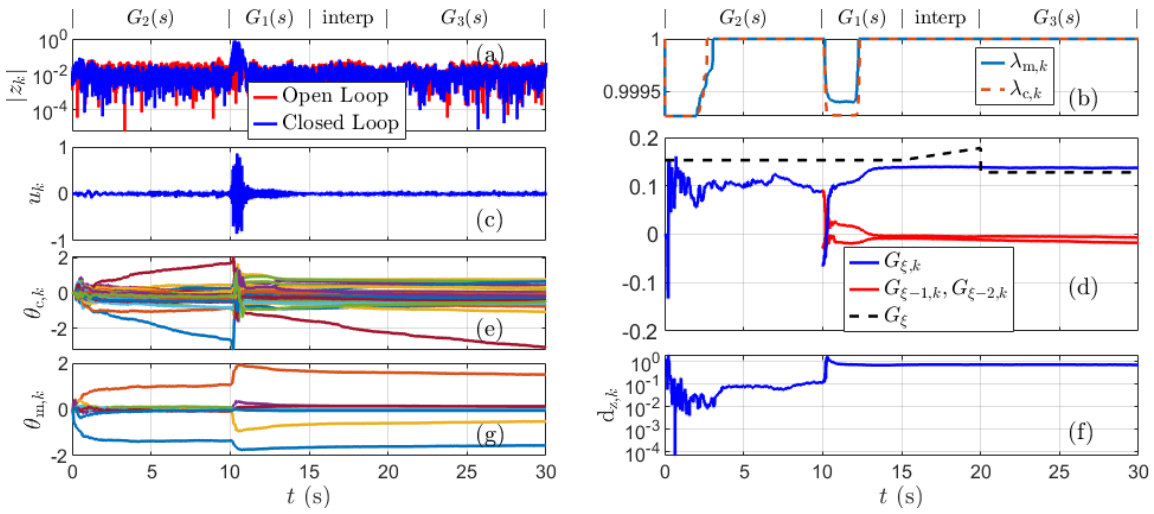

Figure 9.8: Example 24: Disturbance rejection for (9.26) and (9.27). The relative degree changes from 1 to 3 at $t = 10$ s, and, during $t \in [15, 20]$ s, the discretization of (9.26) and (9.27) transitions from one real NMP zero to two complex NMP zeros.

CHAPTER 10

Adaptive Flight Control

In this chapter, DDRCAC is applied to several flight-control problems, namely, (1) roll control of a hypersonic aircraft with an unknown transition from MP to NMP dynamics, (2) pitch-rate control of a flexible aircraft, (3) flutter suppression, and (4) normal-acceleration control a nonlinear planar missile. For consistency in applying DDRCAC, an exactly proper model structure used for RLSID for all of the examples in this section. Furthermore, the signal-to-noise ratio (SNR) between y_k and v_k is computed for all of the subinterval steps of each example. Note that the first three examples are linear, whereas the last example is nonlinear.

Example 25. *Roll control of a hypersonic aircraft with an unknown transition from MP to NMP dynamics.* Consider the linearized lateral dynamics of a hypersonic aircraft

[60–62], given by (3.1), (3.2) with

$$A(t) \triangleq \begin{bmatrix} -0.0771 & 0.269 & -0.9631 & 0.0397 \\ \ell(t,-25.6,-108.8) & 0.0218 & 0.0995 & 0 \\ \ell(t,0.6160,0.4107) & 0.0376 & -0.2687 & 0 \\ 0 & 1 & -0.4202 & 0.0058 \end{bmatrix}, \tag{10.1}$$

$$B(t) = B_w(t) \triangleq \begin{bmatrix} -0.0002 \\ 2.519 \\ \ell(t,-0.0222,-0.0665) \\ 0 \end{bmatrix}, \tag{10.2}$$

$$C \triangleq \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = 0, \quad \ell(t,a,b) \triangleq \begin{cases} a, & t < 80 \text{ s}, \\ a + \frac{t-80}{20}(b-a), & 80 \leq t \leq 100 \text{ s}, \\ b, & t > 100 \text{ s}, \end{cases} \tag{10.3}$$

where the components of $x(t) \triangleq [\ \beta(t)\ \bar{p}(t)\ \bar{r}(t)\ \phi(t)\ ]^{\mathrm{T}}$ are sideslip angle in rad, body x-axis angular velocity in rad/s, body z-axis angular velocity in rad/s, and roll angle in rad, and the dynamics transition from MP to NMP. Note that, in the case of full-state feedback, that is, $C = I_4$, (10.1)–(10.3) possess no zeros and thus no NMP zeros. For this example, however, output feedback is assumed, and thus (10.1)–(10.3) may have NMP zeros. In addition, the measurements of the roll angle $\phi(t)$ are assumed to be noisy. The roll-angle command is given by

$$r_k = \begin{cases} 10 \sin 0.28 T_{\mathrm{s}} k \text{ deg}, & t < 250 \text{ s}, \\ 12 \sin 0.21 T_{\mathrm{s}} k \text{ deg}, & 250 \leq t < 400 \text{ s}, \\ -10 \text{ deg}, & 400 \leq t < 450 \text{ s}, \\ 10 \text{ deg}, & 450 \leq t < 500 \text{ s}, \\ -10 \text{ deg}, & t > 550 \text{ s}, \end{cases} \tag{10.4}$$

which is a harmonic signal that abruptly changes frequency, followed by a sequence of step commands. The instantaneous poles and zeros of $EG_u(s)$ and $EG_{\mathrm{d}}(\mathbf{q})$ as functions of t are shown in Figures 10.1(a) and 10.1(b), respectively. The dynamics (10.1)–(10.3)

and their discretization transition from MP to NMP. The signal $u(t) = \delta_{\mathrm{a}}(t)$ represents the

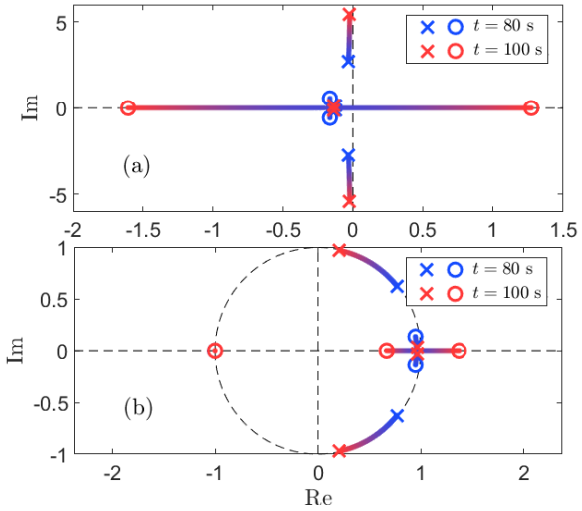

Figure 10.1: Example 25: Instantaneous (a) continuous- and (b) discrete-time poles and zeros of the hypersonic aircraft during the transition from 80 s to 100 s. The onset, duration, and time-dependence of the transition are assumed to be unknown.

asymmetric deflection of the split flaps in rad. The actuator rate-saturation and magnitude-saturation limits are 300 deg/s and 30 deg, respectively. Let $\overline{w}_{k,i}$ be Gaussian white noise with standard deviation 0.01 and mean 0.02, and let v_k be zero-mean, Gaussian white noise with standard deviation 0.001. The onset, duration, and time-dependence of the transition from MP to NMP dynamics, which occurs during $[80, 100]$ s, are assumed to be unknown to the control algorithm.

Adaptive control is applied with $E = 1$, $T_{\mathrm{s}} = 0.25$ s/step, $\tilde{y}_k \stackrel{\triangle}{=} z_k$, $p_{\mathrm{c},0} = 10$, $\eta = 12$, $n_{\mathrm{c}} = 12$, $E_z = 1$, $E_u = 0$, $E_{\Delta u} = 0.1$, $\varepsilon = 0.01$, $\tau_{\mathrm{n}} = 60$, $\tau_{\mathrm{d}} = 300$, and $\bar{u} = 30$ deg. The response to the command (10.4) in the presence of disturbance is shown in Figure 10.2. By adapting to the unknown, changing dynamics in $80 \leq t < 100$ s, RLSID and RLSAC are able to follow commands. ◇

Example 26. *Pitch-rate control of a flexible aircraft.* Consider the pitch dynamics of

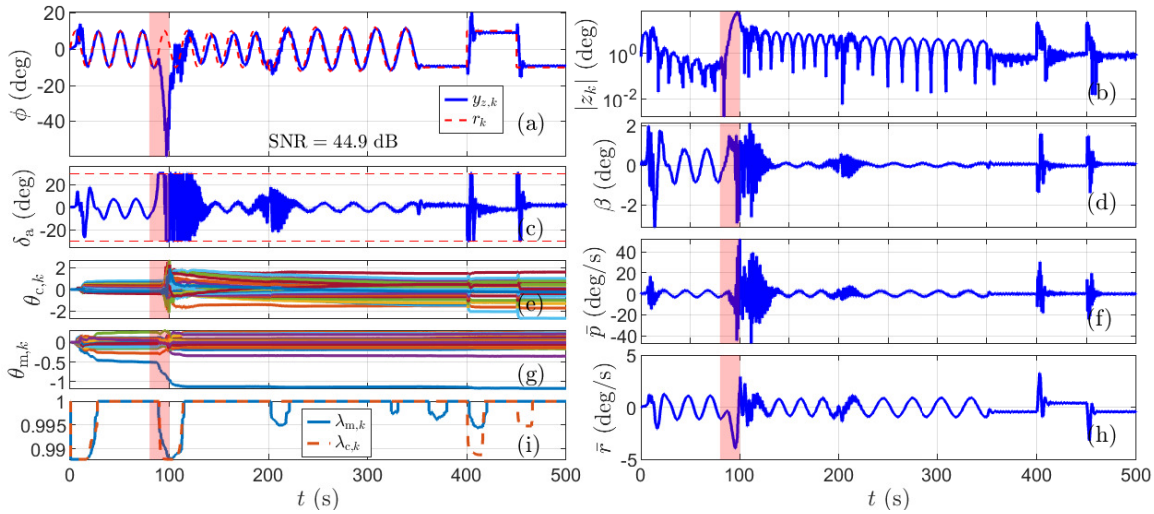

Figure 10.2: Example 25: Response of the lateral dynamics of a hypersonic aircraft to harmonic and step commands with an unknown transition from MP to NMP dynamics, which occurs within the shaded regions.

a flexible aircraft [63] given by

$$G_u(s) = -0.417\frac{s(s-0.0143)(s-0.4)\prod_{i=1}^{4}(s^2+2\bar{\zeta}_i\bar{\omega}_i s+\bar{\omega}_i^2)}{\prod_{i=1}^{6}(s^2+2\zeta_i\omega_i s+\omega_i^2)}, \tag{10.5}$$

where $\bar{\zeta}_1 = 0.0423$, $\bar{\zeta}_2 = 0.147$, $\bar{\zeta}_3 = 0.0136$, $\bar{\zeta}_4 = 0.0125$, $\bar{\omega}_1 = 4.883$, $\bar{\omega}_2 = 17.79$, $\bar{\omega}_3 = 22.04$, $\bar{\omega}_4 = 23.59$, $\zeta_1 = 0.0951$, $\zeta_2 = 0.0358$, $\zeta_3 = 0.0374$, $\zeta_4 = 0.149$, $\zeta_5 = 0.021$, $\zeta_6 = 0.0136$, $\omega_1 = 0.0551$, $\omega_2 = 1.830$, $\omega_3 = 12.40$, $\omega_4 = 18.03$, $\omega_5 = 21.25$, and $\omega_6 = 22.04$. This system represents a flexible aircraft cruising at Mach 0.6 at 5000 ft, and includes aeroelastic effects. The transfer function (10.5) is lightly damped, asymptotically stable, and MP. This transfer function relates the elevator deflection δ_{e} in deg to the pitch rate $\bar{q}$ measured at the cockpit in rad/s. The actuator rate-saturation and magnitude-saturation limits are 300 deg/s and 30 deg, respectively.

Assume that $G_u(s) = G_w(s)$ and let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise

with standard deviations 0.1 and 0.001, respectively. The pitch-rate command is

$$r_k = \begin{cases} 4 \text{ deg/s}, & t < 30 \text{ s}, \\ 0 \text{ deg/s}, & 30 \leq t < 60 \text{ s}, \\ -4 \text{ deg/s}, & 60 \leq t < 90 \text{ s}, \\ 0 \text{ deg/s}, & 90 \leq t < 120 \text{ s}, \\ 4 \text{ deg/s}, & 120 \leq t < 150 \text{ s} \\ 0 \text{ deg/s}, & t \geq 150\text{s}. \end{cases} \tag{10.6}$$

For this example, the adaptive controller is configured for command feedforward by defining

$$\tilde{y}_k \triangleq \begin{bmatrix} z_k \\ r_k \end{bmatrix}. \tag{10.7}$$

Adaptive control is applied with $T_\text{s} = 0.1$ s/step, $E = 1$, $p_{\text{c},0} = 10^4$, $\eta = 8$, $n_\text{c} = 30$, $E_z = 1$, $E_u = 0$, $E_{\Delta u} = 0.01$, $\varepsilon = 0.02$, $\tau_\text{n} = 60$, $\tau_\text{d} = 240$, and $\bar{u} = 30$ deg. The response to a sequence of step commands in the presence of zero-mean, Gaussian white-noise disturbance is shown in Figure 10.3. ◇

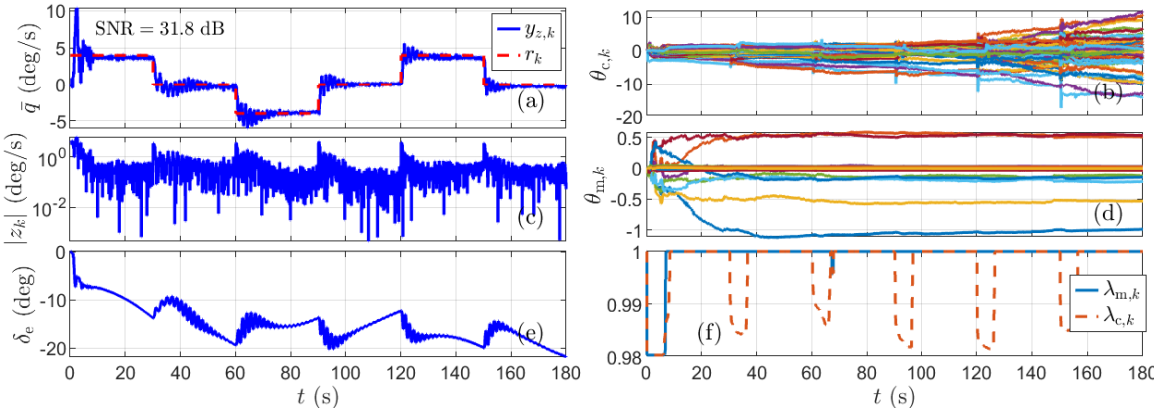

Figure 10.3: Example 26: Response of the flexible aircraft to a sequence of pitch-rate step commands.

Example 27. *Flutter suppression.* Consider the Benchmark Active Control Tech-

nology (BACT) for Active Control Design Applications [64, 65], which represents a wind-tunnel mounted wing that can translate vertically and pitch, and has a trailing edge flap as a control surface, as shown in Figure 10.4. The BACT model incorporates a vertical spring and damper to model vertical aerodynamic forces, as well as rotational spring and damper to model aerodynamic torques. Accelerometers mounted on the leading and

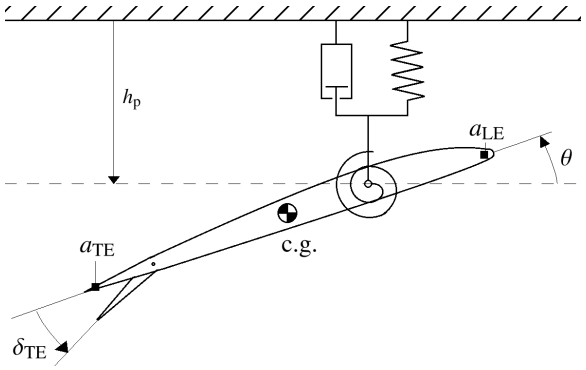

Figure 10.4: Example 27: BACT wing. Leading- and trailing-edge accelerometers measure a_{LE} and a_{TE}. The wing can plunge and pitch. The actuator is a trailing-edge control surface with deflection δ_{TE}.

trailing edges of the wing measure the leading-edge normal acceleration a_{LE} and trailing-edge normal acceleration a_{TE}, respectively. The flutter-suppression objective is to drive a_{LE} and a_{TE} to 0 using the control surface deflection δ_{TE}, in the presence of turbulence. Second-order actuator dynamics and a second-order Dryden wind turbulence model are included in BACT. The disturbance $\overline{w}_{k,i}$ represents the input to the second-order Dryden wind-turbulence model. BACT is an 8th-order, two-output-one-input, continuous-time, unstable, linear time-varying system with nonzero matrix D, whose state-space matrices are functions of the freestream velocity U_0. For this example the freestream velocity is varied as

$$U_0 = \begin{cases} 300 \text{ ft/s}, & t < 2 \text{ s}, \\ 300 + 25(t-2) \text{ ft/s}, & 2 \leq t < 6 \text{ s}, \\ 400 \text{ ft/s}, & t \geq 6\text{s}. \end{cases} \tag{10.8}$$

The onset, duration, and time-dependence of the change of freestream velocity, which oc-

curs during $[2, 6]$ s, are assumed to be unknown to the control algorithm. The details of BACT are found in [65].

Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 1 and 0.05, respectively. Adaptive control is applied with $T_{\mathrm{s}} = 0.02$ s/step, $E = I_2$, $\tilde{y}_k \triangleq z_k$, $r_k = [\ 0\ 0\ ]^{\mathrm{T}}$, $p_{\mathrm{c},0} = 100$, $\eta = 2$, $n_{\mathrm{c}} = 12$, $E_z = I_2$, $E_u = 1$, $E_{\Delta u} = 0$, $\varepsilon = 0.01$, $\tau_{\mathrm{n}} = 40$, $\tau_{\mathrm{d}} = 200$, and $\bar{u} = 12$ deg. The open- and closed-loop responses to a zero-mean, Gaussian white-noise disturbance are shown in Figure 10.5. Note that the signal-to-noise ratio between the the sampled noisy acceleration measurements a_{LE} and a_{TE}, and the sensor noise v_k is approximately 13 dB. That is, the sensor noise root-mean-squared value is approximately 23% as large as the root-mean-squared value of the acceleration measurements.

Furthermore, let $G_{\mathrm{d}}(\mathbf{q})$ represent an exact discretization of the BACT model at $t = 10$ s. Note that the BACT model is a non-square 2×1 system, and therefore, $\mathrm{CZ}(G_{\mathrm{c},k}, G_{\mathrm{d}})$ may be nonempty. Figure 10.6 plots the elements of $\mathrm{CZ}(G_{\mathrm{c},500}, G_{\mathrm{d}})$ and controller poles. No NMP elements of $\mathrm{CZ}(G_{\mathrm{c},500}, G_{\mathrm{d}})$ are cancelled by controller poles. ◇

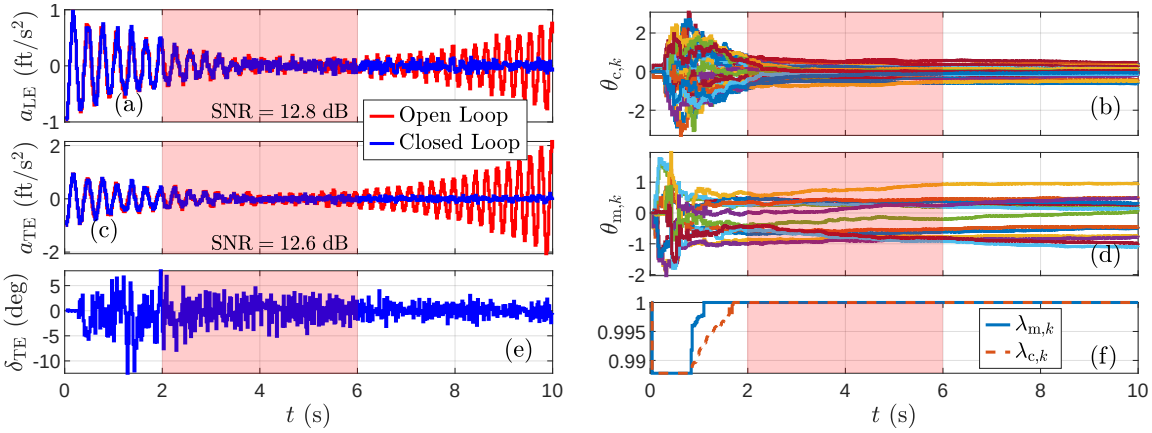

Figure 10.5: Example 27: Open- and closed-loop responses of a_{LE} and a_{TE}. The freestream velocity U_0 is varied in the shaded region.

Example 28. *Normal-acceleration control of a nonlinear planar missile.* Consider a tail-controlled interceptor missile, which is equipped with a strapdown accelerometer placed d_{a} meters forward of the center of mass of the missile, where the distance d_{a} is

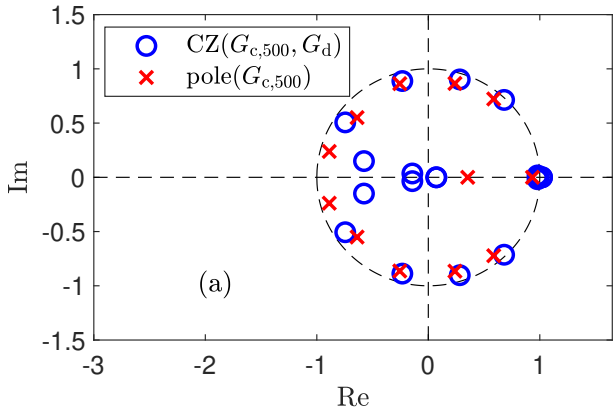

Figure 10.6: Example 27: $\mathrm{CZ}(G_{\mathrm{c},500}, G_{\mathrm{d}})$ and controller poles, where no NMP elements of $\mathrm{CZ}(G_{\mathrm{c},500}, G_{\mathrm{d}})$ are cancelled by a controller pole.

unknown. The missile [66–68] considered in this dissertation represents a missile in planar flight whose dynamics are given by

$$\dot{V} = \frac{1}{\bar{m}}\left[f_{\mathrm{d}}(C_{X\alpha}\cos\alpha + C_{Z\alpha}\sin\alpha) + T\cos\alpha - \bar{m}g\sin\gamma\right] + \frac{1}{\bar{m}}f_{\mathrm{d}}\sin(\alpha)C_{Z\delta}\delta, \quad (10.9)$$

$$\dot{\alpha} = \frac{1}{\bar{m}V}\left[f_{\mathrm{d}}(C_{Z\alpha}\cos\alpha - C_{X\alpha}\sin\alpha) - T\sin\alpha + \bar{m}V\bar{q} + \bar{m}g\cos\gamma\right] + \frac{1}{\bar{m}V}f_{\mathrm{d}}\cos(\alpha)C_{Z\delta}\delta + w, \quad (10.10)$$

$$\dot{\bar{q}} = \frac{d}{I_{yy}}f_{\mathrm{d}}(C_{M\alpha} + C_{Mq}\bar{q}) + \frac{d}{I_{yy}}f_{\mathrm{d}}C_{M\delta}\delta, \quad (10.11)$$

$$\dot{\gamma} = \frac{1}{\bar{m}V}\left[f_{\mathrm{d}}(C_{X\alpha}\sin\alpha - C_{Z\alpha}\cos\alpha) + T\sin\alpha - \bar{m}g\cos\gamma\right] - \frac{1}{\bar{m}V}f_{\mathrm{d}}\cos(\alpha)C_{Z\delta}\delta, \quad (10.12)$$

$$\dot{h} = V\sin\gamma, \quad (10.13)$$

where arguments of t are omitted for brevity, $V(t)$ is the missile speed in $\mathrm{m/s}$, T is the thrust in N, g is the acceleration due to gravity in $\mathrm{m/s^2}$, $\alpha(t)$ is the angle of attack in rad, $\bar{q}(t)$ is the y-axis angular velocity in rad/s, $\gamma(t)$ is the flight-path angle in rad, $h(t)$ is the altitude in m, $\delta(t)$ is the applied fin angle in rad, $f_{\mathrm{d}} \triangleq \frac{1}{2}\rho V(t)^2 S$ is the dynamic force in N, $\rho(t) = \rho(h(t))$ is the air density in $\mathrm{kg/m^3}$ at an altitude $h(t)$ m given by the Internal Standard Atmosphere model, S is the reference surface area in $\mathrm{m^2}$, d is the reference length

in m, $\bar{m}$ is the mass of the missile in kg, and I_{yy} is the moment of inertia of the missile relative to its center of mass and around a transverse axis in kg-m^2. The angles α, γ, θ, and $\delta_{\rm f}$ are shown in Figure 10.7. The values of the aerodynamic coefficients and parameter values are given in Tables 10.1 and 10.2, respectively. Note that the aerodynamic coefficients

Table 10.1: Aerodynamic coefficients. α is the angle of attack in rad, V is the missile speed in m/s, and $a_{\rm s} = a_{\rm s}(h)$ is the local speed of sound given by the Internal Standard Atmosphere model at the altitude h.

Aerodynamic Coefficient	Value	Units
$C_{X\alpha}$	-0.3005	-
$C_{Z\alpha}$	$9.717(\frac{V}{3a_{\rm s}} - 2)\alpha - 31.023\alpha\|\alpha\| + 19.373\alpha^3$	-
$C_{M\alpha}$	$2.922(\frac{8V}{3a_{\rm s}} - 7)\alpha - 64.015\alpha\|\alpha\| + 40.440\alpha^3$	-
$C_{Z\delta}$	-1.948	-
$C_{M\delta}$	-11.803	-
C_{Mq}	-1.719	s

Table 10.2: Parameter values for the nonlinear planar missile.

Parameter	Value	Units
$\bar{m}$	204.0227	kg
I_{yy}	247.4366	kg-$\rm m^2$
g	9.81	$\rm m/s^2$
S	0.0409	$\rm m^2$
d	0.2286	m
T	1000	N
$d_{\rm a}$	0.5	m

are nonlinear functions of the missile speed $V(t)$, angle of attack $\alpha(t)$, and the local speed of sound $a_{\rm s}$, which depends on the altitude $h(t)$. The applied fin angle $\delta(t)$ is related to the requested fin angle $u_k = \delta_{\rm r}(kT_{\rm s})$ by means of second-order actuator dynamics with natural frequency 150 rad/s, damping ratio 0.7, and magnitude and rate limits 30 deg and 500 deg/sec, respectively. The gravity-corrected normal acceleration measured by an accelerometer placed at a distance $d_{\rm a}$ forward of the center of mass of the missile is given by

$$n_z = f_{\rm d}(\mu C_{Z\alpha} - \mu_y C_{M\alpha} - \mu_y C_{Mq}\bar{q}) + f_{\rm d}(\mu C_{Z\delta} - \mu_y C_{M\delta})\delta, \tag{10.14}$$

where $\mu = \frac{1}{\bar{m}}$, and $\mu_y = \frac{dd_{\rm a}}{I_{yy}}$. A noisy measurement $y_k = n_z(kT_{\rm s}) + v_k$, of the normal

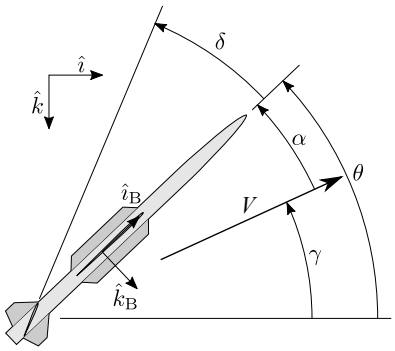

Figure 10.7: Example 28: $(\hat{\imath}, \hat{k})$ and $(\hat{\imath}_{\mathrm{B}}, \hat{k}_{\mathrm{B}})$ are Earth-fixed and body-fixed unit vectors, δ is the fin deflection, α is the angle of attack, V is the missile velocity vector, γ is the flight-path angle, and θ is the pitch angle.

acceleration $n_z(t)$, is used by the controller. The output equation (10.14) shows that there is a direct feedthrough of the applied fin $\delta(t)$ to the normal acceleration used by the controller.

For this example, the adaptive controller is configured for command feedforward by defining

$$\tilde{y}_k \triangleq \begin{bmatrix} z_k \\ r_k \end{bmatrix}, \tag{10.15}$$

where the normal-acceleration command is $r_k = 100 \sin 0.025k^{1.2}$ m/s^2. Let $\overline{w}_{k,i}$ and v_k be zero-mean, Gaussian white noise with standard deviations 0.01 and 0.1, respectively. Furthermore, let $V(0) = 985.7$ m/s, $\alpha(0) = 0$ rad, $\bar{q}(0) = 0$ rad/s, $\gamma(0) = \frac{\pi}{4}$ rad, and $h(0) = 3000$ m. Adaptive control is applied with $T_{\mathrm{s}} = 0.05$ s/step, $E = 1$, $p_{\mathrm{c},0} = 10^3$, $\eta = 4$, $n_{\mathrm{c}} = 4$, $E_z = 1$, $E_u = 0$, $E_{\Delta u} = 0.005$, $\varepsilon = 0.5$, $\tau_{\mathrm{n}} = 20$, $\tau_{\mathrm{d}} = 60$, and $\bar{u} = 30$ deg. The command-following response of the nonlinear planar missile is shown in Figure 10.8. After an initial transient, the command-following error is less than 5 g. Note that, starting with no prior knowledge of the nonlinear dynamics (10.9)–(10.13), the adaptive controller converges to a controller that facilitates command following. $\diamond$

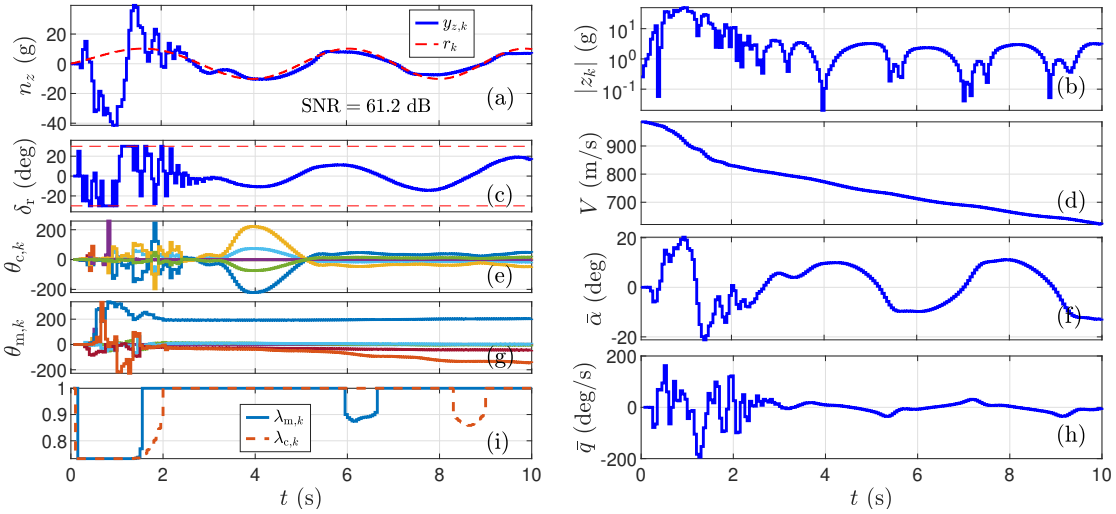

Figure 10.8: Example 28: Normal-acceleration command-following response of the nonlinear planar missile.

CHAPTER 11

Real-Time Implementation of the Optimal Predictor and Optimal Filter: Accuracy versus Latency

This chapter explores the digital optimal two-step predictor and the optimal two-step filter [69], more commonly known as a Kalman filter. Although the Kalman filter is often presented within a continuous-time context [70–73], the original derivation was carried out in discrete time [74]. In practice, controllers and observers are invariably implemented digitally, and thus discrete-time algorithms deserve special attention. This chapter focuses on the real-time implementation of the discrete-time Kalman filter and its relation to the discrete-time Kalman predictor. These algorithms are almost identical but differ in subtle ways, as highlighted in this chapter. Equations for the discrete-time Kalman filter and predictor are given in [75]; unfortunately, the covariance matrix becomes indefinite after a few steps, which shows that these equations are erroneous. This chapter thus corrects and extends the results of [75].

This derivation is based on necessary conditions and thus does not prove that OOSP is the globally optimal predictor; however, this derivation is succinct and uses minimal mathematics, making it efficient for classroom presentation. In fact, as shown in [71, p. 46], in the absence of Gaussian processes, this derivation yields the optimal linear predictor. Next, we recast OOSP as an optimal two-step predictor (OTSP). We then reverse the steps

of OTSP to obtain the optimal two-step filter (OTSF). Finally, we combine the steps of OTSF to obtain the optimal one-step filter (OOSF).

The optimal two-step predictor and the optimal two-step filter both consist of assimilation and forecast updates. In particular, OTSP begins with an assimilation update followed by a forecast update, whereas OTSF begins with a forecast update followed by an assimilation update. An additional distinction between these two algorithms concerns their real-time implementation. In particular, within the context of constraints on data collection and computation, OTSP and OTSF use different data to produce state estimates that are available at different times. In a nutshell, the OTSF uses more recent data but produces state estimates at a later time; we thus expect the OTSF estimates to be more accurate than the OTSP estimates with latency the price paid for the enhanced accuracy of the filter.

To investigate the validity of the expected accuracy/latency tradeoff, we compare the accuracy of OTSP and OTSF by means of a numerical example. Somewhat surprisingly, the numerical results show that the filter estimates are not uniformly better than the predictor estimates. This discrepancy is traced to the use of the initial data. This suggests a third algorithm, namely, a two-step filter that invokes an assimilation update at startup (OTSFSU) in order to utilize the initial data. Aside from the initial startup step, OTSFSU is identical to OTSF. We then revisit the accuracy/latency tradeoff by numerically comparing the accuracy of OTSP and OTSFSU to determine whether OTSFSU is uniformly more accurate than OTSP.

11.1 The Optimal One-Step Predictor

Consider the system

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \tag{11.1}$$

$$y_k = C_k x_k + D_k u_k + v_k, \tag{11.2}$$

where, for all $k \geq 0$, $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$, $w_k \in \mathbb{R}^n$, and $v_k \in \mathbb{R}^p$, and the real matrices A_k, B_k, C_k, D_k are of corresponding size. All stochastic processes are assumed to have finite second moments; the details of the densities are not relevant to the derivation below. In particular, $w_k \in \mathbb{R}^n$ is a zero-mean white noise disturbance signal with covariance $Q_k \stackrel{\triangle}{=} \mathbb{E}[w_k w_k^{\mathrm{T}}] \in \mathbb{R}^{n \times n}$, and $v_k \in \mathbb{R}^p$ is zero-mean white sensor noise with covariance $R_k \stackrel{\triangle}{=} \mathbb{E}[v_k v_k^{\mathrm{T}}] \in \mathbb{R}^{p \times p}$. The cross-covariance between v_k and w_k is denoted by $S_k \stackrel{\triangle}{=} \mathbb{E}[w_k v_k^{\mathrm{T}}] \in \mathbb{R}^{n \times p}$. The input u_k is assumed to be known; in practice, u_k is typically a control input. We consider the *optimal one-step predictor* (OOSP)

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + K_k(y_k - \hat{y}_k), \tag{11.3}$$

$$\hat{y}_k = C_k \hat{x}_k + D_k u_k, \tag{11.4}$$

where $\hat{x}_k \in \mathbb{R}^n$ is an estimate of the state x_k and the optimal gain $K_k \in \mathbb{R}^{n \times p}$ is determined below.

Defining the state error

$$e_k \stackrel{\triangle}{=} x_k - \hat{x}_k, \tag{11.5}$$

it follows that

$$\begin{aligned} e_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= \tilde{A}_k e_k + \tilde{w}_k, \end{aligned} \tag{11.6}$$

where

$$\tilde{A}_k \stackrel{\triangle}{=} A_k - K_k C_k, \tag{11.7}$$

$$\tilde{w}_k \stackrel{\triangle}{=} w_k - K_k v_k. \tag{11.8}$$

Next, we define the cost

$$\begin{aligned} J_k(K_k) &\stackrel{\triangle}{=} \mathbb{E}[e_{k+1}^{\mathrm{T}} e_{k+1}] \\ &= \operatorname{tr} \mathbb{E}[e_{k+1}^{\mathrm{T}} e_{k+1}] \\ &= \mathbb{E}[\operatorname{tr}(e_{k+1}^{\mathrm{T}} e_{k+1})] \\ &= \mathbb{E}[\operatorname{tr}(e_{k+1} e_{k+1}^{\mathrm{T}})] \\ &= \operatorname{tr} \mathbb{E}[e_{k+1} e_{k+1}^{\mathrm{T}}] \\ &= \operatorname{tr} P_{k+1}, \end{aligned} \tag{11.9}$$

where the *error covariance* is defined by

$$P_{k+1} \stackrel{\triangle}{=} \mathbb{E}[e_{k+1} e_{k+1}^{\mathrm{T}}] \in \mathbb{R}^{n \times n}. \tag{11.10}$$

It thus follows that

$$\begin{aligned} e_{k+1} e_{k+1}^{\mathrm{T}} &= (\tilde{A}_k e_k + \tilde{w}_k)(\tilde{A}_k e_k + \tilde{w}_k)^{\mathrm{T}} \\ &= \tilde{A}_k e_k e_k^{\mathrm{T}} \tilde{A}_k^{\mathrm{T}} + \tilde{A}_k e_k \tilde{w}_k^{\mathrm{T}} + \tilde{w}_k e_k^{\mathrm{T}} \tilde{A}_k^{\mathrm{T}} + \tilde{w}_k \tilde{w}_k^{\mathrm{T}}. \end{aligned} \tag{11.11}$$

Next, since w_k and v_k are white noise sequences that affect e_{k+1} but not e_k, it follows that e_k and $\tilde{w}_k$ are uncorrelated. Furthermore, since w_k and v_k have zero mean, it follows that $\tilde{w}_k$ also has zero mean. Therefore, $\mathbb{E}[e_k \tilde{w}_k^{\mathrm{T}}] = \mathbb{E}[e_k]\mathbb{E}[\tilde{w}_k^{\mathrm{T}}] = 0$, Now, taking the expected

value of (11.11) yields

$$\begin{aligned}
P_{k+1} &= \mathbb{E}[e_{k+1}e_{k+1}^{\mathrm{T}}] \\
&= \mathbb{E}[\tilde{A}_k e_k e_k^{\mathrm{T}} \tilde{A}_k^{\mathrm{T}} + \tilde{A}_k e_k \tilde{w}_k^{\mathrm{T}} + \tilde{w}_k e_k^{\mathrm{T}} \tilde{A}_k^{\mathrm{T}} + \tilde{w}_k \tilde{w}_k^{\mathrm{T}}] \\
&= \tilde{A}_k \mathbb{E}[e_k e_k^{\mathrm{T}}] \tilde{A}_k^{\mathrm{T}} + \tilde{A}_k \mathbb{E}[e_k \tilde{w}_k^{\mathrm{T}}] + \mathbb{E}[\tilde{w}_k e_k^{\mathrm{T}}] \tilde{A}_k^{\mathrm{T}} + \mathbb{E}[\tilde{w}_k \tilde{w}_k^{\mathrm{T}}] \\
&= \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \mathbb{E}[\tilde{w}_k \tilde{w}_k^{\mathrm{T}}] \\
&= \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \mathbb{E}[(w_k - K_k v_k)(w_k - K_k v_k)^{\mathrm{T}}] \\
&= \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \mathbb{E}[w_k w_k^{\mathrm{T}} - w_k v_k^{\mathrm{T}} K_k^{\mathrm{T}} - K_k v_k w_k^{\mathrm{T}} + K_k v_k v_k^{\mathrm{T}} K_k^{\mathrm{T}}] \\
&= \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \mathbb{E}[w_k w_k^{\mathrm{T}}] - \mathbb{E}[w_k v_k^{\mathrm{T}}] K_k^{\mathrm{T}} - K_k \mathbb{E}[v_k w_k^{\mathrm{T}}] + K_k \mathbb{E}[v_k v_k^{\mathrm{T}}] K_k^{\mathrm{T}} \\
&= \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + Q_k + K_k R_k K_k^{\mathrm{T}} - \mathbb{E}[w_k v_k^{\mathrm{T}}] K_k^{\mathrm{T}} - K_k [\mathbb{E}[w_k v_k^{\mathrm{T}}]]^{\mathrm{T}}. \qquad (11.12)
\end{aligned}$$

The cost (11.9) is thus given by

$$\begin{aligned}
J_k(K_k) &= \operatorname{tr} P_{k+1} \\
&= \operatorname{tr}[(A_k - K_k C_k) P_k (A_k - K_k C_k)^{\mathrm{T}} + Q_k + K_k R_k K_k^{\mathrm{T}} - S_k K_k^{\mathrm{T}} - K_k S_k^{\mathrm{T}}] \\
&= \operatorname{tr}[A_k P_k A_k^{\mathrm{T}} - K_k C_k P_k A_k^{\mathrm{T}} - A_k P_k C_k^{\mathrm{T}} K_k^{\mathrm{T}} + K_k C_k P_k C_k^{\mathrm{T}} K_k^{\mathrm{T}} + Q_k + K_k R_k K_k^{\mathrm{T}} \\
&\quad - S_k K_k^{\mathrm{T}} - K_k S_k^{\mathrm{T}}] \\
&= \operatorname{tr}[K_k (C_k P_k C_k^{\mathrm{T}} + R_k) K_k^{\mathrm{T}} - K_k C_k P_k A_k^{\mathrm{T}} - A_k P_k C_k^{\mathrm{T}} K_k^{\mathrm{T}} + A_k P_k A_k^{\mathrm{T}} + Q_k \\
&\quad - S_k K_k^{\mathrm{T}} - K_k S_k^{\mathrm{T}}] \\
&= \operatorname{tr}[K_k (C_k P_k C_k^{\mathrm{T}} + R_k) K_k^{\mathrm{T}}] - 2 \operatorname{tr} K_k (C_k P_k A_k^{\mathrm{T}} + S_k^{\mathrm{T}}) + \operatorname{tr}(A_k P_k A_k^{\mathrm{T}} + Q_k).
\end{aligned} \qquad (11.13)$$

To minimize $J_k(K_k)$, note that

$$\frac{\mathrm{d}J_k(K_k)}{\mathrm{d}K_k} = 2(C_k P_k C_k^{\mathrm{T}} + R_k) K_k^{\mathrm{T}} - 2 C_k P_k A_k^{\mathrm{T}} - 2 S_k^{\mathrm{T}}. \qquad (11.14)$$

Setting the derivative to zero yields the *optimal one-step predictor gain*

$$K_k = (A_k P_k C_k^{\mathrm{T}} + S_k)(C_k P_k C_k^{\mathrm{T}} + R_k)^{-1}. \tag{11.15}$$

By substituting (11.7) and (11.15) in (11.12) the *one-step predictor* is thus given by

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + (A_k P_k C_k^{\mathrm{T}} + S_k)(C_k P_k C_k^{\mathrm{T}} + R_k)^{-1}(y_k - C_k \hat{x}_k), \tag{11.16}$$

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} - (A_k P_k C_k^{\mathrm{T}} + S_k)(C_k P_k C_k^{\mathrm{T}} + R_k)^{-1}(C_k P_k A_k^{\mathrm{T}} + S_k^{\mathrm{T}}) + Q_k. \tag{11.17}$$

Note that the error-covariance propagation equation is independent of data. Furthermore, (11.16) and (11.17) can be written in terms of K_k as

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + K_k(y_k - C_k \hat{x}_k), \tag{11.18}$$

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} - K_k(C_k P_k A_k^{\mathrm{T}} + S_k^{\mathrm{T}}) + Q_k. \tag{11.19}$$

11.2 The Optimal Two-Step Predictor

As an alternative but equivalent implementation of the optimal predictor, (11.16) and (11.17) can be implemented as the *optimal two-step predictor* (OTSP). For simplicity in this and subsequent sections, we consider the case where $S_k = 0$; the case where S_k is nonzero is discussed in the section "OTSP and OTSF with Correlated Disturbance and Sensor Noise." The *assimilation update* is given by

$$x_k^{\mathrm{a}} = x_k^{\mathrm{f}} + P_k^{\mathrm{f}} C_k^{\mathrm{T}}(C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}(y_k - C_k x_k^{\mathrm{f}}), \tag{11.20}$$

$$P_k^{\mathrm{a}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}}(C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}, \tag{11.21}$$

and the *forecast update* is given by

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{a}} + B_k u_k, \tag{11.22}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k. \tag{11.23}$$

Figure 11.1 shows a timing diagram for the real-time implementation of OTSP. By sub-

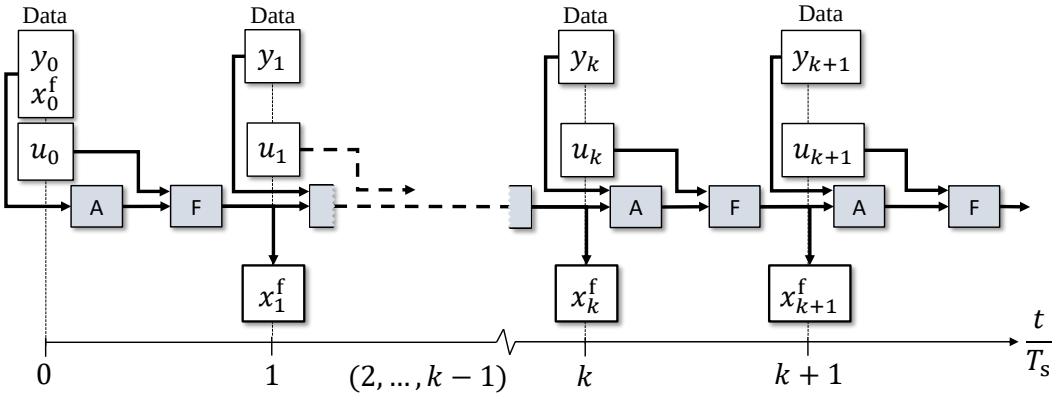

Figure 11.1: Timing diagram for the optimal two-step predictor (OTSP), where "A" denotes an assimilation update and "F" denotes a forecast update. The measurement y_k and the applied control u_k are available at step k for computation. For all $k \geq 1$, OTSP produces an estimate of x_k at step k without latency.

stituting (11.20) into (11.22) and (11.21) into (11.23), it can be seen that $x_k^{\mathrm{f}} = \hat{x}_k$ and $P_k^{\mathrm{f}} = P_k$. Defining the *OTSP gain*

$$K_k^{\mathrm{f}} \triangleq P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{11.24}$$

(11.20) and (11.21) can be written as

$$x_k^{\mathrm{a}} = x_k^{\mathrm{f}} + K_k (y_k - C_k x_k^{\mathrm{f}}), \tag{11.25}$$

$$P_k^{\mathrm{a}} = (I - K_k C_k) P_k^{\mathrm{f}}. \tag{11.26}$$

It is interesting to note that the computational requirements of OTSP are more burdensome than OOSP, and yet OTSP and OOSP produce identical state estimates. Therefore,

there is no implementation advantage of OTSP over OOSP. However, the value of OTSP is in providing a framework for the optimal two-step filter, as discussed next.

11.3 The Optimal Two-Step Filter

In contrast to the two-step optimal predictor (11.20)–(11.23), the *optimal two-step filter* (OTSF) is given by the *forecast update*

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{a}} + B_k u_k, \tag{11.27}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k, \tag{11.28}$$

and the *assimilation update*

$$x_{k+1}^{\mathrm{a}} = x_{k+1}^{\mathrm{f}} + P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} (C_{k+1} P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1} (y_{k+1} - C_{k+1} x_{k+1}^{\mathrm{f}}), \tag{11.29}$$

$$P_{k+1}^{\mathrm{a}} = P_{k+1}^{\mathrm{f}} - P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} (C_{k+1} P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1} C_{k+1} P_{k+1}^{\mathrm{f}}. \tag{11.30}$$

Note that (11.20), (11.21) are slightly different from (11.29), (11.30); specifically, (11.20), (11.21) use the measurement y_k, whereas (11.29), (11.30) use the measurement y_{k+1}. Moreover, the forecast and assimilation updates of OTSF appear in reverse order compared to the forecast and assimilation updates of OTSP; this reversal explains the index k in (11.20), (11.21) and the index $k+1$ in (11.29), (11.30).

Next, note that, since x_{k+1}^{a} depends on y_{k+1} and since the required computation cannot be performed instantaneously, the estimate x_{k+1}^{a} of x_{k+1} is not available at step $k+1$ but rather at some time after step $k+1$, where the latency depends on the computer speed and architecture. Therefore, if the latency of x_k^{a} as an estimate of x_k is critical in a real-time application, then OTSP may be a better choice than OTSF. Figure 11.2 shows a timing diagram for the real-time implementation of OTSF.

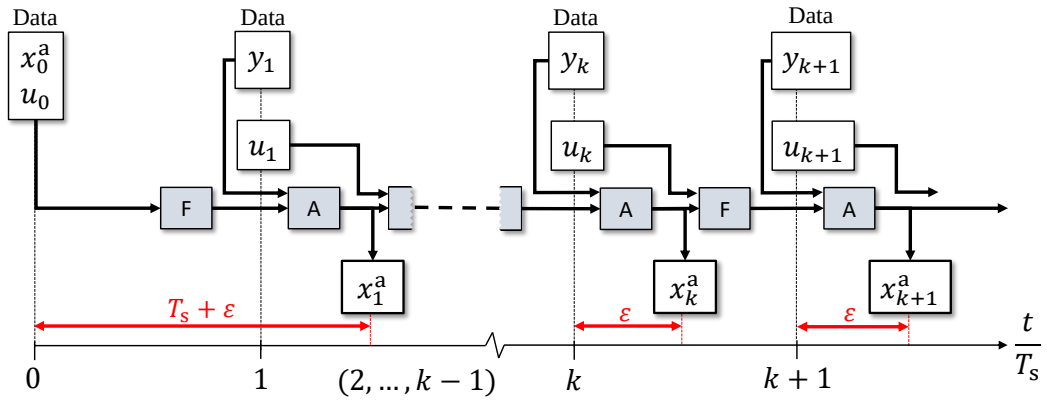

Figure 11.2: Timing diagram for the optimal two-step filter (OTSF), where "A" denotes an assimilation update and "F" denotes a forecast update. The measurement y_k and the applied control u_k are available at step k for computation. For all $k \geq 1$, OTSF produces an estimate of x_k with latency ε.

In terms of the *OTSF gain*

$$K_{k+1}^{\mathrm{f}} \stackrel{\triangle}{=} P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} (C_{k+1} P_{k+1}^{\mathrm{f}} C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1}, \tag{11.31}$$

(11.29) and (11.30) can be written as

$$x_{k+1}^{\mathrm{a}} = x_{k+1}^{\mathrm{f}} + K_{k+1}(y_{k+1} - C_{k+1} x_{k+1}^{\mathrm{f}}), \tag{11.32}$$

$$P_{k+1}^{\mathrm{a}} = (I - K_{k+1} C_{k+1}) P_{k+1}^{\mathrm{f}}. \tag{11.33}$$

Note that, aside from a shift in the index, the OTSF gain is identical to the OTSP gain.

11.4 The Optimal One-Step Filter

The optimal one-step filter (OOSF) can be obtained by substituting (11.27) and (11.28) into (11.29) and (11.30). In particular, OOSF is given by

$$\begin{aligned} x_{k+1}^{\mathrm{a}} &= A_k x_k^{\mathrm{a}} + B_k u_k \\ &\quad + (A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k) C_{k+1}^{\mathrm{T}} [C_{k+1}(A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k) C_{k+1}^{\mathrm{T}} + R_{k+1}]^{-1} \\ &\quad \cdot [y_{k+1} - C_{k+1}(A_k x_k^{\mathrm{a}} + B_k u_k)], \end{aligned} \tag{11.34}$$

$$\begin{aligned} P_{k+1}^{\mathrm{a}} &= A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k \\ &\quad - (A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k) C_{k+1}^{\mathrm{T}} \\ &\quad \cdot [C_{k+1}(A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k) C_{k+1}^{\mathrm{T}} + R_{k+1}]^{-1} C_{k+1}(A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} + Q_k). \end{aligned} \tag{11.35}$$

OTSF and the Traditional Kalman Filter

We now restate OTSF in standard notation in order to show that it is precisely the Kalman filter. Defining

$$\hat{x}_{k|k-1} \triangleq x_k^{\mathrm{f}}, \quad P_{k|k-1} \triangleq P_k^{\mathrm{f}}, \tag{11.36}$$

$$\hat{x}_{k|k} \triangleq x_k^{\mathrm{a}}, \quad P_{k|k} \triangleq P_k^{\mathrm{a}}, \tag{11.37}$$

(11.27)–(11.30) can be written as

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k} + B_k u_k, \tag{11.38}$$

$$P_{k+1|k} = A_k P_{k|k} A_k^{\mathrm{T}} + Q_k, \tag{11.39}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + P_{k+1|k} C_{k+1}^{\mathrm{T}} (C_{k+1} P_{k+1|k} C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1} (y_{k+1} - C_{k+1} \hat{x}_{k+1|k}), \tag{11.40}$$

$$P_{k+1|k+1} = P_{k+1|k} - P_{k+1|k} C_{k+1}^{\mathrm{T}} (C_{k+1} P_{k+1|k} C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1} C_{k+1} P_{k+1|k}, \tag{11.41}$$

which is the standard notation for the Kalman filter [71]. Furthermore, the OTSF gain (11.31) can be written as

$$K^{\mathrm{f}}_{k+1} = P^{\mathrm{f}}_{k+1|k} C^{\mathrm{T}}_{k+1} (C_{k+1} P^{\mathrm{f}}_{k+1|k} C^{\mathrm{T}}_{k+1} + R_{k+1})^{-1}, \tag{11.42}$$

and thus (11.40) and (11.41) can be written as

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K^{\mathrm{f}}_{k+1} (y_{k+1} - C_{k+1} \hat{x}_{k+1|k}), \tag{11.43}$$

$$P_{k+1|k+1} = (I - K^{\mathrm{f}}_{k+1} C_{k+1}) P_{k+1|k}. \tag{11.44}$$

An alternative notation [70, 72, 73] is given by

$$\hat{x}^{+}_{k} \triangleq x^{\mathrm{f}}_{k}, \quad P^{+}_{k} \triangleq P^{\mathrm{f}}_{k}, \tag{11.45}$$

$$\hat{x}^{-}_{k} \triangleq x^{\mathrm{a}}_{k}, \quad P^{-}_{k} \triangleq P^{\mathrm{a}}_{k}. \tag{11.46}$$

11.5 OTSP versus OTSF: Which Estimator Is More Accurate?

In this section we investigate whether or not the OTSF estimates are more accurate than the OTSP estimates. As an example, we consider an undamped oscillator with mass 4 kg and stiffness 2 N/m, sampled at 4 Hz using a zero-order hold input. The velocity is measured, and the position is to be estimated. The sampled-data system representing this

system has the dynamics

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9844 & 0.2487 \\ -0.1243 & 0.9844 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} 0.0078 \\ 0.0622 \end{bmatrix} u_k + w_k, \tag{11.47}$$

$$y_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + v_k, \tag{11.48}$$

where $x_k \in \mathbb{R}$ is the position in m, $y_k = \dot{x}_k \in \mathbb{R}$ is the velocity in m/s, u_k is the applied force in N, $w_k \in \mathbb{R}^2$ is the disturbance, and $v_k \in \mathbb{R}$ is the sensor noise.

To investigate the effect of the disturbance w_k, sensor noise v_k, and initial condition x_0 on the estimates of the position produced by OTSP and OTSF, let

$$u_k \sim \mathcal{N}(0,1), \quad w_k \sim \mathcal{N}(0, \alpha_{\mathrm{d}} I), \quad v_k \sim \mathcal{N}(0, \alpha_{\mathrm{sn}}), \quad x_0 \sim \mathcal{N}(0, \alpha_{\mathrm{ic}} I), \tag{11.49}$$

where $\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}}$ are varied one at a time with the remaining variables fixed. In particular, we first vary $\alpha_{\mathrm{d}} \in [10^{-5}, 10^{5}]$ with $\alpha_{\mathrm{sn}} = \alpha_{\mathrm{ic}} = 10^{-5}$ fixed. Next, we vary $\alpha_{\mathrm{sn}} \in [10^{-5}, 10^{5}]$ with $\alpha_{\mathrm{d}} = \alpha_{\mathrm{ic}} = 10^{-5}$ fixed. Finally, we vary $\alpha_{\mathrm{ic}} \in [10^{-5}, 10]$ with $\alpha_{\mathrm{d}} = \alpha_{\mathrm{sn}} = 10^{-5}$ fixed. In all cases, the initial states of OTSP and OTSF are set to zero and $P_0^{\mathrm{f}} = I_2$, $P_0^{\mathrm{a}} = I_2$.

In each case and for each choice of $\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}}$, 10,000 simulations are run with randomly generated values of u_k, w_k, v_k, and the initial condition x_0. For each simulation the root-mean-square (RMS) position-estimation errors for OTSP and OTSF are computed for $0 \leq k \leq 80$. The RMS position-estimation errors for OTSP and OTSF are then averaged over the 10,000 simulations. We define the RMS position-estimation errors for OTSP and OTSF to be $e_{\mathrm{P,RMS}}(\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}})$ and $e_{\mathrm{F,RMS}}(\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}})$, respectively. In particular, we

compute

$$e_{\mathrm{P,RMS}} = \frac{1}{10000}\sum_{j=1}^{10000}\sqrt{\frac{1}{80}\sum_{i=1}^{80}(x_{i,j}^{\mathrm{f}} - x_{i,j})^2}, \tag{11.50}$$

$$e_{\mathrm{F,RMS}} = \frac{1}{10000}\sum_{j=1}^{10000}\sqrt{\frac{1}{80}\sum_{i=1}^{80}(x_{i,j}^{\mathrm{a}} - x_{i,j})^2}, \tag{11.51}$$

where $x_{i,j}^{\mathrm{f}}$ is the position estimate produced by OTSP at the ith step for the jth simulation, $x_{i,j}^{\mathrm{a}}$ is the position estimate produced by OTSF at the ith step for the jth simulation, and $x_{i,j}$ is the true position at the ith step for the jth simulation. Furthermore, $|e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}|$ is plotted to show the accuracy of the OTSF position estimate relative to the OTSP position estimate in m. In particular, $e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}$ is positive in the case where the OTSF RMS position-estimation error is smaller than the OTSP RMS position-estimation error, and vice versa.

For the three cases considered above, Figures 11.4-11.6(a) show $e_{\mathrm{P,RMS}}, e_{\mathrm{F,RMS}}$ versus $\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}}$, respectively. Figures 11.4–11.6(b) show $|e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}|$ versus $\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}}$, respectively, where the values of $|e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}|$ are color coded based on the sign of $e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}$. Figure 11.6 shows that the OTSP position estimate is more accurate than the OTSF position estimate for all choices of α_{ic}. This situation is surprising since we expect the latency of the OTSF estimates to be offset by greater accuracy. We thus seek a variation of OTSF that produces the expected improved accuracy in return for latency, as discussed in the next section.

Optimal Two-Step Filter with Startup

As shown in Figure 11.6, OTSP may be more accurate than OTSF. This is surprising in view of the fact that the latency of the estimates produced by OTSF and its use of more recent data are expected to produce more accurate state estimates. This phenomenon can

be traced to the fact that OTSF does not use the measurement y_0. In fact, at step $k = 0$, (11.20) uses y_0, whereas, at step $k = 0$, (11.29) does not use y_0. This situation suggests the possibility of a variant of OTSF that employs an additional assimilation update before the initial forecast update; otherwise, all subsequent updates of OTSF with startup (OTSFSU) are identical to OTSF. Figure 11.3 shows a timing diagram for the real-time implementation of OTSFSU. The next section investigates the accuracy of OTSFSU compared to OTSP in

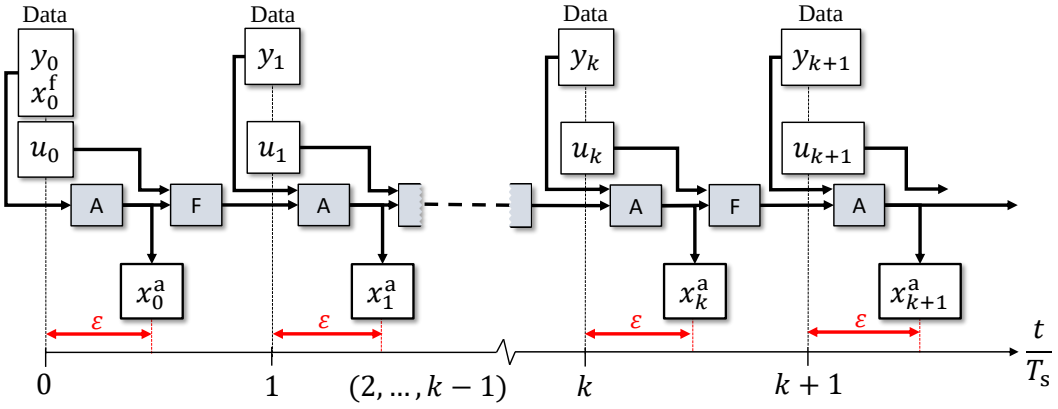

Figure 11.3: Timing diagram for the optimal two-step filter with startup (OTSFSU), where "A" denotes an assimilation update and "F" denotes a forecast update. The measurement y_k and the applied control u_k are available at step k for computation. For all $k \geq 0$, OTSFSU produces an estimate of x_k with latency ε.

order to determine whether or not OTSFSU is more accurate than OTSP as compensation for its inherent latency.

Real-Time Implementation of OTSP, OTSF, and OTSFSU

To clarify the timing of the computation required for OTSP, OTSF, and OTSFSU, note that for OTSP the state estimate is x_k^{f}, whereas, for OTSF and OTSFSU, the state estimate is x_k^{a}. Figures 11.1–11.3 show timing diagrams for the real-time implementation of OTSP, OTSF, and OTSFSU. Note that OTSP is the only estimator that produces an estimate of x_k at step k. In contrast, OTSF and OTSFSU produce estimates of x_k at time $kT_{\mathrm{s}} + \varepsilon$, where T_{s} is the sample time of the digital implementation and $\varepsilon \in (0, T_{\mathrm{s}}]$ is the latency, which is

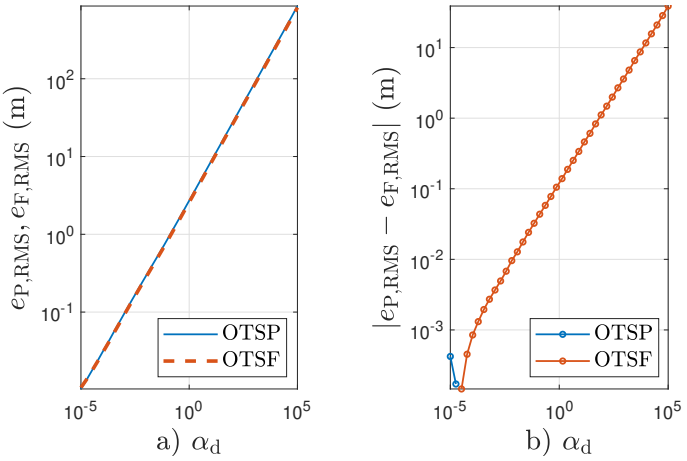

Figure 11.4: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter (OTSF) versus disturbance-covariance scaling α_{d}. a) shows the RMS position-estimation errors for OTSP and OTSF, $e_{\mathrm{P,RMS}}$ and $e_{\mathrm{F,RMS}}$, respectively. b) shows $|e_{\mathrm{P,RMS}} - e_{\mathrm{F,RMS}}|$. In all cases where $e_{\mathrm{P,RMS}} > e_{\mathrm{F,RMS}}$, OTSF is more accurate than OTSP; these are shown in red. In all cases where $e_{\mathrm{P,RMS}} < e_{\mathrm{F,RMS}}$, OTSP is more accurate than OTSF; these are shown in blue. Note that OTSP is more accurate than OTSF for $\alpha_{\mathrm{d}} < 10^{-4.5}$, whereas OTSF is more accurate than OTSP for $\alpha_{\mathrm{d}} > 10^{-4.5}$.

platform-dependent. As shown in Figure 11.2, the first state estimate produced by OTSF is an estimate of x_1; this estimate is not available until $t = T_{\mathrm{s}} + \varepsilon$. In addition, this estimate of x_1 does not use the data y_0, as can be seen by the absence of y_0 in Figure 11.2. On the other hand, as shown in Figure 11.3, OTSFSU uses y_0 to produce an estimate of x_0 at $t = \varepsilon$. Table 11.1 lists the data used by each algorithm and the associated latency.

Table 11.1: The initial estimate $\hat{x}_0$ and the real-time data y and u used by the estimators to estimate x_k. The time at which the estimate of x_k becomes available is given in terms of the step k, the sample time T_{s}, and the latency ε. OTSP, OTSF, and OTSFSU are the optimal two-step predictor, filter, and filter with startup, respectively.

Estimator	$\hat{x}_0$	y data	u data	$\hat{x}_k$	When available?
OTSP	x_0^{f}	y_{k-1}	u_{k-1}	x_k^{f}	kT_{s}
OTSF	x_0^{a}	y_k	u_{k-1}	x_k^{a}	$kT_{\mathrm{s}} + \varepsilon$
OTSFSU	x_0^{f}	y_k	u_{k-1}	x_k^{a}	$kT_{\mathrm{s}} + \varepsilon$

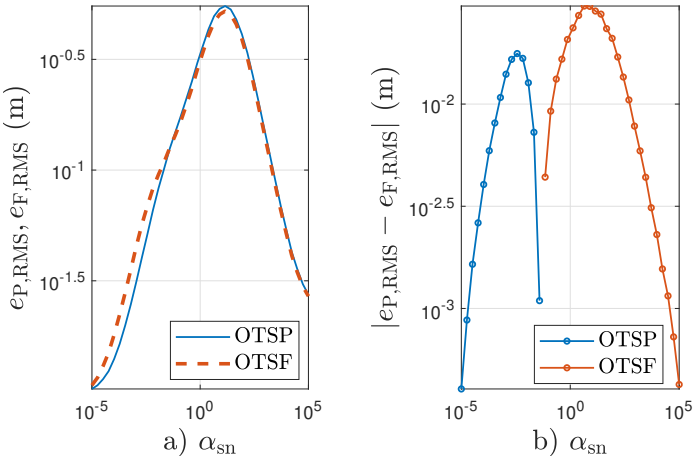

Figure 11.5: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter (OTSF) versus sensor-noise covariance α_{sn}. a) shows the RMS position-estimation errors for OTSP and OTSF, $e_{\mathrm{P,RMS}}$ and $e_{\mathrm{F,RMS}}$, respectively. b) shows that OTSP is more accurate than OTSF for $\alpha_{\mathrm{sn}} < 10^{-2}$, whereas OTSF is more accurate than OTSP for $\alpha_{\mathrm{sn}} > 10^{-2}$.

11.6 OTSP versus OTSFSU: Which Estimator Is More Accurate?

We now compare OTSP and OTSFSU using the same procedure used to compare OTSP and OTSF. For OTSFSU we define $e_{\mathrm{FSU,RMS}}(\alpha_{\mathrm{d}}, \alpha_{\mathrm{sn}}, \alpha_{\mathrm{ic}})$, and plot $|e_{\mathrm{P,RMS}} - e_{\mathrm{FSU,RMS}}|$. In particular, we compute

$$e_{\mathrm{FSU,RMS}} = \frac{1}{10000} \sum_{j=1}^{10000} \sqrt{\frac{1}{80} \sum_{i=1}^{80} (x_{i,j}^{\mathrm{a}} - x_{i,j})^2}, \tag{11.52}$$

where $x_{i,j}^{\mathrm{a}}$ is the position estimate produced by OTSFSU at the ith step for the jth simulation and $x_{i,j}$ is the true position at the ith step for the jth simulation. Figures 11.7–11.9 show that the OTSFSU position estimate is more accurate than the OTSP position estimate for all choices of α_{d} and α_{ic}. Furthermore, as shown in Figure 11.8, OTSP shows no distinct advantage over OTSFSU for variations of α_{sn}.

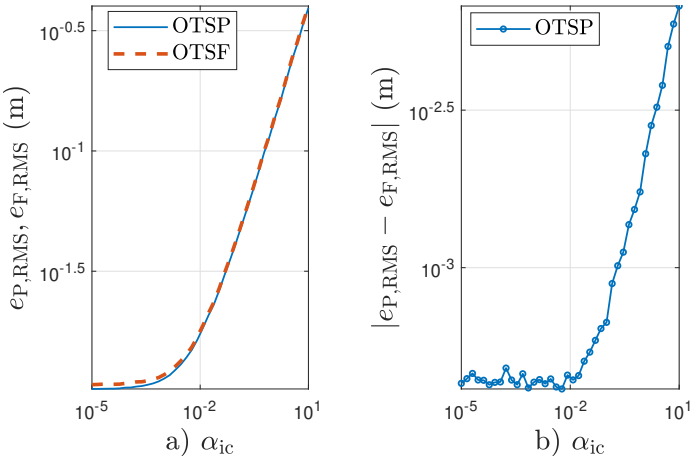

Figure 11.6: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter (OTSF) versus initial-condition-covariance scaling α_{ic}. a) shows RMS position-estimation errors for OTSP and OTSF, $e_{\mathrm{P,RMS}}$, and $e_{\mathrm{F,RMS}}$, respectively. b) shows that OTSP uniformly more accurate than OTSF.

11.7 OTSP and OTSF with Correlated Disturbance and Sensor Noise

This section gives the update equations for the optimal two-step predictor (OTSP) and the optimal two-step filter (OTSF) in the case where $S_k \neq 0$. In particular, for OTSP the *assimilation update* is given by

$$x_k^{\mathrm{a}} = x_k^{\mathrm{f}} + P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} (y_k - C_k x_k^{\mathrm{f}}), \tag{11.53}$$

$$P_k^{\mathrm{a}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}, \tag{11.54}$$

which are identical to (11.20), (11.21). The *forecast update* is given by

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{a}} + B_k u_k + S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} (y_k - C_k x_k^{\mathrm{f}}), \tag{11.55}$$

$$\begin{aligned} P_{k+1}^{\mathrm{f}} = {} & A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} - S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} \\ & - A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} \\ & - S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} + Q_k, \end{aligned} \tag{11.56}$$

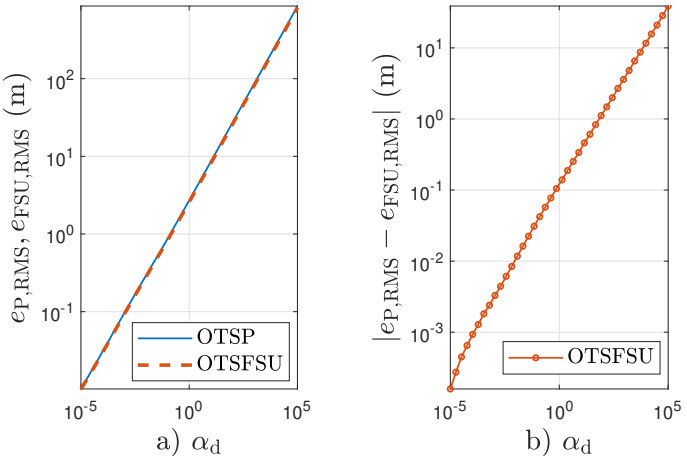

Figure 11.7: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter with startup (OTSFSU) versus disturbance-covariance scaling α_{d}. a) shows the RMS position-estimation errors for OTSP and OTSFSU, $e_{\mathrm{P,RMS}}$, and $e_{\mathrm{FSU,RMS}}$, respectively. b) shows that OTSFSU is uniformly more accurate than OTSP.

which are (11.22), (11.23) with additional terms involving S_k. Defining the *OTSP gain*

$$K_k^{\mathrm{f}} \triangleq P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{11.57}$$

(11.53) and (11.54) can be written as

$$x_k^{\mathrm{a}} = x_k^{\mathrm{f}} + K_k (y_k - C_k x_k^{\mathrm{f}}), \tag{11.58}$$

$$P_k^{\mathrm{a}} = (I - K_k C_k) P_k^{\mathrm{f}}, \tag{11.59}$$

which are identical to (11.25), (11.26).

For OTSF the *forecast update* is given by

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{a}} + B_k u_k + S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} (y_k - C_k x_k^{\mathrm{f}}), \tag{11.60}$$

$$\begin{aligned} P_{k+1}^{\mathrm{f}} = {} & A_k P_k^{\mathrm{a}} A_k^{\mathrm{T}} - S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} \\ & - A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} \\ & - S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} + Q_k, \end{aligned} \tag{11.61}$$

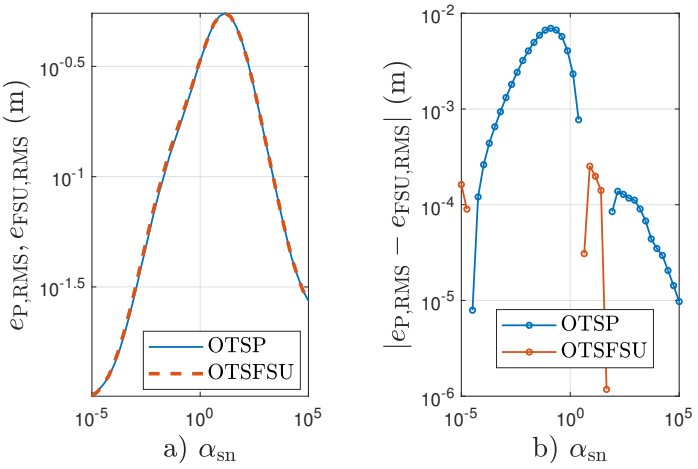

Figure 11.8: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter with startup (OTSFSU) versus sensor-noise-covariance scaling α_{sn}. a) shows the RMS position-estimation errors for OTSP and OTSFSU, $e_{\mathrm{P,RMS}}$, and $e_{\mathrm{FSU,RMS}}$, respectively. b) shows OTSP is more accurate than OTSFSU for $10^{-4.5} < \alpha_{\mathrm{sn}} < 10^{-2}$ and $\alpha_{\mathrm{sn}} > 10^{2}$.

which are (11.27), (11.28) with additional terms involving S_k. The *assimilation update* is given by

$$x^{\mathrm{a}}_{k+1} = x^{\mathrm{f}}_{k+1} + P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1}(C_{k+1}P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1} + R_{k+1})^{-1}(y_{k+1} - C_{k+1}x^{\mathrm{f}}_{k+1}), \tag{11.62}$$

$$P^{\mathrm{a}}_{k+1} = P^{\mathrm{f}}_{k+1} - P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1}(C_{k+1}P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1} + R_{k+1})^{-1}C_{k+1}P^{\mathrm{f}}_{k+1}, \tag{11.63}$$

which are identical to (11.29), (11.30). In terms of the *OTSF gain*

$$K^{\mathrm{f}}_{k+1} \triangleq P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1}(C_{k+1}P^{\mathrm{f}}_{k+1}C^{\mathrm{T}}_{k+1} + R_{k+1})^{-1}, \tag{11.64}$$

(11.62) and (11.63) can be written as

$$x^{\mathrm{a}}_{k+1} = x^{\mathrm{f}}_{k+1} + K_{k+1}(y_{k+1} - C_{k+1}x^{\mathrm{f}}_{k+1}), \tag{11.65}$$

$$P^{\mathrm{a}}_{k+1} = (I - K_{k+1}C_{k+1})P^{\mathrm{f}}_{k+1}, \tag{11.66}$$

which are identical to (11.32), (11.33).

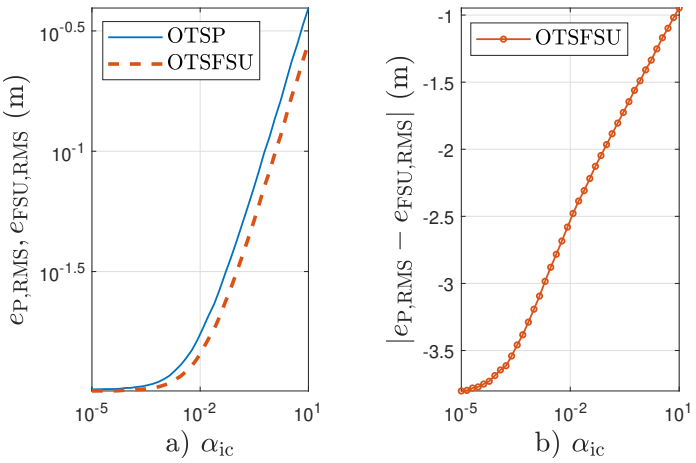

Figure 11.9: Root-mean-square (RMS) position-estimation errors for the optimal two-step predictor (OTSP) and the optimal two-step filter with startup (OTSFSU) versus initial-condition-covariance scaling α_{ic}. a) shows the RMS position-estimation errors for OTSP and OTSFSU, $e_{\mathrm{P,RMS}}$, and $e_{\mathrm{FSU,RMS}}$, respectively. b) shows that OTSFSU is uniformly more accurate than OTSP.

OTSF in the traditional notation is given by

$$\hat{x}_{k+1|k} = A_k\hat{x}_{k|k} + B_k u_k + S_k(C_k P_{k|k-1}C_k^{\mathrm{T}} + R_k)^{-1}(y_k - C_k\hat{x}_{k|k-1}), \tag{11.67}$$

$$\begin{aligned} P_{k+1|k} = {} & A_k P_{k|k}A_k^{\mathrm{T}} - S_k(C_k P_{k|k-1}C_k^{\mathrm{T}} + R_k)^{-1}C_k P_{k|k-1}A_k^{\mathrm{T}} \\ & - A_k P_{k|k-1}C_k^{\mathrm{T}}(C_k P_{k|k-1}C_k^{\mathrm{T}} + R_k)^{-1}S_k^{\mathrm{T}} \\ & - S_k(C_k P_{k|k-1}C_k^{\mathrm{T}} + R_k)^{-1}S_k^{\mathrm{T}} + Q_k, \end{aligned} \tag{11.68}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + P_{k+1|k}C_{k+1}^{\mathrm{T}}(C_{k+1}P_{k+1|k}C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1}(y_{k+1} - C_{k+1}\hat{x}_{k+1|k}), \tag{11.69}$$

$$P_{k+1|k+1} = P_{k+1|k} - P_{k+1|k}C_{k+1}^{\mathrm{T}}(C_{k+1}P_{k+1|k}C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1}C_{k+1}P_{k+1|k}, \tag{11.70}$$

where (11.67), (11.68) are (11.38), (11.39) with additional terms involving S_k, and (11.69), (11.70) are identical to (11.40), (11.41). The OTSF gain (11.64) can be written as

$$K_{k+1}^{\mathrm{f}} = P_{k+1|k}^{\mathrm{f}}C_{k+1}^{\mathrm{T}}(C_{k+1}P_{k+1|k}^{\mathrm{f}}C_{k+1}^{\mathrm{T}} + R_{k+1})^{-1}, \tag{11.71}$$

and thus (11.69) and (11.70) can be written as

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K^{\mathrm{f}}_{k+1}(y_{k+1} - C_{k+1}\hat{x}_{k+1|k}), \tag{11.72}$$

$$P_{k+1|k+1} = (I - K^{\mathrm{f}}_{k+1}C_{k+1})P_{k+1|k}, \tag{11.73}$$

which are identical to (11.43), (11.44).

CHAPTER 12

Conclusions and Future Work

12.1 Conclusions

This dissertation presented a decomposition of the retrospective performance variable, which was used to gain insights into why RCAC and DDRCAC converge to controllers that provide command-following, disturbance and sensor-noise rejection performance. Furthermore, this decomposition provides the basis of rules-of-thumb for the construction of the target model $G_{\mathrm{f}}(\mathbf{q})$ in the SISO case.

Additionally, this dissertation developed and demonstrated DDRCAC as an extension of RCAC from direct adaptive control to a hybrid direct/indirect adaptive control, where RLS with variable-rate forgetting is used for online system identification. For SISO and MIMO systems, the identified model is used to construct the target model, which provides an essential model of the closed-loop dynamics, including NMP zeros. In the presence of sensor noise and actuator magnitude and rate limits, DDRCAC was shown to be effective for plants with a priori unknown NMP zeros.

Using RLS with variable-rate forgetting, DDRCAC was found to provide self-generated persistency, thus facilitating system identification. Furthermore, although closed-loop identification can entail parameter-estimate bias, it was found that, in DDRCAC, identification and control interact so as reduce the effect of bias.

In the MIMO case, RCAC was shown to create and cancel NMP cascade zeros, which

lead to the control diverging. DDRCAC was shown to avoid cancellation of NMP cascade zeros, which are created due to the cascade of a nonsquare system and a controller.

Finally, DDRCAC was demonstrated in various flight-control applications, namely, an aircraft with unknown, time-dependent transition from MP to NMP dynamics, a flexible aircraft, wing flutter, and nonlinear planar missile dynamics. Finally, flight-control examples showed that DDRCAC is effective for both linear and nonlinear applications as either a standalone embedded controller or as a simulation-based offline tuning technique for assessing achievable performance without requiring explicit knowledge of the underlying equations of motion.

12.2 Future Work

Adaptive control of nonsquare systems using DDRCAC revealed that DDRCAC has the ability to avoid the creation and cancellation of NMP cascade zeros. This result has important practical implications and thus the underlying mechanisms for this property warrant deeper investigation, analysis, and proof.

The retrospective performance variable decomposition was derived for MIMO systems, but the numerical examples were confined to SISO systems. Future work is required for numerical verification of this result in the MIMO case. Numerical investigations using MIMO retrospective performance variable decomposition may lead to new insights into MIMO control using RCAC and DDRCAC.

Finally, it was demonstrated through numerical examples that online identification and adaptive control interact synergistically to reduce parameter-estimate bias without the need for probing signals as in dual control. This phenomenon has significant practical implications and thus warrants further exploration.

BIBLIOGRAPHY

[1] Parikh, A., Kamalapurkar, R., and Dixon, W. E., "Integral Concurrent Learning: Adaptive Control with Parameter Convergence using Finite Excitation," *International Journal of Adaptive Control and Signal Processing*, Vol. 33, No. 12, 2019, pp. 1775–1787.

[2] Gaudio, J. E., Annaswamy, A. M., Lavretsky, E., and Bolender, M. A., "Fast Parameter Convergence in Adaptive Flight Control," *AIAA Scitech*, 2020, AIAA 2020-0594.

[3] Ioannou, P. and Fidan, B., *Adaptive Control Tutorial*, SIAM, 2006.

[4] Ioannou, P. and Sun, J., *Robust Adaptive Control*, Dover, 2013.

[5] Tao, G., *Adaptive Control Design and Analysis*, Wiley, 2003.

[6] Ilchmann, A., "Non-Identifier-Based Adaptive Control of Dynamical Systems: A Survey," *IMA Journal of Mathematical Control and Information*, Vol. 8, 1991, pp. 321–366.

[7] Hovakimyan, N., Cao, C., Kharisov, E., Xargay, E., and Gregory, I. M., "L1 Adaptive Control for Safety-Critical Systems," *IEEE Control Systems Magazine*, Vol. 31, No. 5, 2011, pp. 54–104.

[8] Wiese, D. P., Annaswamy, A. M., Muse, J. A., Bolender, M. A., and Lavretsky, E., "Adaptive Output Feedback Based on Closed-Loop Reference Models for Hypersonic Vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 12, 2015, pp. 2429–2440.

[9] Tao, G., "Multivariable adaptive control: A survey," *Automatica*, Vol. 50, No. 11, 2014, pp. 2737–2764.

[10] den Hof, P. V. and Schrama, R. J. P., "Identification and control — Closed-loop Issues," *Automatica*, Vol. 31, No. 12, 1995, pp. 1751–1770.

[11] Gevers, M., "Towards a Joint Design of Identification and Control?" *Progress in Systems and Control Theory*, Vol. 14, 1993, pp. 111—-151.

[12] Hjalmarsson, H., "From Experiment Design to Closed-Loop Control," *Automatica*, Vol. 41, No. 3, 2005, pp. 393–438.

[13] Feldbaum, A. A., "Dual Control Theory," *Avtomatika i Telemekhanika*, Vol. 22, No. 1, 1961, pp. 3–16.

[14] Wittenmark, B., "Adaptive Dual Control Methods: An Overview," *Adaptive Systems in Control and Signal Processing*, Vol. 28, 1995, pp. 67–72.

[15] Filatov, N. M. and Unbehauen, H., *Adaptive Dual Control: Theory and Applications*, Springer, 2004.

[16] Islam, S. A. U. and Bernstein, D. S., "Recursive Least Squares for Real-Time Implementation," *IEEE Control Systems Magazine*, Vol. 39, No. 3, 2019, pp. 82–85.

[17] Bruce, A. L., Goel, A., and Bernstein, D. S., "Convergence and Consistency of Recursive Least Squares with Variable-Rate Forgetting," *Automatica*, Vol. 119, 2020, pp. 109052.

[18] Venugopal, R. and Bernstein, D. S., "Adaptive Disturbance Rejection Using ARMARKOV/Toeplitz Models," *IEEE Trans. Contr. Sys. Tech.*, Vol. 8, 2000, pp. 257–269.

[19] Hoagg, J. B., Santillo, M. A., and Bernstein, D. S., "Discrete-time Adaptive Command Following and Disturbance Rejection with Unknown Exogenous Dynamics," *IEEE Trans. Autom. Contr.*, Vol. 53, 2008, pp. 912–928.

[20] Santillo, M. A. and Bernstein, D. S., "Adaptive Control Based on Retrospective Cost Optimization," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 289–304.

[21] Hoagg, J. B. and Bernstein, D. S., "Retrospective Cost Model Reference Adaptive Control for Nonminimum-Phase Systems," *AIAA J. Guid. Contr. Dyn.*, Vol. 35, 2012, pp. 1767–1786.

[22] Rahman, Y., Xie, A., Hoagg, J. B., and Bernstein, D. S., "A Tutorial and Overview of Retrospective-Cost-Based Adaptive Control," *Proc. Amer. Contr. Conf.*, Boston, MA, July 2016, pp. 3386–3409.

[23] Sumer, E. D., Hoagg, J. B., and Bernstein, D. S., "Broadband Disturbance Rejection Using Retrospective Cost Adaptive Control," *Proc. Dyn. Sys. Contr. Conf.*, Fort Lauderdale, FL, October 2012, pp. 1–10.

[24] Rahman, Y., Xie, A., and Bernstein, D. S., "Retrospective Cost Adaptive Control: Pole Placement, Frequency Response, and Connections with LQG Control," *IEEE Control Systems Magazine*, Vol. 37, October 2017, pp. 28–69.

[25] Islam, S. A. U., Xie, A., and Bernstein, D. S., "Adaptive Control of Systems with Unknown Nonminimum-Phase Zeros Using Cancellation-Based Pseudo-identification," *Proceedings of the American Control Conference*, Philadelphia, PA, 2019, pp. 441–446.

[26] Willems, J. C., Rapisarda, P., Markovsky, I., and De Moor, B. L. M., " A Note on Persistency of Excitation ," *Systems and Control Letters*, Vol. 77, No. 4, 2005, pp. 325–329.

[27] Goel, A., Bruce, A., and Bernstein, D. S., "Recursive Least Squares with Variable-Direction Forgetting: Compensating for the Loss of Persistency," *IEEE Control Systems Magazine*, Vol. 40, August 2020, pp. 80–102.

[28] Chowdhary, G., Mühlegg, M., and Johnson, E. N., "Exponential Parameter and Tracking Error Convergence Guarantees for Adaptive Controllers without Persistency of Excitation," *International Journal of Control*, Vol. 87, No. 8, 2014, pp. 1583–1603.

[29] Forssell, U. and Ljung, L., "Closed-Loop Identification Revisited," *Automatica*, Vol. 35, No. 7, 1999, pp. 1215–1241.

[30] Aljanaideh, K. F. and Bernstein, D. S., "Closed-Loop Identification of Unstable Systems Using Noncausal FIR Models," *International Journal of Control*, Vol. 90, No. 2, 2017, pp. 168–185.

[31] Sobolic, F., Aljanaideh, K. F., and Bernstein, D. S., "A Numerical Investigation of Direct and Indirect Closed-Loop Architectures for Estimating Nonminimum-Phase Zeros," *International Journal of Control*, Vol. 93, No. 6, 2020, pp. 1251–1265.

[32] Gilson, M. and den Hof, P. V., "Instrumental Variable Methods for Closed-loop System Identification," *Automatica*, Vol. 41, No. 7, 2005, pp. 241–249.

[33] Islam, S. A. U., Nguyen, T. W., Kolmanovsky, I., and Bernstein, D. S., "Data-Driven Retrospective Cost Adaptive Control for Flight Control Applications," *AIAA Journal of Guidance, Control, and Dynamics*, 2021, to appear.

[34] Vardulakis, A. I. G., "Zero Placement and the 'Squaring Down' Problem: A Polynomial Matrix Approach," *International Journal of Control*, Vol. 31, No. 5, 1980, pp. 821–832.

[35] Davison, E., "Some Properties of Minimum Phase Systems and "Squared-down" Systems," *IEEE Transactions on Automatic Control*, Vol. 28, No. 2, 1983, pp. 221–222.

[36] Saberi, A. and Sannuti, P., "Squaring Down of Non-Strictly Proper Systems," *International Journal of Control*, Vol. 51, No. 3, 1990, pp. 621–629.

[37] Leventides, J. and Karcanias, N., "Structured Squaring Down and Zero Assignment," *International Journal of Control*, Vol. 81, No. 2, 2008, pp. 294–306.

[38] Oara, C., Flutur, C., and Jungers, M., "Squaring Down with Zeros Cancellation in Generalized Systems," *Systems and Control Letters*, Vol. 92, 2016, pp. 5 – 13.

[39] Lavretsky, E., "Robust and Adaptive Output Feedback Control for Non-Minimum Phase Systems with Arbitrary Relative Degree," *AIAA Scitech*, 2017, AIAA 2017-1490.

[40] Bernstein, D. S., *Scalar, Vector, and Matrix Mathematics: Theory, Fact, and Formulas*, Princeton University Press, revised and expanded ed., 2018.

[41] Albert, A. and Sittler, R. W., "A Method for Computing Least Squares Estimators that Keep Up with the Data," *SIAM J. Contr.*, Vol. 3, No. 3, 1965, pp. 384–417.

[42] Astrom, K. and Wittenmark, B., *Adaptive Control: Second Edition*, Dover, 2013.

[43] M. E. Salgado, G. C. Goodwin and R. H. Middleton, "Modified Least Squares Algorithm Incorporating Exponential Resetting and Forgetting," *Int. J. Contr.*, Vol. 47, No. 2, 1988, pp. 477–491.

[44] S. Dasgupta and Y.-F. Huang, "Asymptotically Convergent Modified Recursive Least-Squares with Data-Dependent Updating and Forgetting Factor for Systems with Bounded Noise," *IEEE Trans. Inf. Theory*, Vol. 33, No. 3, 1987, pp. 383–392.

[45] P. Stoica and P. Ahgren, "Exact Initialization of the Recursive Least-Squares Algorithm," *Int. J. Adapt. Contr. Sig. Proc.*, Vol. 16, 2002, pp. 219–230.

[46] Ljung, L. and Soderstrom, T., *Theory and Practice of Recursive Identification*, MIT Press, 1985.

[47] Johnson, C., *Lectures on Adaptive Parameter Estimation*, Prentice-Hall, 1988.

[48] Chen, T., Tapn, T., and Francis, B., *Optimal Sampled-Data Control Systems*, Springer, 1995.

[49] Aljanaideh, K. F. and Bernstein, D. S., "Initial Conditions in Time- and Frequency-Domain System Identification: Implications of the Shift Operator Versus the Z and Discrete Fourier Transforms," *IEEE Control Systems Magazine*, Vol. 38, No. 2, 2018, pp. 80–93.

[50] Middleton, R. and Goodwin, G., *Digital Control and Estimation: A Unified Approach*, Prentice Hall, 1990.

[51] Islam, S. A. U., Xie, A., and Bernstein, D. S., "Closed-Loop Performance versus Target-Model Matching in Retrospective Cost Adaptive Control," *Proceedings of the American Control Conference*, New Orleans, LA, 2021, submitted.

[52] Goel, A. and Bernstein, D. S., "Gradient-, Ensemble-, and Adjoint-Free Data-Driven Parameter Estimation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1743–1754.

[53] Islam, S. A. U., Ratnam, E. L., Goel, A., and Bernstein, D. S., "Phasor-Based Adaptive Control of a Test-Feeder Distribution Network: Application of Retrospective Cost Adaptive Control to the IEEE 13-Node Test Feeder," *IEEE Control Systems Magazine*, Vol. 39, No. 4, 2019, pp. 56–74.

[54] Goel, A., Islam, S. A. U., and Bernstein, D. S., "Adaptive Control of MIMO Systems Using Sparsely Parameterized Controllers," *Proc. Amer. Contr. Conf.*, 2020, pp. 5340–5345.

[55] "Pecan Street," http://www.pecanstreet.org/, Accessed: 2018-03-14.

[56] Hansen, P. C. and O'Leary, D. P., "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems," *SIAM Journal on Scientific Computing*, Vol. 14, No. 6, 1993, pp. 1487–1503.

[57] Golub, G. H., Hansen, P. C., and O'Leary, D. P., "Tikhonov Regularization and Total Least Squares," *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, 1999, pp. 185–194.

[58] Cucker, F. and Smale, S., "Best Choices for Regularization Parameters in Learning Theory: On the Bias-Variance Problem," *Foundations of Computational Mathematics*, Vol. 2, 2002, pp. 413–428.

[59] Lu, S., Pereverzev, S. V., and Tautenhahn, U., "Regularized Total Least Squares: Computational Aspects and Error Bounds," *SIAM Journal on Matrix Analysis and Applications*, Vol. 31, No. 3, 2010, pp. 918–941.

[60] Rahman, Y., Aljanaideh, K., Sumer, E. D., and Bernstein, D. S., "Adaptive Control of Aircraft Lateral Motion with an Unknown Transition to Nonminimum-Phase Dynamics," *Proceedings of the American Control Conference*, Portland, OR, June 2014, pp. 2359–2364.

[61] Ansari, A. and Bernstein, D. S., "Adaptive Control of an Aircraft with Uncertain Nonminimum-Phase Dynamics," *Proceedings of the American Control Conference*, Chicago, IL, July 2015, pp. 844–849.

[62] Islam, S. A. U., Bruce, A. L., Nguyen, T. W., Kolmanovsky, I., and Bernstein, D. S., "Adaptive Flight Control with Unknown Time-Varying Unstable Zero Dynamics," *AIAA Scitech*, 2020, AIAA 2020-0839.

[63] Waszak, M. R. and Schmidt, D. K., "Flight Dynamics of Aeroelastic Vehicles," *Journal of Aircraft*, Vol. 25, No. 6, 1988, pp. 563–571.

[64] Waszak, M. R., "Modeling the Benchmark Active Control Technology Wind-Tunnel Model for Application to Flutter Suppression," *21st AIAA Atmospheric Flight Mechanics Conference*, San Diego, CA, 1996, pp. 1–13, AIAA 96-3437.

[65] Waszak, M. and Center, L. R., *Modeling the Benchmark Active Control Technology Wind-tunnel Model for Active Control Design Applications*, NASA technical paper, NASA Langley Research Center, 1998, NASA/TP-1998-206270.

[66] Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain Scheduling for H-infinity Controllers: A Flight Control Example," *IEEE Transactions on Control System and Technology*, Vol. 1, No. 2, June 1993, pp. 69–79.

[67] Mracek, C. P. and Cloutier, J. R., "Full Envelope Missile Longitudinal Autopilot Design Using the State-Dependent Riccati Equation Method," *Proceedings of the Guidance, Navigation, and Control Conference*, 1997, pp. 1697–1705, AIAA-97-3767.

[68] Bennani, S., Willemsen, D., and Scherer, C., "Robust LPV Control with Bounded Parameter Rates," *Proceedings of the Guidance, Navigation, and Control Conference*, 1997, pp. 1080–1089, AIAA-97-3641.

[69] Islam, S. A. U., Goel, A., and Bernstein, D. S., "Real-Time Implementation of the Optimal Predictor and Optimal Filter: Accuracy Versus Latency," *IEEE Control Systems Magazine*, Vol. 40, No. 2, 2020, pp. 84–91.

[70] Jazwinski, A., *Stochastic Processes and Filtering Theory*, Elsevier Science, 1970.

[71] Anderson, B. and Moore, J., *Optimal Filtering*, Prentice-Hall, 1979.

[72] Crassidis, J. and Junkins, J., *Optimal Estimation of Dynamic Systems*, CRC Press, 2004.

[73] Simon, D., *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*, Wiley, 2006.

[74] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, Vol. 82, No. 1, 1960, pp. 35–45.

[75] Teixeira, B., "Kalman Filters," *IEEE Control Systems*, Vol. 28, No. 2, April 2008, pp. 16–18.