Supervisory Adaptive Control Revisited: Linear-like Convolution Bounds

by

Craig Lalumiere

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Classical feedback control for LTI systems enjoys many desirable properties including exponential stability, a bounded noise-gain, and tolerance to a degree of unmodeled dynamics. However, an accurate model for the system must be known. The field of adaptive control aims to allow one to control a system with a great deal of parametric uncertainty, but most such controllers do not exhibit those nice properties of an LTI system, and may not tolerate a time-varying plant. In this thesis, it is shown that an adaptive controller constructed via the machinery of Supervisory Control yields a closed-loop system which is exponentially stable, and where the effects of the exogenous inputs are bounded above by a linear convolution - this is a new result in the Supervisory Control literature. The consequences of this are that the system enjoys linear-like properties: it has a bounded noise-gain, is robust to a degree of unmodeled dynamics, and is tolerant of a degree of time-varying plant parameters.

This is demonstrated in two cases: the first is the typical application of Supervisory Control - an integral control law is used to achieve step tracking in the presence of a constant disturbance. It is shown that the tracking error exponentially goes to zero when the disturbance is constant, and is bounded above by a linear convolution when it is not. The second case is a new application of Supervisory Control: it is shown that for a minimum phase plant, the d-step-ahead control law may be used to achieve asymptotic tracking of an *arbitrary* bounded reference signal. In addition to the convolution bound, a crisp bound is found on the 1-norm of the tracking error when a disturbance is absent.

Acknowledgements

I would like to express my sincere thanks to my supervisor, Professor Daniel E. Miller, for providing me both academic and professional guidance, for directing my research, pointing me in interesting directions, helping me solve problems, and for his many hours spent proof reading this work.

I would also like to show my appreciation to my readers Professor Andrew Heunis and Professor Christopher Nielsen for providing their valuable feedback and comments.

Dedication

This work is dedicated to my parents, whose unconditional love and support guided me through my personal troubles encountered during graduate school. Without them, this thesis would not have been possible.

Table of Contents

List of Figures

Chapter 1

Introduction

Adaptive control is an approach used to control systems with uncertain and possibly time-varying parameters. Instead of being a classical, fixed controller, designed using a priori information about the plant dynamics, an adaptive controller uses a posteriori information gathered during runtime to continuously tweak itself in the hopes of improving its own performance. This allows the controller to adapt to a wide range of operating conditions, creating the potential for a system with an immense degree of robustness to parametric uncertainty. Adaptive controllers are often separated into two categories: 'indirect' adaptive controllers and 'direct' adaptive controllers. The former works on the principle of real-time system identification: some sort of model fitting algorithm is used to continuously update an estimated model of the plant, and each time the model changes, some classical feedback control strategy is used to create a new control law. The latter is any adaptive control scheme that does not use the intermediate step of modeling the plant.

This thesis is concerned with indirect adaptive control: a parameter estimator paired with a 'certainty equivalence' [1] control law. In many instances the control law is linear, but the effect of the parameter estimator in the feedback loop results in a nonlinear system, and may exhibit chaotic behavior [27]. Indeed, adaptive systems are often very difficult to analyze. Adaptive controllers were pursued in the 1960's in an attempt to cope with the complicated dynamics of the supersonic X-15 experimental aircraft. They were used with some success, but their nature was not fully understood, and the adaptive controller was in part responsible for the crash of the X-15-3 [6]. It wasn't until the 1980's that stability of

[1] The certainty equivalence principle is to take the parameter estimate at face value and design a control law as though it were the true parameters, in contrast to other control laws which may take the estimate uncertainty into account [28].

adaptive systems began to be proven, see e.g. [9, 11, 39, 44, 45]. These adaptive controllers were shown to be highly robust to a large degree of parametric uncertainty, but they were often intolerant of time-varying parameters, disturbances or unmodeled dynamics, e.g. see [7], thus they may be entirely non-robust in another sense.

Subsequent research over the following two decades used various innovations to attempt to alleviate some of these shortcomings to a degree. Some of these changes include the use of signal normalization, deadzones, $\sigma-$modification, and using projection onto a convex set of admissible parameters, e.g. see [29, 8, 51, 23, 21, 55, 56, 43, 54, 53, 26]. However, in general these redesigned controllers still only provide asymptotic stability and not exponential stability; furthermore, typically a bounded gain on the disturbance is not proven, although some of the proposed controllers, especially those which use projection, provide a bounded-disturbance bounded-state property, as well as tolerance of some degree of unmodeled dynamics and/or time-variations.

In the 1990s a new technique labeled Supervisory Control was shown in [40, 41] to produce exponential stability, a bounded gain on the noise/disturbance, step tracking, and robustness to a degree of unmodeled dynamics. While tolerance to time variations has not been proven for this controller, in [52] a modified version using hysteresis is shown to have some degree of tolerance.

1.1 Objective

Classical LTI feedback control enjoys a bounded gain on the disturbance and tolerance to a degree of unmodeled dynamics, but is limited to only plants with fixed dynamics known a priori. In contrast, most adaptive controllers can stabilize a plant with uncertain dynamics, but may fail in the presence of a disturbance or unmodeled dynamics or when the plant's dynamics change over time. The proposed solution of this thesis is to devise an adaptive controller with linear-like closed-loop behavior. To understand what this means, consider an LTI system with state vector $x(t)$, initial condition $x(t_0)$ at time $t_0 \in \mathbf{Z}$, exogenous input $w(t)$ and state transition matrix Φ satisfying $\|\Phi(t)\| \leq \gamma\lambda^t$ for some $\gamma \geq 1$ and $\lambda \in (0, 1)$. Then the state's evolution over time is described by the convolution

$$x(t) = \Phi(t - t_0)x(t_0) + \sum_{i=t_0}^{t-1} \Phi(t - i - 1)w(i), \quad t \geq t_0.$$

An adaptive system is nonlinear, so such a result will not exist. Instead we aim to show that the magnitude of the state vector is upper bounded by a similar convolution. In fact, it is sufficient if only a *part* of the system's overall state is bounded as such. If one can break the closed-loop system state up into two components:

$$x(t) = \begin{bmatrix} \bar{x}(t) \\ \tilde{x}(t) \end{bmatrix},$$

with the plant's output $y(t)$ and control input $u(t)$ elements of $\bar{x}(t)$, while $\tilde{x}(t)$ contains other variables such as parameter estimates, and then show that there exists a $\gamma \geq 1$ and $\lambda \in (0, 1)$ such that the bound

$$\|\bar{x}(t)\| \leq \gamma\lambda^{t-t_0}\|\bar{x}(t_0)\| + \sum_{i=t_0}^{t-1} \lambda^{t-i-1}|w(i)|, \quad t \geq t_0$$

holds uniformly for all admissible initial conditions $x(t_0)$, then one can show that the system has various desirable properties. An obvious consequence of the convolution bound is that the signals y and u are exponentially stable, and the system has a bounded gain on the disturbance. What is less obvious is that using a modular technique analysed in [49, 37] yields robustness to a degree of unmodeled dynamics and slowly time-varying parameters.

Such a result has been seen once before using the *Projection Algorithm* parameter estimator. This is shown in [31, 35] where it is paired with a pole placement controller to achieve step tracking, and in [32, 36] where it is paired with the d-step-ahead controller to track an arbitrary reference signal. However, in the d-step-ahead setup, some stringent assumptions are required, namely that the delay of the plant must be known a priori, as does the sign of the high frequency gain. However, this last issue is partially mitigated in [46, 47, 48, 50] by using multiple convex regions and multiple estimators, though only for the first order case.

The objective of this thesis is to design an adaptive controller which provides the same linear-like property of [35, 36, 38], while relaxing some of the assumptions; namely, we remove the need to know the plant delay and the sign of the high frequency gain. To achieve this, we apply Supervisory Control in discrete-time to both the classical d-step-ahead adaptive tracking problem (this is, in itself, new) and to the step tracking problem using an integral pole placement controller. In both cases a convolution bound is proven, demonstrating that this forms a truly robust closed-loop adaptive system: robustness to large scale initial uncertainty in the plant model and to bounded disturbances, and courtesy of [49, 37], robustness to a degree of unmodeled dynamics and time-varying parameters.

1.2 Notation

We use standard notation throughout this thesis. We use the Euclidean 2-norm for vectors and the corresponding induced norm for matrices, and denote the norm of a vector or matrix by $\|\cdot\|$. For an arbitrary subset $\mathcal{X}$ of $\mathbf{R}^n$, we define its norm to be

$$\|\mathcal{X}\| := \sup_{x\in\mathcal{X}} \|x\|.$$

We let $\{a,\ldots,b\}$ denote the integer interval $\{x \in \mathbf{Z} : a \leq x \leq b\}$. We let l_∞ denote the set of real-valued bounded sequences. We let $\ker\{\cdot\}$ denote the kernel (nullspace) of a matrix, and $\mathrm{vec}(\cdot)$ denotes the vectorization operator. For an arbitrary signal $a : \mathbf{Z} \to \mathbf{R}$, we let $\Delta a(t)$ denote $a(t) - a(t-1)$, and we let q^{-1} denote the unit backwards shift operator: $q^{-1}a(t) = a(t-1)$. For an LTI system $H(z)$ with right-sided impulse response $h(t)$, and an arbitrary signal $a : \{t \in \mathbf{Z} : t \geq t_0\} \to \mathbf{R}$, we define their convolution product as

$$\{h \circ a\}(t) := \sum_{j=0}^{t-t_0} h(j)a(t-j).$$

Lastly, it is assumed that

$$\sum_{i=i_1}^{i_2} (\cdot) = 0, \quad i_2 < i_1.$$

Chapter 2

A Discussion of Parameter Estimation Techniques

In this chapter we discuss several parameter estimators which may be used to form an indirect adaptive control system, and consider which may be candidates for our desired adaptive system with 'linear-like' properties. The goal for these adaptive controllers is to control a discrete-time, LTI plant of the form

$$y(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{=:\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{=:\theta^*} + w(t), \quad t \geq t_0.$$

Here $y(t) \in \mathbf{R}$ is the plant's output, $u(t) \in \mathbf{R}$ is the control input, and $w \in l_\infty$ the disturbance (or noise) input. The controller's goal is to make the output y asymptotically track a reference signal r from a suitable class. The system's initial condition $\phi(t_0)$ is assumed to be known (since it consists of measurable input-output data) but the parameter vector θ^* is unknown, but lies within a known, compact set $\mathcal{S}$. The model corresponds to the transfer function

$$P(z^{-1}) = \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-m}}{1 - a_1^* z^{-1} - \ldots - a_n^* z^{-n}}.$$

If θ^* were known, then classical feedback design techniques could be used to stabilize the system. Instead, the indirect adaptive controller uses its estimator to generate an estimate $\hat{\theta}$ of the true parameters θ^*. This is then used with the 'certainty equivalence' control method, which is to design a control law which would stabilize the system if the $\hat{\theta}$ were the true parameters. Thus, the control law takes the form [1]

$$u(t) = f\left(\hat{\theta}(t), y(t), \phi(t-1), r\right). \tag{2.1}$$

As discussed in Chapter 1, we hope to create an adaptive controller where the closed-loop behavior obeys a linear-like bound: specifically, where the total system state is split into two parts:

$$x(t) = \begin{bmatrix} \bar{x}(t) \\ \tilde{x}(t) \end{bmatrix},$$

and there exists a $\gamma \geq 1$ and $\lambda \in (0,1)$ such that the bound

$$\|\bar{x}(t)\| \leq \gamma \lambda^{t-t_0} \|\bar{x}(t_0)\| + \sum_{i=t_0}^{t-1} \lambda^{t-i-1} \left(|w(i)| + |r(i)|\right), \quad t \geq t_0 \tag{2.2}$$

holds uniformly for any initial starting time $t_0 \in \mathbf{Z}$, initial condition $\begin{bmatrix} \bar{x}(t_0) \\ \tilde{x}(t_0) \end{bmatrix}$, and $\theta^* \in \mathcal{S}$. The motivation for this is that one could then apply the results of [49] to show that the closed-loop system retains exponential stability even in the presence of unmodeled dynamics and slowly time-varying parameters. The purpose of this chapter is to informally consider several well known parameter estimates, draw parallels between them, and provide justification as to why we believe they may or may not be candidates for such a result.

2.1 The Projection Algorithm

This is a recursive estimator which is initialized with some initial guess $\hat{\theta}(t_0)$. Then at every time-step, a new observation of $y(t+1)$ is used in conjunction with the previous estimate $\hat{\theta}(t)$ to produce the new estimate $\hat{\theta}(t+1)$. We know that

$$y(t+1) = \phi(t)^\top \theta^* + w(t),$$

[1] We use r here rather than $r(t)$ since in some cases, the controller needs access to r a short distance into the future.

but the disturbance $w(t)$ is unknown, so we use the equation

$$y(t+1) = \phi(t)^\top \hat{\theta}(t+1)$$

as the basis for selecting $\hat{\theta}(t+1)$. Notice that if $\phi(t) = 0$, this has no solution; otherwise, it has many solutions. Since $\hat{\theta}(t) \in \mathbf{R}^{n+m}$, in the case where $\phi(t) \neq 0$, the set of solutions to this equation represents a $n+m-1$ dimensional hyperplane. With the assumption that $\hat{\theta}(t)$ has been well chosen and is 'good' in some way, we choose $\hat{\theta}(t+1)$ to be the element on this hyperplane which is nearest to $\hat{\theta}(t)$. This is achieved by projecting the point $\hat{\theta}(t)$ onto the hyperplane. Thus, the 'ideal'[2] Projection Algorithm parameter estimator is described by the recursive-update equation:

$$\hat{\theta}(t+1) = \begin{cases} \hat{\theta}(t) & \text{if } \phi(t) = 0 \\ \hat{\theta}(t) + \frac{\phi(t)}{\phi(t)^\top \phi(t)} \Big(y(t+1) - \phi(t)^\top \hat{\theta}(t) \Big) & \text{otherwise.} \end{cases}$$

When paired with a control law of the form (2.1) to create an adaptive controller, it produces a nonlinear system; its state is described by $\phi(t)$ and $\hat{\theta}(t)$, but instead we will define the **parameter estimation error** as

$$\tilde{\theta}(t) := \hat{\theta}(t) - \theta^*,$$

and choose to define the state vector as

$$x(t) = \begin{bmatrix} \phi(t) \\ \tilde{\theta}(t) \end{bmatrix}.$$

If the exogenous inputs (disturbance and reference) were zero, one would hope that the state x would be exponentially stable, thus $y \to 0$, $u \to 0$ and $\hat{\theta} \to \theta^*$. However, this is not the case: it is well known that the parameter estimate will generally not converge to the true parameters.

A slight variation of this estimator is used in [31, 35] and [32, 36] for the pole placement and d-step-ahead cases respectively. In that setup, the set $\mathcal{S}$ must be convex, and the parameter estimate is projected onto the set at each step to ensure that $\hat{\theta}(t) \in \mathcal{S}$. Then, by splitting the state vector into two components:

$$\bar{x}(t) := \phi(t), \quad \tilde{x}(t) := \tilde{\theta}(t),$$

[2] In much of the literature, this algorithm is modified to avoid having to treat the case $\phi(t) = 0$ separately, but doing so makes it lose the nice property we are interested in [38].

then $\bar{x}$ does indeed satisfy a bound of the sort (2.2) which holds uniformly for any $t_0 \in \mathbf{Z}$, initial condition $\phi(t_0)$, $\hat{\theta}(t_0) \in \mathcal{S}$, and $\theta^* \in \mathcal{S}$. This in turn is leveraged to show that the adaptive system enjoys various robustness properties. Additionally, in the pole placement approach of [48], the constraint that $\mathcal{S}$ be convex is relaxed through the use of multiple estimators.

2.2 Ordinary Least-Squares

The first in a series of optimization-based estimators, the 'ordinary least-squares' estimator is where $\hat{\theta}(t)$ is chosen to minimize the overall squared prediction error over all past observations:

$$\hat{\theta}(t) = \arg\min_{\theta} \sum_{i=t_0}^{t-1} (y(i+1) - \phi(i)^\top \theta)^2, \quad t \geq t_0. \tag{2.3}$$

A very important term related to this estimator is the matrix

$$R(t) := \sum_{i=t_0}^{t-1} \phi(i)\phi(i)^\top, \quad t \geq t_0.$$

At a given time $t \geq t_0$, if $R(t)$ is not full-rank, then the optimization routine described above has many solutions. If $R(t)$ is full-rank, then there is a unique solution. Also, in the latter case, the estimator can be equivalently described by a recursive-update equation. If $\bar{t}_0 \geq t_0$ is such that $R(\bar{t}_0)$ is full rank, then $R(t)$ is full rank for all $t \geq \bar{t}_0$. In this case, we introduce what is referred to as the 'covariance matrix', $P(t) := R(t)^{-1}, t \geq \bar{t}_0$. Then, with initial condition $\hat{\theta}(\bar{t}_0)$ and $P(\bar{t}_0) := R(\bar{t}_0)^{-1}$, the estimator above is equivalent to

$$\begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) + \frac{P(t)\phi(t)}{1 + \phi(t)^\top P(t)\phi(t)} \left(y(t+1) - \phi(t)^\top \hat{\theta}(t) \right), \\ P(t+1) &= P(t) - \frac{P(t)\phi(t)\phi(t)^\top P(t)}{1 + \phi(t)^\top P(t)\phi(t)}, \quad t \geq \bar{t}_0. \end{aligned}$$

$P(t) \in \mathbf{R}^{(n+m)\times(n+m)}$ is known as the covariance matrix since it represents the uncertainty in the estimate. Observe that $\|P(t)\|$ is non-increasing: as more data is observed, the covariance decreases, hence the estimator's confidence in its estimate increases. To avoid the added complexity in the case where $R(t)$ is singular, let us assume that the system is initialized at a given time $\bar{t}_0$ with $R(\bar{t}_0)$ positive definite. When paired with a control law of the form (2.1) to create an adaptive controller, the total system state is described by $\phi(t)$, $\hat{\theta}(t)$ and $P(t)$; however, it will be more useful to define it as

$$x(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \\ \tilde{\theta}(t) \end{bmatrix}.$$

Now let us consider if all or part of the state vector might be exponentially stable. As with the Projection Algorithm, it is well known that $\hat{\theta}$ generally does not converge to θ^*, so it is not true that the entire state x is exponentially stable. Next let us consider only the partial state

$$\bar{x}(t) := \phi(t).$$

Consider the simple case where the disturbance and reference are both zero: then the bound (2.2) simplifies to the standard definition [13] for global exponential stability for the partial state $\bar{x}$: there exists a $\gamma \geq 1$ and $\lambda \in (0,1)$ such that the bound

$$\|\phi(t)\| \leq \gamma \lambda^{t-\bar{t}_0} \|\phi(\bar{t}_0)\|, \quad t \geq \bar{t}_0 \tag{2.4}$$

holds for any $\bar{t}_0 \in \mathbf{Z}$ and initial condition $\phi(\bar{t}_0)$, $\hat{\theta}(\bar{t}_0)$ and $R(\bar{t}_0)$. However, it is easy to show that this is not the case. Let $\hat{\theta}(\bar{t}_0)$ be a poor initial guess such that the control law is initially destabilizing, thus in the short term, y is growing exponentially. Now suppose that $R(\bar{t}_0)$ is some small, positive definite matrix. Then we expect the estimator to rapidly 'learn' and the parameter estimate will improve, the control law will become stabilizing, and ϕ will converge to zero relatively quickly.

Next observe that as $P(t) \to 0$, the update in $\hat{\theta}$ becomes zero. In this extreme case, the estimator is certain that it has found the correct parameters, so it ignores any future information - the estimator 'turns off'. So now suppose that the system is initialized with the same $\hat{\theta}(\bar{t}_0)$ as before, but $R(\bar{t}_0)$ is very large (hence $P(\bar{t}_0)$ is very small). In this case, the estimator will be very hesitant to change $\hat{\theta}$ away from $\hat{\theta}(\bar{t}_0)$, so the control law will remain destabilizing for a much longer time. The parameter estimate will eventually improve and ϕ will asymptotically approach zero, but $\|\phi\|_\infty$ will be much greater than in the previous example. This shows that the upper bound of ϕ generally scales with the initial condition $R(t_0)$, which suggests that (2.4) will not hold uniformly for all positive definite $R(t_0)$. A formal proof is not provided as this is only of peripheral interest.

The only remaining option would be if the partial state

$$\bar{x}(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \end{bmatrix}$$

satisfied the bound

$$\left\| \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \end{bmatrix} \right\| \leq \gamma \lambda^{t-\bar{t}_0} \left\| \begin{bmatrix} \phi(\bar{t}_0) \\ \mathrm{vec}(R(\bar{t}_0)) \end{bmatrix} \right\| \tag{2.5}$$

uniformly for all initial conditions. But since R is non-decreasing, this is clearly not the case. Therefore an indirect adaptive controller created using the ordinary least-squares estimator does not exhibit the linear-like property that we are pursuing, which agrees with the fact that it is well known that such a controller is intolerant of plants with time-varying parameters due to the phenomenon of the estimator 'turning off' as $R(t)$ grows. This shortcoming is what motivates the following two estimators, which are similar to the ordinary least-squares estimator, but do not tend to 'turn off', so they may be able to adaptively control systems whose parameters change slowly over time.

2.3 Kalman Filter

One way to handle slowly time-varying parameters is to model them as a stochastic process. Consider a Gaussian random vector ξ with constant covariance matrix Q. Now model the plant's parameter vector as a Gauss-Markov process with the dynamics

$$\theta^*(t+1) = \theta^*(t) + \xi(t).$$

Suppose also that the disturbance w is a Gaussian random signal with constant variance σ. Now, with some initial guess $\hat{\theta}(t_0)$ and initial parameter covariance matrix $P(t_0)$, a Kalman Filter may be used to estimate the parameter vector:

$$\begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) + \frac{P(t)\phi(t)}{\sigma + \phi(t)^\top P(t)\phi(t)} \left(y(t+1) - \phi(t)^\top \hat{\theta}(t) \right), \\ P(t+1) &= P(t) - \frac{P(t)\phi(t)\phi(t)^\top P(t)}{\sigma + \phi(t)^\top P(t)\phi(t)} + Q, \quad t \geq t_0. \end{aligned}$$

Notice the similarities between the Kalman filter equation and the recursive form of the ordinary least-squares estimator. The important distinction is the additive Q term, which has the effect of injecting 'uncertainty' into the estimator. This means that in the absence of new observations (i.e., $\phi(t) = 0$), the estimator begins to lose confidence in the accuracy of its estimate. This ensures that $P(t)$ does not asymptotically decrease to zero, so the estimator does not 'turn off', and it shall be able to estimate slowly time-varying parameters. When paired with a control law of the form (2.1) to create an adaptive controller, and with $R(t) := P(t)^{-1}$, it too can have its state represented by

$$x(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \\ \tilde{\theta}(t) \end{bmatrix}.$$

In simulation, this estimator is quite effective at adaptively stabilizing systems with slowly time-varying parameters, but let us determine if it may be exponentially stable. For the very same reasons as the ordinary least-squares estimator, the full state $x(t)$ and partial state $\bar{x}(t) = \phi(t)$ are not exponentially stable, so consider the partial state:

$$\bar{x}(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \end{bmatrix}.$$

Suppose that on some interval, $\phi(t) = 0$. Then the covariance update simplifies to $P(t+1) = P(t) + Q$. Then $\|P(t)\|$ is increasing linearly over time, so $\|R(t)\|$ is decaying like $\frac{1}{t}$. This is certainly an improvement over the previous estimator, but $\frac{1}{t}$ is slower than any exponential, so $\bar{x}(t)$ is not exponentially stable, hence the system cannot satisfy the desired exponential bound. This is not to say that the estimator is bad, as its simulated performance is very good. However, it may be very difficult to prove any meaningful bounds on its performance, and it might be only locally stable.

2.4 Weighted Least Squares

Another often-used technique to estimate slowly time-varying parameters is to modify the ordinary least-squares cost function by placing greater importance on more recent observations. With forgetting factor $\lambda \in (0, 1)$, the 'weighted least-squares' estimator is

$$\hat{\theta}(t) = \arg\min_{\theta} \sum_{i=t_0}^{t-1} \lambda^{t-i-1} \left(y(i+1) - \phi(i)^\top \theta \right)^2, \quad t \geq t_0.$$

Like the ordinary least-squares estimator, this can be written in recursive form. Define $R(t)$ as

$$R(t) := \sum_{i=t_0}^{t-1} \lambda^{t-i-1} \phi(i)\phi(i)^\top, \quad t \geq t_0.$$

If $\bar{t}_0 \geq t_0$ is such that $R(\bar{t}_0)$ is full rank, then $R(t)$ is full rank for all $t \geq \bar{t}_0$. Now let $P(t) := R(t)^{-1}, t \geq \bar{t}_0$. Then, with initial condition $\hat{\theta}(\bar{t}_0)$ and $P(\bar{t}_0) := R(\bar{t}_0)^{-1}$, the estimator above is equivalent to

$$\begin{aligned}\hat{\theta}(t+1) &= \hat{\theta}(t) + \frac{P(t)\phi(t)}{\lambda + \phi(t)^\top P(t)\phi(t)}\left(y(t+1) - \phi(t)^\top \hat{\theta}(t)\right), \\ P(t+1) &= \frac{1}{\lambda}\left(P(t) - \frac{P(t)\phi(t)\phi(t)^\top P(t)}{\lambda + \phi(t)^\top P(t)\phi(t)}\right), \quad t \geq \bar{t}_0.\end{aligned}$$

Alternatively,

$$\begin{aligned}\hat{\theta}(t+1) &= \hat{\theta}(t) + \frac{R(t)^{-1}\phi(t)}{\lambda + \phi(t)^\top R(t)^{-1}\phi(t)}\left(y(t+1) - \phi(t)^\top \hat{\theta}(t)\right), \\ R(t+1) &= \lambda R(t) + \phi(t)\phi(t)^\top, \quad t \geq \bar{t}_0.\end{aligned}$$

Notice again the similarities between these equations and of the preceding two estimators. The key difference here is that in the case where $\phi(t) = 0$, the covariance matrix P grows exponentially. This conceptually has the same effect as the $+Q$ term in the Kalman filter, except that here the growth is faster. By pairing this with a control law of the form (2.1) to create an adaptive controller, its state may be represented by

$$x(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \\ \tilde{\theta}(t) \end{bmatrix}.$$

Like the Kalman filter, this works quite well in simulation at adaptively stabilizing systems with slowly time-varying parameters, but let us determine if it may be exponentially stable. As before, the full state $x(t)$ and partial state $\bar{x}(t) = \phi(t)$ are not exponentially stable, so consider the partial state:

$$\bar{x}(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}(R(t)) \end{bmatrix}.$$

Suppose that on some interval, $\phi(t) = 0$. Then the update in R simplifies to $R(t+1) = \lambda R(t)$. This is exponential decay, exactly as we would hope. Thus, it *might* be that an indirect adaptive controller created using the weighted least-squares estimator obeys a convolution bound such as (2.2). However, it does not appear that this has ever been proven, nor any sort of stability proof at all when a disturbance is present.

2.5 Supervisory Control

The Supervisory Control method was first proposed in [40, 41] as a continuous-time adaptive control algorithm functioning on the basis of switching. It is further expanded in e.g. [3, 4, 14, 16, 17, 18, 20, 42, 52, 1, 19]. An overview can be found in [15]. A formal definition of this estimator is given in Section 3.2, but a simplified description of its behavior is as follows. It is assumed that the true plant parameters θ^* lies inside some known, compact space $\mathcal{S}$. For each element in $\mathcal{S}$, the designer creates a stabilizing LTI controller via the certainty equivalence principle. Each element of $\mathcal{S}$ also has some performance metric which tracks how well it matches the observed input-output behavior. A system known as the 'supervisor' monitors each of these performance signals and occasionally switches which controller is in the feedback loop, based on which element of $\mathcal{S}$ has the smallest error, as illustrated in Figure 2.1. This occasional switching between continuous-time controllers results in a hybrid dynamical system. A 'dwell time'[3] is imposed to limit the rate at which the estimator is permitted to switch, which is a key component of the stability proof.

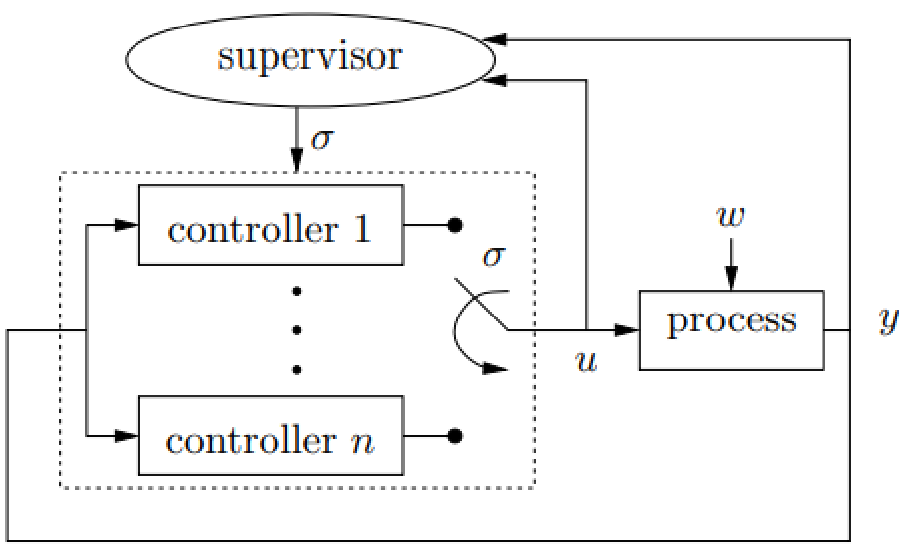

Figure 2.1: System diagram of a Supervisory Controller with n controller candidates [15]

We propose a different point of view for considering Supervisory Control: it turns out

[3]A dwell time of τ_D means that whenever the controller gains are changed, they cannot be changed again for at least τ_D steps.

that by taking Morse's original controller, converting it into discrete-time, and removing a couple extra complexities that aren't needed in the discrete-time setup, it becomes simply the following parameter estimator: for $t > t_0$,

$$\hat{\theta}(t) = \begin{cases} \hat{\theta}(t-1) & \text{if supervisor 'dwelling' at time } t \\ \underset{\theta \in \mathcal{S}}{\arg\min} \sum_{i=t_0}^{t-1} \lambda^{t-i-1} \left(y(i+1) - \phi(i)^\top \theta \right)^2 & \text{otherwise,} \end{cases}$$

combined with a suitable certainty equivalence control law. Here $\lambda \in (0, 1)$ is a forgetting factor chosen by the designer.

Observe that this estimator is the same as the weighted least-squares estimator, with two exceptions. Firstly, the true parameters θ^* and estimated parameters $\hat{\theta}$ are confined to a known set $\mathcal{S}$ in much the same way as in the Projection Algorithm setup used in [36, 38] and other successful adaptive controllers. Secondly, instead of allowing the estimator to change its estimate at every time-step, it is required to 'pause' for the duration of the dwell time. This similarity between Supervisory Control and the weighted least-squares estimator is not obvious to the casual reader of the rest of the Supervisory Control literature. In making these changes, we are indeed able to show that the closed-loop system admits a convolution bound of the sort (2.2). This is proven in Chapter 4 when the Supervisory estimator is paired with a d-step-ahead control law, and in Chapter 5 when it is paired with a pole placement controller with integrator. The convolution bound is then used to show that the system remains stable in the presence of unmodeled dynamics and time-varying parameters. While it is already proven in e.g. [41] that Supervisory Control is robust to unmodeled dynamics in the step tracking setup, it is not shown to tolerate time-varying parameters, except in [52], where a local stability result is shown for a slightly different estimator in the presence of slowly time-varying parameters. Since Supervisory Control has not been applied to d-step-ahead tracking before, the robustness results in this case are totally new.

Chapter 3

Supervisory Control Setup

3.1 Plant Definition

Let $t_0 \in \mathbf{Z}$ be a fixed, arbitrary initialization time. We begin by considering a plant described by a linear, time-invariant difference equation made up of a vector of true parameters θ^* and regression vector $\phi(t)$:

$$y(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{=:\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{=:\theta^*} + w(t), \quad t \geq t_0. \tag{3.1}$$

Here $y(t) \in \mathbf{R}$ is the plant's output, $u(t) \in \mathbf{R}$ is the control input, and $w \in l_\infty$ the disturbance (or noise) input. The system's initial condition $\phi(t_0)$ is assumed to be known. Later we will allow the plant's parameters to be time-varying, but the initial analysis is for a time-invariant plant.

Remark 3.1 *It turns out that if the system has a disturbance at both the input and output, then it can be easily converted to a system of the above form.*

Associated with this plant are the polynomials

$$A_{\theta^*}(z^{-1}) = 1 - a_1^* z^{-1} - \ldots - a_n^* z^{-n},$$
$$B_{\theta^*}(z^{-1}) = b_1^* z^{-1} + \ldots + b_m^* z^{-m},$$

and the associated transfer function

$$\frac{B_{\theta^*}(z^{-1})}{A_{\theta^*}(z^{-1})}.$$

The true parameter vector θ^* is assumed to be unknown but belongs to a known set $\mathcal{S} \subset \mathbf{R}^{n+m}$. We impose a set of assumptions on the set of admissible parameters.

Assumption 3.1 *n is known;*

Assumption 3.2 *m is known;*

Assumption 3.3 *$\mathcal{S}$ is compact.*

In later chapters, additional assumptions will be added, depending on what specific control law is being used.

Remark 3.2 *Since we do not require $a_n \neq 0$ nor $b_m \neq 0$, Assumptions 3.1 and 3.2 can be interpreted as assuming that an upper bound on n and m is known.*

Remark 3.3 *The restraint that $\mathcal{S}$ be compact in Assumption 3.3 is quite reasonable in practical situations; it is used here to prove uniform bounds and decay rates on the closed-loop behavior.*

Remark 3.4 *The upcoming parameter estimation will require optimization of a convex function over a set of admissible parameter estimates. Carrying out such optimization over the (typically) non-convex set $\mathcal{S}$ may be challenging. To alleviate this, since $\mathcal{S}$ is compact, it follows that there exists a finite number p of compact convex sets $\mathcal{S}_i \subset \mathbf{R}^{n+m}$ so that*

$$\hat{\mathcal{S}} := \bigcup_{i=1}^{p} \mathcal{S}_i \supset \mathcal{S}.$$

Thus the optimization will be performed over $\hat{\mathcal{S}}$, first by optimizing over each $\mathcal{S}_i$, and then choosing the best out of each of the p candidate optimizers.

3.2 Estimator Definition

We now formalize the precise definition of the Supervisory Estimator. Consider the plant definition in (3.1) and let $H(z)$ be a stable LTI filter with impulse response $h(t)$. If $\theta \in \hat{\mathcal{S}}$ is an estimate of θ^*, then the **prediction error** is defined as

$$e_\theta(t) := \sum_{j=0}^{t-t_0} h(j) \left(y(t-j) - \phi(t-j-1)^\top \theta \right), \quad t \geq t_0 + 1, \quad \theta \in \hat{\mathcal{S}};$$

using our convolution notation, this can be written as

$$e_\theta(t) = \{h \circ y\}(t) - (\{h \circ \phi\}(t-1))^\top \theta \quad t \geq t_0 + 1, \quad \theta \in \hat{\mathcal{S}}. \tag{3.2}$$

The filter $H(z)$ is referred to as the 'data filter' by Åström and Wittenmark [58, Chapter 11]. Its purpose is to filter out components of the observed data which are considered to be unreliable. For instance, if the plant is known to be influenced by a constant disturbance, then introducing a zero at $z = 1$ into $H(z)$ will eliminate its influence on the parameter estimation. Similarly, if the observations are corrupted by high frequency measurement noise, then $H(z)$ should be low-pass. In most practical applications, $H(z)$ should combine both of these, resulting in a band-pass filter.

The parameter estimator is built around the exponentially-weighted cost function with forgetting factor $\lambda \in (0, 1)$:

$$J(\theta, t) := \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_\theta(i+1)|^2, \quad t \geq t_0 + 1. \tag{3.3}$$

The parameter estimation routine is loosely defined as follows: occasionally, $\hat{\theta}(t)$ is chosen to be any $\theta \in \hat{\mathcal{S}}$ which minimizes $J(\theta, t)$. There may not be a unique minimizer: in this case, if $\hat{\theta}(t-1)$ is one of the minimizers, then $\hat{\theta}(t) = \hat{\theta}(t-1)$; if not, then $\hat{\theta}(t)$ is arbitrarily chosen to be any one of the minimizers. To limit the rate at which the estimator may switch, we introduce the notion of 'dwelling'; this is quite unusual in the adaptive control literature, but it will be needed in the later stability proof. Any time when the value of $\hat{\theta}$ changes, the supervisor goes into a state of 'dwelling' and remains as such for the next $\tau_D - 1$ steps, with τ_D being a positive integer chosen to be sufficiently long (the criteria for choosing the dwell-time will be provided later). During this time, $\hat{\theta}$ is held constant. Thus, the rate of switching of $\hat{\theta}$ is limited to at most once every τ_D time-steps.

To formalize this switching procedure, we introduce a new dynamic variable $\tau(t)$, dubbed the 'dwell-timer', constrained to the set $\{0, \ldots, \tau_D - 1\}$. Whenever τ takes a positive value, the supervisor is in the 'dwelling' state. With an initial condition $\tau(t_0) \in \{0, \ldots, \tau_D - 1\}$ and an initial parameter guess $\hat{\theta}(t_0) \in \hat{\mathcal{S}}$, for $t \geq t_0$, the parameter estimation routine is defined by

$$\begin{bmatrix} \hat{\theta}(t+1) \\ \tau(t+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} \arg\min\limits_{\theta \in \hat{\mathcal{S}}} J(\theta, t+1) \\ \tau_D - 1 \end{bmatrix} & \begin{array}{l} \text{if } \tau(t) = 0 \text{ and} \\ J(\hat{\theta}(t), t+1) > \min\limits_{\theta \in \hat{\mathcal{S}}} J(\theta, t+1) \end{array} \\ \begin{bmatrix} \hat{\theta}(t) \\ \max\{\tau(t) - 1, 0\} \end{bmatrix} & \text{otherwise.} \end{cases} \tag{3.4}$$

This fully defines the estimator dynamics. However, the act of minimizing $J(\theta, t)$ as defined in (3.3) seems difficult to implement. To make this optimization computationally easier, we employ a technique shown in [40]. The cost function $J(\theta, t)$ is reformulated by means of a new dynamical system called the 'performance weight generator' whose output is a weighting matrix $W(t) \in \mathbf{R}^{(n+m+1)\times(n+m+1)}$ with dynamics defined as

$$W(t+1) = \lambda W(t) + \begin{bmatrix} \{h \circ \phi\}(t) \\ \{h \circ y\}(t+1) \end{bmatrix} \begin{bmatrix} \{h \circ \phi\}(t) \\ \{h \circ y\}(t+1) \end{bmatrix}^\top, \quad t \geq t_0, \tag{3.5}$$

with initial condition $W(t_0) = 0$. In this manner, we find that

$$W(t) = \sum_{i=t_0}^{t-1} \lambda^{t-i-1} \left\{ \begin{bmatrix} \{h \circ \phi\}(i) \\ \{h \circ y\}(i+1) \end{bmatrix} \begin{bmatrix} \{h \circ \phi\}(i) \\ \{h \circ y\}(i+1) \end{bmatrix}^\top \right\}, \quad t > t_0.$$

Thus, the cost function (3.3) may be equivalently expressed in quadratic form as

$$J(\theta, t) = \begin{bmatrix} \theta \\ -1 \end{bmatrix}^\top W(t) \begin{bmatrix} \theta \\ -1 \end{bmatrix}, \quad t \geq t_0 + 1. \tag{3.6}$$

By Remark 3.4, $\hat{\mathcal{S}}$ is made up of a finite union of compact, convex sets. Thus, minimizing J over $\hat{\mathcal{S}}$ reduces to p straightforward convex optimization problems, which can be efficiently computed.

A consequence of this parameter estimation routine is that at every step for which the system is not dwelling ($\tau(t-1) = 0$), the inequality

$$\sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(t)}(i+1)|^2 \leq \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\theta}(i+1)|^2$$

holds for every θ in $\hat{\mathcal{S}}$. Most importantly, since θ^* is a member of $\hat{\mathcal{S}}$, it follows that

$$\sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(t)}(i+1)|^2 \leq \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |\{h \circ w\}(i)|^2, \quad t \geq t_0 + 1 \text{ s.t. } \tau(t-1) = 0. \tag{3.7}$$

3.3 Preliminary Technical Results

Here are some lemmas to be used in the following proofs. We start with a reformulation of Section VIII B of [41].

Lemma 3.1 *Let δ be a positive constant and let $\mathcal{X}$ be a list of vectors $x_1, x_2, \ldots, x_m \in \mathbf{R}^n$ whose first element satisfies $\|x_1\| \geq \delta$. Then there exists an ordered subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{\bar{n}}}\} \subseteq \mathcal{X}$, with $1 \leq \bar{n} \leq \max\{n, m\}$ and*

$$1 = i_1 < i_2 < \ldots < i_{\bar{n}} \leq m,$$

as well as coefficients $\alpha_{(i,j)} : \{1, \ldots, m\} \times \{1, \ldots, \bar{n}\} \to \mathbf{R}$ which satisfy

$$\begin{aligned} \alpha_{(i,j)} &= 0, & i &= 1, \ldots, i_j - 1, & j &= 2, \ldots, \bar{n}, \\ |\alpha_{(i,j)}| &\leq \left(1 + \frac{\|\mathcal{X}\|}{\delta}\right)^{\bar{n}}, & i &= i_j, \ldots, m, & j &= 1, \ldots, \bar{n}, \end{aligned}$$

as well as

$$\left\| x_i - \sum_{j=1}^{\bar{n}} \alpha_{(i,j)} x_{i_j} \right\| \leq \delta, \quad i = 1, \ldots, m.$$

This is quite similar to the ordinary procedure of taking a set of vectors $\{x_1, x_2, \ldots, x_m\}$ and extracting from it an ordered subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{\bar{n}}}\}$ which span the set, in which case for each x_i there exists a list of coefficients $\alpha_{(i,1)}, \ldots \alpha_{(i,\bar{n})}$ such that

$$x_i = \sum_{j=1}^{\bar{n}} \alpha_{(i,j)} x_{i_j}.$$

For this standard basis vector decomposition, the coefficients may be arbitrarily large. However, with our construction, we have traded equality for an approximation, and in return we are able to ensure that all the coefficients are bounded.

Proof: A proof is shown in [41], but for completeness one is also provided in Appendix A of this work. □

We also use a modified version of Kreisselmeier's Lemma from [22].

Lemma 3.2 *Consider the time-varying square matrix:*

$$\bar{A}(t) = A(t) + \Delta(t).$$

Let $\Phi(t,\tau)$ and $\bar{\Phi}(t,\tau)$ be the state transition matrices of $A(t)$ and $\bar{A}(t)$, respectively. Suppose there exists constants $c \geq 1$, $\mu \in (0,1)$ such that

$$\|\Phi(t,\tau)\| \leq c\mu^{t-\tau}, \quad t \geq \tau. \tag{3.8}$$

Then, for every $\bar{\mu} \in (\mu, 1)$, $\alpha \geq 0$, and $\beta \in [0, \frac{\bar{\mu}-\mu}{c})$, there exists a $\bar{c} \geq 1$ such that if

$$\sum_{t=t_1}^{t_2-1} \|\Delta(t)\| \leq \alpha + \beta(t_2 - t_1), \quad t_2 > t_1, \tag{3.9}$$

then the following bound holds:

$$\|\bar{\Phi}(t,\tau)\| \leq \bar{c}\bar{\mu}^{t-\tau}, \quad t \geq \tau. \tag{3.10}$$

Proof: If $t = \tau$ then $\bar{\Phi}(t,\tau) = I$, so clearly (3.10) holds. Now consider the case when $t > \tau$:

$$\begin{aligned}\|\bar{\Phi}(t,\tau)\| &\leq \|\Phi(t,\tau)\| + \sum_{i=\tau}^{t-1} \|\Phi(t,i+1)\|\|\Delta(i)\|\|\bar{\Phi}(i,\tau)\| \\ &\leq c\mu^{t-\tau} + \sum_{i=\tau}^{t-1} c\mu^{t-i-1}\|\Delta(i)\|\|\bar{\Phi}(i,\tau)\|\end{aligned}$$

Now introduce the change of variables $z(i) := \mu^{\tau-i}\|\bar{\Phi}(i,\tau)\|$:

$$\mu^{t-\tau}z(t) \leq c\mu^{t-\tau} + \sum_{i=\tau}^{t-1} c\mu^{t-i-1}\|\Delta(i)\|\mu^{i-\tau}z(i)$$

$$z(t) \leq c + \sum_{i=\tau}^{t-1} \frac{c}{\mu}\|\Delta(i)\|z(i),$$

and apply the Bellman-Gronwall inequality (e.g. [5, Appendix E])

$$
\begin{aligned}
z(t) &\le c \prod_{i=\tau}^{t-1} \left(1 + \frac{c}{\mu}\|\Delta(i)\|\right) \\
\|\bar{\Phi}(t,\tau)\| &\le c\mu^{t-\tau} \prod_{i=\tau}^{t-1} \left(1 + \frac{c}{\mu}\|\Delta(i)\|\right) \\
&= c \prod_{i=\tau}^{t-1} (\mu + c\|\Delta(i)\|).
\end{aligned}
$$

Finally, apply the inequality of arithmetic and geometric means and use (3.9):

$$
\begin{aligned}
\|\bar{\Phi}(t,\tau)\| &\le c \left(\frac{1}{t-\tau} \sum_{i=\tau}^{t-1} (\mu + c\|\Delta(i)\|) \right)^{t-\tau} \\
&\le c \left(\mu + c\beta + \frac{c\alpha}{t-\tau} \right)^{t-\tau}, \quad t > \tau.
\end{aligned}
$$

Since β is chosen such that $\mu + c\beta < \bar{\mu}$, there exists a $\bar{c} \ge 1$ such that (3.10) holds. □

Last of all, since we desire a convolution of the sort (2.2), but the cost function (3.3) that the estimator uses is a sum-of-squares, we will require a means to convert between them:

Lemma 3.3 *For any constant $\lambda \in (0,1)$ and signal $x : \mathbf{Z} \to \mathbf{R}$, the following inequalities hold:*

$$
\begin{aligned}
\sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)| &\le \sqrt{\frac{1}{1-\lambda} \sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)|^2}, \quad && i_2 > i_1, \\
\sqrt{\sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)|^2} &\le \sum_{i=i_1}^{i_2-1} \sqrt{\lambda}^{i_2-i-1}|x(i)|, && i_2 > i_1.
\end{aligned}
$$

Proof: The first inequality comes from the Cauchy-Schwarz inequality:

$$\begin{aligned}
\sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)| &\leq \sqrt{\sum_{i=i_1}^{i_2-1}\left(\sqrt{\lambda}^{i_2-i-1}\right)^2 \sum_{i=i_1}^{i_2-1}\left(\sqrt{\lambda}^{i_2-i-1}|x(i)|\right)^2}, \\
&= \sqrt{\frac{1-\lambda^{i_2-i_1}}{1-\lambda} \sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)|^2}, \\
&\leq \sqrt{\frac{1}{1-\lambda} \sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)|^2}.
\end{aligned}$$

The second inequality comes from simple algebra:

$$\begin{aligned}
\sum_{i=i_1}^{i_2-1} \lambda^{i_2-i-1}|x(i)|^2 &= \sum_{i=i_1}^{i_2-1}\left(\sqrt{\lambda}^{i_2-i-1}|x(i)|\right)^2 \\
&\leq \left(\sum_{i=i_1}^{i_2-1}\sqrt{\lambda}^{i_2-i-1}|x(i)|\right)^2.
\end{aligned}$$

□

Remark 3.5 *We can find a continuous-time analog to the first inequality of Lemma 3.3 via the Cauchy-Schwarz inequality: for any constant $\lambda > 0$ and signal $x : \mathbf{R} \to \mathbf{R}$,*

$$\begin{aligned}
\int_0^t e^{-\lambda(t-\tau)}|x(\tau)|d\tau &\leq \sqrt{\int_0^t \left(e^{-\frac{\lambda}{2}(t-\tau)}\right)^2 d\tau}\sqrt{\int_0^t \left(e^{-\frac{\lambda}{2}(t-\tau)}|x(\tau)|\right)^2 d\tau} \\
&\leq \sqrt{\frac{1}{\lambda}\int_0^t e^{-\lambda(t-\tau)}|x(\tau)|^2 d\tau}, \quad t \geq 0,
\end{aligned}$$

but there does not exist a continuous-time analog to the second inequality: it is not true that for any $\lambda > 0$, there exists constant $c \geq 0$ such that for all $x : \mathbf{R} \to \mathbf{R}$,

$$\sqrt{\int_0^t e^{-\lambda(t-\tau)}|x(\tau)|^2 d\tau} \leq c \int_0^t e^{-\lambda(t-\tau)}|x(\tau)|d\tau, \quad t \geq 0.$$

To see why this is so, consider the example of $x(\cdot) = 1$: then the above inequality becomes

$$\sqrt{\frac{1-e^{-\lambda t}}{\lambda}} \leq c\frac{1-e^{-\lambda t}}{\lambda}, \quad t \geq 0.$$

This would be true if and only if

$$1 \leq \frac{c}{\sqrt{\lambda}} \frac{1 - e^{-\lambda t}}{\sqrt{1 - e^{-\lambda t}}}, \quad t \geq 0.$$

But

$$\lim_{t \to 0^+} \frac{1 - e^{-\lambda t}}{\sqrt{1 - e^{-\lambda t}}} = 0,$$

so such a c cannot exist. Hence, while a bound involving[1]

$$\sqrt{\int_0^t e^{-\lambda(t-\tau)} |w(\tau)|^2 d\tau}$$

is discussed in equation (51) of [41] regarding the closed-loop behavior of continuous-time Supervisory Control, it does not follow that a convolution bound of the desired sort follows from that.

[1]In [41], $b(\cdot)$ represents the exogenous inputs; it is analogous to $w(\cdot)$ in our setup.

Chapter 4

d-Step-Ahead Adaptive Tracking

Here we consider the adaptive tracking problem where the Supervisory parameter estimator defined by (3.4) - (3.6), with $\lambda \in (0,1)$ to be chosen later, is combined with the d-step-ahead control law. This has never been done before in the Supervisory Control approach. Recall that the plant model being considered is

$$y(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{\theta^*} + w(t), \quad t \geq t_0. \tag{4.1}$$

The controller's goal is to make the plant's output $y(t)$ track an arbitrary bounded reference signal $r(t)$ using a bounded control signal $u(t)$. There are no assumptions made on the nature of the disturbance, so for this chapter we let the data filter $H(z)$ be 1. Beside Assumptions 3.1 - 3.3, which are assumed to hold, we will enforce an additional assumption on the set of admissible parameters $\mathcal{S}$ to ensure that the d-step-ahead control law is well behaved:

Assumption 4.1 *The polynomial $B_\theta(z^{-1})$ corresponding to each θ in $\mathcal{S}$ has all of its zeros in the open unit disk.*

Remark 4.1 *Assumption 4.1 is a requirement that the plant be minimum phase; this is necessary for a causal controller to provide asymptotic tracking of an arbitrary bounded reference signal using a bounded control signal [30]. Moreover, since the roots of a polynomial are a continuous function of its coefficients, and the coefficients of $B_\theta(z^{-1})$ lie within a compact set, it follows that there exists a $\lambda_S < 1$ such that the zeros of $B_\theta(z^{-1})$ all lie within the open disk of radius λ_S.*

Remark 4.2 *Assumption 3.3 and Assumption 4.1 imply that the leading coefficient of $B_\theta(z^{-1})$, which is the system's 'high frequency gain', is bounded away from zero.*

Remark 4.3 *The components $\mathcal{S}_i$ that make up $\hat{\mathcal{S}}$ can be chosen such that each element of $\mathcal{S}_i$ satisfies Assumption 4.1, the corresponding polynomial $B_\theta(z^{-1})$ has all of its zeros within the open disk of radius λ_S, and each model $\frac{B_\theta(z^{-1})}{A_\theta(z^{-1})}$ has the same relative degree, so henceforth we will assume that this is the case.*

4.1 Control Law

Here we define what is known as the 'd-step-ahead' control law, which is the causal controller which would drive y to the arbitrary reference r should the parameter vector θ^* be known and the disturbance w be zero. It turns out that this controller places some of the closed-loop poles at the location of the plant's zeros, with the rest of the poles placed at the origin. Assumption 4.1 guarantees that all these pole-zero cancellations are stable. The simplified closed-loop transfer function is then $z^{-d_{\theta^*}}$, where d_{θ^*} is the relative degree of the plant. To implement this, the controller is provided with $r(t + d_{\theta^*})$ - the reference signal advanced d_{θ^*} steps forward in time. This is what inspires the control law's name. Its construction is as follows.

Consider first the simple case where $d_{\theta^*} = 1$. Then the 'one-step-ahead' control law is the $u(t)$ that satisfies the equation

$$r(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{\theta^*}, \quad t \geq t_0 + 1.$$

However, if $d_{\theta^*} > 1$, then the term b_1^* will be zero, so this equation will not have a solution for $u(t)$. In this case, the system must be reparameterized into the so-called 'predictor form'. The process for doing so is shown in Lemma 4.2.1 of [12] and is also provided here. First, observe that the plant can be rewritten as

$$y(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{\theta^*} + w(t), \quad t \geq t_0.$$

$$\Longrightarrow \begin{bmatrix} 1 \\ -a_1^* \\ \vdots \\ -a_n^* \end{bmatrix}^\top \begin{bmatrix} y(t+1) \\ y(t) \\ \vdots \\ y(t-n+1) \end{bmatrix} = \begin{bmatrix} 0 \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}^\top \begin{bmatrix} u(t+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix} + w(t), \quad t \geq t_0.$$

$$\Longrightarrow \quad A_{\theta^*}(q^{-1})y(t+1) = B_{\theta^*}(q^{-1})u(t+1) + w(t), \quad t \geq t_0. \tag{4.2}$$

Now let $B'_{\theta^*}(q^{-1}) = q^{d_{\theta^*}} B_{\theta^*}(q^{-1})$. This ensures that the leading coefficient of $B'_{\theta^*}(q^{-1})$ is non-zero. Now for each $\theta \in \hat{\mathcal{S}}$, let

$$F_\theta(z^{-1}) := 1 + f_{\theta 1} z^{-1} + \ldots + f_{\theta(d_\theta - 1)} z^{-d_\theta + 1} \tag{4.3}$$

and

$$G_\theta(z^{-1}) := g_{\theta 1} + g_{\theta 2} z^{-1} + \ldots + g_{\theta n} z^{-n+1}$$

be the unique solution to the equation

$$F_\theta(q^{-1}) A_\theta(q^{-1}) + q^{-d_\theta} G_\theta(q^{-1}) = 1. \tag{4.4}$$

This may be obtained, for example, by using long division, as in [36]. Now we operate on each side of (4.2) by $F_{\theta^*}(q^{-1})$:

$$F_{\theta^*}(q^{-1}) A_{\theta^*}(q^{-1}) y(t + d_{\theta^*}) = F_{\theta^*}(q^{-1}) B_{\theta^*}(q^{-1}) u(t + d_{\theta^*}) + F_{\theta^*}(q^{-1}) w(t + d_{\theta^*} - 1), \quad t \geq t_0,$$

and apply $F_{\theta^*}(q^{-1})A_{\theta^*}(q^{-1}) = (1 - q^{-d_{\theta^*}} G_{\theta^*}(q^{-1}))$ to the LHS and rearrange to yield

$$y(t + d_{\theta^*}) = G_{\theta^*}(q^{-1}) y(t) + F_{\theta^*}(q^{-1}) B_{\theta^*}(q^{-1}) u(t + d_{\theta^*}) + F_{\theta^*}(q^{-1}) w(t + d_{\theta^*} - 1), \quad t \geq t_0.$$

Hence, if we define

$$\beta_{\theta 1} + \beta_{\theta 2} z^{-1} + \ldots + \beta_{\theta n} z^{-n+1} := F_\theta(z^{-1}) B'_\theta(z^{-1}),$$
$$\bar{w}(t) := F_{\theta^*}(q^{-1}) w(t + d_{\theta^*-1}),$$

then we can express the plant as

$$y(t+d_{\theta^*}) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}}_{\phi(t)}^{\top} \begin{bmatrix} g_{\theta^* 1} \\ \vdots \\ g_{\theta^* n} \\ \beta_{\theta^* 1} \\ \vdots \\ \beta_{\theta^* m} \end{bmatrix} + \bar{w}(t), \quad t \geq t_0.$$

The nature of a certainty equivalence controller is that we interpret the parameter estimate $\hat{\theta}$ as though it were the true parameter; thus we choose the control input $u(t)$ to be that which makes the predicted output equal to the reference. Hence, the control law is that which satisfies the equation [1]

$$r(t+d_{\hat{\theta}(t)}) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}}_{\phi(t)}^{\top} \begin{bmatrix} g_{\hat{\theta}(t) 1} \\ \vdots \\ g_{\hat{\theta}(t) n} \\ \beta_{\hat{\theta}(t) 1} \\ \vdots \\ \beta_{\hat{\theta}(t) m} \end{bmatrix}, \quad t \geq t_0 + 1,$$

or equivalently, that which satisfies the difference equation

$$r(t+d_{\hat{\theta}(t)}) = G_{\hat{\theta}(t)}(q^{-1}) y(t) + F_{\hat{\theta}(t)}(q^{-1}) B'_{\hat{\theta}(t)}(q^{-1}) u(t), \quad t \geq t_0 + 1. \tag{4.5}$$

We want to rearrange this equation to solve for $u(t)$. To this end, observe that each model's delay lies within $\{1, \ldots, m\}$, and define

$$\bar{r}(t) := \begin{bmatrix} r(t+1) \\ \vdots \\ r(t+m) \end{bmatrix}, \quad f_\theta := -\begin{bmatrix} \frac{g_{\theta 2}}{\beta_{\theta 1}} & \ldots & \frac{g_{\theta n}}{\beta_{\theta 1}} & 0 & \frac{\beta_{\theta 2}}{\beta_{\theta 1}} & \ldots & \frac{\beta_{\theta m}}{\beta_{\theta 1}} & 0 \end{bmatrix}, \quad g_\theta := -\frac{g_{\theta 1}}{\beta_{\theta 1}},$$

[1] $u(t_0)$ is given as part of the initial condition $\phi(t_0)$, so the control law only holds for $t \geq t_0 + 1$.

and let $h_\theta \in \mathbf{R}^{1\times m}$ be the vector whose d_θ^{th} element is equal to $\frac{1}{\beta_{\theta 1}}$ and zero elsewhere. Then the d-step-ahead control law is written as

$$u(t) = f_{\hat{\theta}(t)}\phi(t-1) + g_{\hat{\theta}(t)}y(t) + h_{\hat{\theta}(t)}\bar{r}(t), \quad t \geq t_0 + 1. \tag{4.6}$$

Observe that the term $\beta_{\theta 1}$ is bounded away from zero due to Assumptions 3.3 and 4.1 (see Remark 4.2). This plus the compactness of $\hat{\mathcal{S}}$ means that all coefficients in the control law are uniformly bounded for $\theta \in \hat{\mathcal{S}}$.

4.2 State Space Representation

As in Chapter 2, define the **parameter estimation error** as

$$\tilde{\theta}(t) := \hat{\theta}(t) - \theta^*, \quad t \geq t_0.$$

With the data filter from Chapter 3 chosen as $H(z) = 1$, the **prediction error** (3.2) can be rewritten as

$$\begin{aligned} e_{\hat{\theta}(t)}(i) &= y(i) - \phi(i-1)^\top \hat{\theta}(t) \\ &= \phi(i-1)^\top \theta^* + w(i-1) - \phi(i-1)^\top \hat{\theta}(t), \\ &= w(i-1) - \phi(i-1)^\top \tilde{\theta}(t), \quad t \geq t_0, \quad i \geq t_0 + 1. \end{aligned} \tag{4.7}$$

We can now form an update equation for $\phi(t)$. Start by defining, for every $k \in \mathbf{N}$,

$$\bar{A}_k := \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbf{R}^{k\times k}, \quad \bar{b}_k := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^k;$$

then begin the state-space construction with

$$\phi(t+1) = \begin{bmatrix} \bar{A}_n & 0 \\ 0 & \bar{A}_m \end{bmatrix} \phi(t) + \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} y(t+1) + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} u(t+1), \quad t \geq t_0.$$

Now incorporate the control law (4.6):

$$\begin{aligned} \phi(t+1) = \left(\begin{bmatrix} \bar{A}_n & 0 \\ 0 & \bar{A}_m \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} f_{\hat{\theta}(t+1)} \right) \phi(t) + \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(t+1)} \end{bmatrix} y(t+1) \\ + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} h_{\hat{\theta}(t+1)} \bar{r}(t+1) \quad t \geq t_0. \end{aligned} \tag{4.8}$$

Finally, substitute in the plant dynamics from (4.1): with

$$A_{\theta_1\theta_2} := \begin{bmatrix} \bar{A}_n & 0 \\ 0 & \bar{A}_m \end{bmatrix} + \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} {\theta_1}^\top + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} (f_{\theta_2} + g_{\theta_2}{\theta_1}^\top),$$

we end up with

$$\phi(t+1) = A_{\theta^*\hat{\theta}(t+1)}\phi(t) + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} h_{\hat{\theta}(t+1)}\bar{r}(t+1) + \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(t+1)} \end{bmatrix} w(t), \quad t \geq t_0. \tag{4.9}$$

Alternatively, from (4.7), substitute $y(t+1) = e_{\hat{\theta}(t+1)}(t+1) + \hat{\theta}(t+1)^\top\phi(t)$ into (4.8) to yield another representation:

$$\phi(t+1) = A_{\hat{\theta}(t+1)\hat{\theta}(t+1)}\phi(t) + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} h_{\hat{\theta}(t+1)}\bar{r}(t+1) + \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(t+1)} \end{bmatrix} e_{\hat{\theta}(t+1)}(t+1), \quad t \geq t_0. \tag{4.10}$$

This form is useful because $A_{\theta\theta}$ represents the closed-loop system dynamics if $\hat{\theta}(\cdot) = \theta^* = \theta \in \hat{\mathcal{S}}$. Due to the certainty equivalence control law, the d-step-ahead controller places some of the eigenvalues of this matrix at the zeros of the corresponding plant model, and the rest at the origin. From Remark 4.3, it follows that for each fixed $\theta \in \hat{\mathcal{S}}$, the matrix $A_{\theta\theta}$ is stable with margin[2] $\lambda_S \in (0,1)$. Notice also that $\|A_{\theta\theta}\|$ is uniformly bounded due to the compactness of $\hat{\mathcal{S}}$ and the uniform boundedness of f_θ and g_θ. It follows that there exists a $\hat{\gamma}$ such that

$$\|A^k_{\theta\theta}\| \leq \hat{\gamma}\lambda^k_S, \quad k \geq 0, \quad \theta \in \hat{\mathcal{S}}. \tag{4.11}$$

Finally, define these constants for later:

$$\gamma_1 := 1 + \sup_{\theta\in\hat{\mathcal{S}}} |g_\theta|, \quad \gamma_2 := \sup_{\theta\in\hat{\mathcal{S}}} \|h_\theta\|.$$

4.3 Nominal Tracking

At this point, when the disturbance is zero, we will derive a bound on the size of the **tracking error**

$$\epsilon(t) := y(t) - r(t)$$

[2]Stability margin refers to the eigenvalues of the matrix lying within the open disk of some specified radius.

in terms of the size of the initial condition and the size of the reference signal. Quite surprisingly, we are able to do so before proving any form of closed-loop stability.

Theorem 4.1 *For every $\tau_D \geq 1$, there exists a constant $c \geq 1$ such that for every $t_0 \in \mathbf{Z}$, $\lambda \in (0,1)$, $\theta^* \in \mathcal{S}$, $\phi(t_0) \in \mathbf{R}^{n+m}$, $\tau(t_0) \in \{0, \dots, \tau_D - 1\}$ and $r \in l_\infty$, if $w = 0$, when the supervisory controller given by (3.4) - (3.6) and (4.6) is applied to the plant (4.1), the following bound holds:*

$$\sum_{t=t_0}^{\infty} |\epsilon(t)| \leq c\left(\|r\|_\infty + \|\phi(t_0)\|\right).$$

Remark 4.4 *In most adaptive control papers on the d-step-ahead control problem, it is proven only that*

$$\sum_{t=t_0}^{\infty} |\epsilon(t)|^2 < \infty.$$

Clearly Theorem 4.1 proves something much stronger; using Lemma 3.3, we find

$$\sum_{t=t_0}^{\infty} |\epsilon(t)|^2 \leq c^2\left(\|r\|_\infty + \|\phi(t_0)\|\right)^2.$$

Proof: Fix $\lambda \in (0,1)$, set $w = 0$ and let $\theta^* \in \mathcal{S}$, $\phi(t_0) \in \mathbf{R}^{n+m}$ and $r \in l_\infty$ be arbitrary. Let $t_1, t_2, \dots$ be the time indices where $\hat{\theta}$ switches. Since the supervisor is necessarily not dwelling at these switch times, from (3.7), the following equality holds for each switching time t_j:

$$\sum_{i=t_0}^{t_j-1} \lambda^{t_j-i-1} |e_{\hat{\theta}(t_j)}(i+1)|^2 = 0, \quad j \geq 1. \tag{4.12}$$

Since this is a sum of non-negative components, it must be that each $e_{\hat{\theta}(t_j)}(i+1) = 0$ for $i \in \{t_0, \dots, t_j - 1\}$. Since the disturbance is absent, $e_{\hat{\theta}(t_j)}(i+1)$ is simply $-\phi(i)^\top \tilde{\theta}(t_j)$. Thus, by defining

$$\mathcal{N}_0 := \mathbf{R}^{n+m},$$

$$\mathcal{N}_j := \ker\left\{\begin{bmatrix} \phi(t_0)^\top \\ \vdots \\ \phi(t_j-1)^\top \end{bmatrix}\right\}, \quad j \geq 1,$$

it follows that $\mathcal{N}_{j+1} \subset \mathcal{N}_j, j \geq 0$, and for each $j \geq 0$, $\tilde{\theta}(t_j) \in \mathcal{N}_j$.

Recall that by (3.4), the supervisor only changes $\hat{\theta}$ if it finds a new candidate in $\hat{\mathcal{S}}$ which is *better* than the previous value of $\hat{\theta}$. Thus, it must be that for each switching time t_j,

$$\sum_{i=t_0}^{t_{j+1}-1} \lambda^{t_{j+1}-i-1} |e_{\hat{\theta}(t_j)}(i+1)|^2 > \sum_{i=t_0}^{t_{j+1}-1} \lambda^{t_{j+1}-i-1} |e_{\hat{\theta}(t_{j+1})}(i+1)|^2 = 0, \quad j \geq 0.$$

This implies that $\tilde{\theta}(t_j) \notin \mathcal{N}_{j+1}$, specifically $\tilde{\theta}(t_j) \in \mathcal{N}_j \setminus \mathcal{N}_{j+1}$ for each $j \geq 0$, and hence $\dim\{\mathcal{N}_{j+1}\} < \dim\{\mathcal{N}_j\}, j \geq 0$. Since the dimension of $\mathcal{N}_0$ is $n+m$, it follows that $\hat{\theta}$ can't switch more than $n+m$ times. Thus, let $N \leq n+m$ be the number of switches, so $t_1, t_2, \ldots, t_N$ is the complete list of switching times.

The key part of this proof is to find a relationship between the tracking error ϵ and the prediction error $e_{\hat{\theta}}$. This is done by taking advantage of the d-step-ahead control law. Consider first the simple case where $\hat{\theta}(t)$ corresponds to a plant model $\frac{B_{\hat{\theta}(t)}(z^{-1})}{A_{\hat{\theta}(t)}(z^{-1})}$ with relative degree one. Then we use the 'one-step-ahead' control law [3]

$$r(t+1) = \phi(t)^\top \hat{\theta}(t), \quad t \geq t_0 + 1 \text{ s.t. } d_{\hat{\theta}(t)} = 1,$$

and it is clear that

$$\begin{aligned}
\epsilon(t+1) &= y(t+1) - r(t+1) \\
&= y(t+1) - \phi(t)^\top \hat{\theta}(t) \\
&= e_{\hat{\theta}(t)}(t+1), \quad t \geq t_0 + 1 \text{ s.t. } d_{\hat{\theta}(t)} = 1.
\end{aligned}$$

It is more complicated when the plant model has a greater delay, as the 'predictor form' transformation shown in Section 4.1 must be used. Using (4.4) and (4.5), one can see that:

$$\begin{aligned}
\epsilon(t + d_{\hat{\theta}(t)}) &= y(t + d_{\hat{\theta}(t)}) - r(t + d_{\hat{\theta}(t)}) \\
&= y(t + d_{\hat{\theta}(t)}) - \Big(G_{\hat{\theta}(t)}(q^{-1}) y(t) + F_{\hat{\theta}(t)}(q^{-1}) B'_{\hat{\theta}(t)}(q^{-1}) u(t) \Big) \\
&= \Big(1 - q^{-d_{\hat{\theta}(t)}} G_{\hat{\theta}(t)}(q^{-1}) \Big) y(t + d_{\hat{\theta}(t)}) - F_{\hat{\theta}(t)}(q^{-1}) B_{\hat{\theta}(t)}(q^{-1}) u(t + d_{\hat{\theta}(t)}) \\
&= F_{\hat{\theta}(t)}(q^{-1}) \Big(A_{\hat{\theta}(t)}(q^{-1}) y(t + d_{\hat{\theta}(t)}) - B_{\hat{\theta}(t)}(q^{-1}) u(t + d_{\hat{\theta}(t)}) \Big), \quad t \geq t_0 + 1.
\end{aligned}$$

[3]Recall that $u(t_0)$ is an element of the the initial condition $\phi(t_0)$, so the control law only holds for $t \geq t_0 + 1$.

The last term above corresponds to the prediction error with respect to the estimate $\hat{\theta}(t)$:

$$\begin{aligned} e_\theta(t) &:= y(t) - \underbrace{\begin{bmatrix} y(t-1) \\ \vdots \\ y(t-n) \\ u(t-1) \\ \vdots \\ u(t-m) \end{bmatrix}^\top}_{\phi(t-1)} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_m \end{bmatrix}}_{\theta} \\ &= \begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ -a_n \end{bmatrix}^\top \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-n) \end{bmatrix} - \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}^\top \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-m) \end{bmatrix} \\ &= A_\theta(q^{-1})y(t) - B_\theta(q^{-1})u(t), \quad t \geq t_0 + 1. \end{aligned}$$

Thus, using the coefficients of $F_\theta(q^{-1})$ as defined in (4.3), one can express the tracking error as a function of the delayed prediction errors:

$$\epsilon(t+d_{\hat{\theta}(t)}) = e_{\hat{\theta}(t)}(t+d_{\hat{\theta}(t)}) + f_{\hat{\theta}(t)1} e_{\hat{\theta}(t)}(t+d_{\hat{\theta}(t)}-1) + ... + f_{\hat{\theta}(t)(d_{\hat{\theta}(t)}-1)} e_{\hat{\theta}(t)}(t+1), \quad t \geq t_0+1.$$

Using the fact that $\hat{\theta}$ is constant on the interval $\{t_j, \dots, t_{j+1}-1\}, j \geq 0$:

$$\begin{aligned} \epsilon(t) &= F_{\hat{\theta}(t_j)}(q^{-1}) e_{\hat{\theta}(t_i)}(t) \\ &= e_{\hat{\theta}(t_j)}(t) + f_{\hat{\theta}(t_j)1} e_{\hat{\theta}(t_j)}(t-1) + ... + f_{\hat{\theta}(t_j)(d_{\hat{\theta}(t_j)}-1)} e_{\hat{\theta}(t_j)}(t - d_{\hat{\theta}(t_j)} + 1), \\ & t \in \left\{ t_j + d_{\hat{\theta}(t_j)}, \dots, t_{j+1} + d_{\hat{\theta}(t_j)} - 1 \right\}, \quad t \neq t_0 + d_{\hat{\theta}(t_0)}, \quad j \geq 0. \end{aligned} \tag{4.13}$$

We use (4.12) in conjunction with (4.13) to find a bound on the tracking error. To do so, we break up the sum of the tracking errors as several partial sums:

$$\sum_{t=t_0}^{\infty} |\epsilon(t)| = \sum_{t=t_0}^{t_1-1} |\epsilon(t)| + \sum_{t=t_1}^{t_2-1} |\epsilon(t)| + \ldots + \sum_{t=t_N}^{\infty} |\epsilon(t)|. \tag{4.14}$$

Now we analyze each of the partial sums; each will fall into one of two cases, to be analyzed separately.

Case 1: $e_{\hat{\theta}(t_j)}(t) \neq 0$ for some t in $\{t_j+1, \ldots, t_j+\tau_D-1\}$. In this case, the estimator will switch as soon as it ceases dwelling, i.e. $t_{j+1} = t_j + \tau_D$. We simply use the worst-case dynamics to find a bound on the tracking error. By observing that $y(t)$ is an element of $\phi(t)$, we begin with:

$$\sum_{t=t_j}^{t_{j+1}-1} |\epsilon(t)| \leq \sum_{t=t_j}^{t_j+\tau_D-1} (\|\phi(t)\| + \|r\|_\infty). \tag{4.15}$$

Now use (4.9) to find a bound for ϕ on this interval. Let $\bar{a} := \sup_{\theta_1 \in \mathcal{S}, \theta_2 \in \hat{\mathcal{S}}} \|A_{\theta_1\theta_2}\|$, which is finite because $\mathcal{S}$ and $\hat{\mathcal{S}}$ are compact and f_θ and g_θ are uniformly bounded. Thus,

$$\|\phi(t)\| \leq \bar{a}^{t-t_j+1}\|\phi(t_j-1)\| + \gamma_2 \frac{\bar{a}^{t-t_j+1}-1}{\bar{a}-1}\|r\|_\infty, \quad t \in \{t_j, \ldots, t_j+\tau_D-1\}. \tag{4.16}$$

Case 2: $e_{\hat{\theta}(t_j)}(t) = 0$ for for t in $\{t_j+1, \ldots, t_j+\tau_D-1\}$. Since the supervisor ceases dwelling at time $t_j+\tau_D$, we know that the prediction error must remain zero up until the next time $\hat{\theta}$ switches, thus:

$$e_{\hat{\theta}(t_j)}(t) = 0, \quad t \in \{t_j, \ldots, t_{j+1}-1\}.$$

Since $\hat{\theta}(t)$ and hence $d_{\hat{\theta}(t)}$ are constant on the interval $\{t_j, \ldots, t_{j+1}-1\}$, it follows from (4.13) that the tracking error $\epsilon(t)$ is exactly zero on the interval $\left\{t_j + d_{\hat{\theta}(t_j)}, \ldots, t_{j+1}-1\right\}$ (the exception is in the first switching period - $\epsilon(t)$ is zero on the interval $\left\{t_0 + d_{\hat{\theta}(t_0)}+1, \ldots, t_1-1\right\}$). Therefore,[4]

$$\sum_{t=t_j}^{t_{j+1}-1} |\epsilon(t)| \leq \begin{cases} \displaystyle\sum_{t=t_0}^{t_0+m} (\|\phi(t)\| + \|r\|_\infty) & j=0 \\ \displaystyle\sum_{t=t_j}^{t_j+m-1} (\|\phi(t)\| + \|r\|_\infty) & \text{otherwise.} \end{cases} \tag{4.17}$$

Notice that on the interval $\{t_j, \ldots, t_{j+1}-1\}$, (4.10) is a stable system whose input $e_{\hat{\theta}(t)}(t)$ is zero. Thus we use (4.11) to show that

$$\|\phi(t)\| \leq \hat{\gamma}\lambda_S^{t-t_j+1}\|\phi(t_j-1)\| + \hat{\gamma}\gamma_2 \frac{1-\lambda_S^{t-t_j+1}}{1-\lambda_S}\|r\|_\infty, \quad t \in \{t_j, \ldots, t_{j+1}-1\}. \tag{4.18}$$

[4] Using the knowledge that the delay can be no greater than m.

Both cases: By combining (4.16) and (4.18), it follows that there exists a constant γ_3 such that for each $j \in \{1, \ldots, N\}$,

$$\|\phi(t_j - 1)\| \leq \bar{a}^{j\tau_D} \|\phi_(t_0)\| + \gamma_3 \frac{\bar{a}^{j\tau_D} - 1}{\bar{a}^{\tau_D} - 1} \|r\|_\infty. \tag{4.19}$$

Now consider (4.14). The final term, for $t \geq t_N$, must fall into Case 2. For it and every other interval in Case 2, we use (4.17), (4.18) and (4.19). Similarly, for every interval in Case 1, we use (4.15), (4.16) and (4.19). This yields the desired result. $\square$

4.4 Closed-loop Stability

Before presenting the main result on stability, we return to the issue of dwell time. The matrix $A_{\theta\theta}$ plays a critical role here. We know from (4.11) that $A_{\theta\theta}$ is stable for any fixed $\theta \in \hat{\mathcal{S}}$. Of course, here $\hat{\theta}$ is time-varying and we will need to impose a constraint to ensure that the state transition matrix Φ corresponding to $A_{\hat{\theta}(t)\hat{\theta}(t)}$ is also well-behaved. To make our stability proof work we will require that the filter gain λ be larger than the maximum of the plant zeros: $\lambda \in (\lambda_S, 1)$. Now choose $\tilde{\lambda} \in (\lambda_S, \lambda)$, and suppose that the goal is to ensure that $\Phi(t_2, t_1)$ goes to zero at least as fast as $\tilde{\lambda}^{t_2 - t_1}$ as $t_2 - t_1 \to \infty$. If the dwell time is τ_D, then we expect that this will be achieved if

$$\hat{\gamma}\lambda_S^{\tau_D} \leq \tilde{\lambda}^{\tau_D} \quad \Leftrightarrow \quad \tau_D \geq \frac{ln(\hat{\gamma})}{ln\left(\frac{\tilde{\lambda}}{\lambda_S}\right)}; \tag{4.20}$$

indeed, if this is the case, then it is easy to see that

$$\|\Phi(t_2, t_1)\| \leq \hat{\gamma}\tilde{\lambda}^{t_2 - t_1}, \quad t_2 \geq t_1 \geq t_0.$$

Remark 4.5 *If all admissible plant models contain no zeros, or only zeros at the origin, then each $A_{\theta\theta}$ is deadbeat and the minimum required dwell time is $n + m$.*

Therefore, if $\hat{\theta}(t)$ is any piecewise-constant signal with dwell-time satisfying (4.20), then the system (4.10) is exponentially stable, which reveals the convolution bound, albeit in terms of internal signals rather than exogenous ones: with $\bar{t}_0 \geq t_0$,

$$\|\phi(t)\| \leq \hat{\gamma}\tilde{\lambda}^{t - \bar{t}_0} \|\phi(\bar{t}_0)\| + \hat{\gamma} \sum_{i=\bar{t}_0}^{t-1} \tilde{\lambda}^{t-i-1} \left(\gamma_1 |e_{\hat{\theta}(i+1)}(i+1)| + \gamma_2 \|\bar{r}(i+1)\|\right), \quad t \geq \bar{t}_0.$$

Thus, a stability proof relies on finding a meaningful bound on the term

$$\sum_{i=\bar{t}_0}^{t-1} \tilde{\lambda}^{t-i-1} |e_{\hat{\theta}(i+1)}(i+1)|. \tag{4.21}$$

In the trivial case where $n = 0$ and $m = 1$ (the plant is a delay with a gain), we find that the minimum dwell-time is $\tau_D = 1$ and the parameter estimation law (3.4) - (3.6) simplifies to

$$\hat{\theta}(t) = \arg\min_{\theta \in \hat{\mathcal{S}}} \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_\theta(i+1)|^2, \quad t \geq t_0.$$

From this it may be inferred that

$$\sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(t)}(i+1)|^2 \leq \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |w(i)|^2, \quad t \geq t_0,$$

and

$$\sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(t)}(i+1)|^2 \leq \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(t+1)}(i+1)|^2, \quad t \geq t_0.$$

Claim 4.1 *The two inequalities above may be used to show that*

$$\sum_{i=t_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(i+1)}(i+1)|^2 \leq \sum_{i=t_0}^{t-1} \lambda^{t-i-1} |w(i)|^2, \quad t \geq t_0.$$

Proof: See Appendix B. □

This result, when combined with Lemma 3.3 and the assertion that $\tilde{\lambda} < \lambda$, can be used to find a useful bound for (4.21), showing that the system is exponentially stable with a convolution bound. Thus, the stability proof is very straightforward in this trivial case, but in the general case where $n + m > 1$, it is much more complicated. In essence, the following stability proof is just finding a useful bound for (4.21) in the case where the dwell-time is greater than 1.

Now we present the main stability proof of this chapter. First notice that the overall system state is uniquely defined by $\phi(t)$, $W(t)$, $\hat{\theta}(t)$ and $\tau(t)$. Since $W(t)$ is a matrix, we vectorize it so that the overall system state vector may be expressed as

$$x(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \\ \hat{\theta}(t) \\ \tau(t) \end{bmatrix}.$$

It will become clear later why we choose $W(t)^{\frac{1}{2}}$ instead of $W(t)$. Now split the state up into components $\bar{x}(t)$ and $\tilde{x}(t)$:

$$\bar{x}(t) := \begin{bmatrix} \phi(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}, \quad \tilde{x}(t) := \begin{bmatrix} \hat{\theta}(t) \\ \tau(t) \end{bmatrix}.$$

The following theorem shows that $\bar{x}(t)$ is exponentially stable uniformly in $\tilde{x}(t)$, and admits a convolution bound.

Theorem 4.2 *For every $\lambda \in (\lambda_S, 1)$, $\tilde{\lambda} \in (\lambda_S, \lambda)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and τ_D satisfying (4.20), there exists a $\gamma \geq 1$ so that for every $\theta^* \in \mathcal{S}$, $\bar{t}_0 \geq t_0$, $\phi(\bar{t}_0) \in \mathbf{R}^{n+m}$, $W(\bar{t}_0) \in \mathbf{R}^{(n+m+1)\times(n+m+1)}$ positive semidefinite and symmetric, $\tau(\bar{t}_0) \in \{0, \ldots, \tau_D - 1\}$, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, and $r, w \in l_\infty$, when the supervisory controller given by (3.4) - (3.6) and (4.6) is applied to the plant (4.1), the following bounds hold:*

$$\|\bar{x}(t)\| \leq \gamma\bar{\lambda}^{t-\bar{t}_0}\|\bar{x}(\bar{t}_0)\| + \gamma\sum_{i=\bar{t}_0}^{t-1}\bar{\lambda}^{t-i-1}\left(|w(i)| + \|\bar{r}(i+1)\|\right), \qquad t \geq \bar{t}_0, \quad (4.22)$$

$$\|\bar{x}(t)\| \leq \gamma\bar{\lambda}^{t-\bar{t}_0}\|\bar{x}(\bar{t}_0)\| + \gamma\sum_{i=\bar{t}_0}^{t-1}\bar{\lambda}^{t-i-1}|w(i)| + \gamma\sum_{i=\bar{t}_0+1}^{t+m-1}\bar{\lambda}^{t-i-1}|r(i+1)|, \quad t \geq \bar{t}_0. \quad (4.23)$$

Proof of Theorem 4.2

Fix $\lambda \in (\lambda_S, 1)$, $\tilde{\lambda} \in (\lambda_S, \lambda)$, and $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and let $\theta^* \in \mathcal{S}$, $\bar{t}_0 \geq t_0$, $\phi(\bar{t}_0) \in \mathbf{R}^{n+m}$, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, $W(\bar{t}_0)$ positive semidefinite and symmetric, $\tau(\bar{t}_0) \in \{0, \ldots, \tau_D - 1\}$, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, and $r, w \in l_\infty$ be arbitrary.

To prove this, we split up time into those for which $\|\tilde{\theta}(t)\|$ is small and those for which it is not. Before proceeding, recall from (4.11) that

$$\|A_{\theta\theta}^k\| \leq \hat{\gamma}\lambda_S^k, \quad k \geq 0, \quad \theta \in \hat{\mathcal{S}}.$$

Since $\lambda \in (\lambda_S, 1)$, it follows that $\lambda_S < \sqrt{\lambda}$, so from standard linear systems theory there exists a $\sigma > 0$ and $\bar{\gamma} \geq 1$ so that the state transition matrix $\Phi_{\theta^*\hat{\theta}(t)}$ corresponding to $A_{\theta^*\hat{\theta}(t)}$ satisfies

$$\|\Phi_{\theta^*\hat{\theta}(t)}(t_2, t_1)\| \leq \bar{\gamma}\sqrt{\lambda}^{t_2-t_1} \tag{4.24}$$

for $t_2 \geq t_1 \geq t_0$ for which $\|\tilde{\theta}(t+1)\| = \|\hat{\theta}(t+1) - \theta^*\| \leq \sigma$. With $\delta \leq \sigma$ chosen sufficiently small, we now partition the time line of $t \geq \bar{t}_0$ into two parts:

- intervals of the form $\{\underline{t}, \ldots, \bar{t}\}$ satisfying $\|\tilde{\theta}(t)\| < \delta \leq \sigma, t \in \{\underline{t}+1, \ldots, \bar{t}\}$ in which case we can obtain a bound on $\|\phi(t)\|$ in terms of exogenous inputs and $\|\phi(\underline{t})\|$, and
- times $t \geq \bar{t}_0$ for which $\|\tilde{\theta}(t+1)\| \geq \delta$ in which case we obtain a bound on $\|\phi(t)\|$ in terms of the exogenous inputs and $\|\phi(\bar{t}_0)\|$.

Part 1: A bound on $\|\phi(t)\|$ on intervals $\{\underline{t}, \ldots, \bar{t}\}$, $t_0 \leq \underline{t} < \bar{t} < \infty$ for which $\|\tilde{\theta}(t)\| < \delta, t \in \{\underline{t}+1, \ldots, \bar{t}\}$

For intervals of this sort, (4.24) holds. It follows from (4.9) that

$$\|\phi(t)\| \leq \bar{\gamma}\sqrt{\lambda}^{t-\underline{t}}\|\phi(\underline{t})\| + \bar{\gamma}\sum_{i=\underline{t}}^{t-1}\sqrt{\lambda}^{t-i-1}\left(\gamma_1|w(i)| + \gamma_2\|\bar{r}(i+1)\|\right), \quad t \in \{\underline{t}, \ldots, \bar{t}\}. \tag{4.25}$$

Part 2: A bound on $\|\phi(t)\|$ for $\|\tilde{\theta}(t)\| \geq \delta$ and $t \geq \bar{t}_0$

Here we will obtain a bound on $\|\phi(t)\|$ in terms of $\|\phi(\bar{t}_0)\|$ and the exogenous inputs (this differs from Part 1). To construct the bound we analyze (4.10) on the whole interval $\{\bar{t}_0, \ldots, t\}$.

Recall that from the dwell-time constraint (4.20), we know that (4.10) is a stable system. What remains is to find a meaningful bound for (4.21) . This is the goal of the following analysis, which borrows heavily from [41]. First, define the constant:

$$\bar{k} := 1 + \|\mathcal{S}\|.$$

We shall use the following preliminary result; here we use Morse's terminology of a projection operator, which is more commonly termed a characteristic function.

Claim 4.2 *For every fixed* $\bar{t}_0 \geq 0$ *and* $t \geq \bar{t}_0$*, there exists a projection operator* $\psi : \{t \in \mathbf{Z} : t \geq \bar{t}_0\} \to \{0, 1\}$ *that satisfies*

$$\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}(1-\psi(i))|e_{\hat{\theta}(t)}(i+1)|^2} \leq \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|w(i)|^2} + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|$$

and

$$\sum_{i=\bar{t}_0}^{\infty} |\psi(i)| \leq \tau_D - 1. \tag{4.26}$$

Proof: From (3.7), if the supervisor is *not* dwelling at time t (if $\tau(t-1) = 0$), then

$$\sqrt{\sum_{i=t_0}^{t-1} \lambda^{t-i-1}|e_{\hat{\theta}(t)}(i+1)|^2} \leq \sqrt{\sum_{i=t_0}^{t-1} \lambda^{t-i-1}|w(i)|^2}.$$

But this does not hold whenever the supervisor is dwelling. In that case, let $\bar{t}$ be the most recent time when the supervisor was not dwelling. Since the system only dwells for τ_D time-steps, we know that $t - \bar{t} \leq \tau_D - 1$. We introduce a projection operator $\psi : \{t \in \mathbf{Z} : t \geq \bar{t}_0\} \to \{0, 1\}$ defined by

$$\psi(i) = \begin{cases} 1 & i \in \{\bar{t}, \ldots, t-1\} \\ 0 & else; \end{cases}$$

thus (4.26) holds and we obtain:

$$\begin{aligned}
\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}(1-\psi(i))|e_{\hat{\theta}(t)}(i+1)|^2} &= \sqrt{\lambda^{t-\bar{t}} \sum_{i=t_0}^{\bar{t}-1} \lambda^{\bar{t}-i-1}|e_{\hat{\theta}(\bar{t})}(i+1)|^2} \\
&\leq \sqrt{\lambda^{t-\bar{t}} \sum_{i=t_0}^{\bar{t}-1} \lambda^{\bar{t}-i-1}|w(i)|^2} \\
&\leq \sqrt{\lambda^{t-\bar{t}_0} \sum_{i=t_0}^{\bar{t}_0-1} \lambda^{\bar{t}_0-i-1}|w(i)|^2} + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|w(i)|^2} \\
&= \sqrt{\lambda^{t-\bar{t}_0} \begin{bmatrix} \theta^* \\ -1 \end{bmatrix}^\top W(\bar{t}_0) \begin{bmatrix} \theta^* \\ -1 \end{bmatrix}} + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|w(i)|^2} \\
&\leq \bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|w(i)|^2}. \qquad \square
\end{aligned}$$

The next step of the proof is quite unusual, so first consider a special example to help illustrate the motivation. Firstly, suppose that $\hat{\theta}(i)$ is *almost* constant for i in $\{\bar{t}_0+1,\ldots,t\}$. Specifically, let us assume that $\|\hat{\theta}(i)-\hat{\theta}(t)\| \leq \delta$ for all i in $\{\bar{t}_0+1,\ldots,t\}$ (hence, $\|\tilde{\theta}(i)-\tilde{\theta}(t)\| \leq \delta$). Then by using (4.7), Claim 4.2 and Lemma 3.3,

$$
\begin{aligned}
&\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |e_{\hat{\theta}(i+1)}(i+1)| \\
&\quad = \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i) - \phi(i)^\top \left(\tilde{\theta}(t) + \tilde{\theta}(i+1) - \tilde{\theta}(t) \right)| \quad \text{(using (4.7))} \\
&\quad \leq \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left| \phi(i)^\top \left(\tilde{\theta}(i+1) - \tilde{\theta}(t) \right) \right| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i) - \phi(i)^\top \tilde{\theta}(t)| \\
&\quad \leq \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \delta \|\phi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i) - \phi(i)^\top \tilde{\theta}(t)| \quad \left(\text{using } \|\tilde{\theta}(i) - \tilde{\theta}(t)\| \leq \delta \right) \\
&\quad = \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \delta \|\phi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \psi(i) |w(i) - \phi(i)^\top \tilde{\theta}(t)| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} (1-\psi(i)) |e_{\hat{\theta}(t)}(i+1)| \\
&\quad \leq \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\delta + 2\|\hat{\mathcal{S}}\| \psi(i) \right) \|\phi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i)| \\
&\qquad + \sqrt{\frac{1}{1-\lambda} \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} (1-\psi(i)) |e_{\hat{\theta}(t)}(i+1)|^2} \quad \left(\text{using } \|\tilde{\theta}(i)\| \leq 2\|\hat{\mathcal{S}}\| \text{ and Lemma 3.3} \right) \\
&\quad \leq \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\delta + 2\|\hat{\mathcal{S}}\| \psi(i) \right) \|\phi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i)| + \frac{\bar{k}}{\sqrt{1-\lambda}} \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| \\
&\qquad + \frac{1}{\sqrt{1-\lambda}} \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i)|^2} \quad \text{(using Claim 4.2)} \\
&\quad \leq \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\delta + 2\|\hat{\mathcal{S}}\| \psi(i) \right) \|\phi(i)\| + \frac{\bar{k}}{\sqrt{1-\lambda}} \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| \\
&\qquad + \left(1 + \frac{1}{\sqrt{1-\lambda}} \right) \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |w(i)| \quad \text{(using Lemma 3.3)}
\end{aligned}
$$

This forms a nice convolution, except for that first term involving $\delta + 2\|\mathcal{S}\|\psi(i)$. However, due to the finite support of ψ from (4.26), and since δ can be chosen to be arbitrarily small, one can use a perturbation result e.g., Lemma 3.2, to show that the closed-loop system is stable.

In this example, we've been able to approximate each $\tilde{\theta}(i), i \in \{\bar{t}_0 + 1, \ldots, t\}$ by $\tilde{\theta}(t)$, but this generally will not be possible, so we use Lemma 3.1 to produce a similar result in the general case. To do so, we are going to approximate each $\tilde{\theta}(i)$ as a linear combination of a finite set of carefully chosen basis vectors. Since $\tilde{\theta}(i) \in \mathbf{R}^{n+m}$, we need no more than $n+m$ of these basis vectors. The basis vectors will be chosen from the set $\{\tilde{\theta}(\bar{t}_0+1), \ldots, \tilde{\theta}(t)\}$, and in order to make use of Claim 4.2, we require that each $\tilde{\theta}(i)$ be approximated only by vectors from its future. To apply Lemma 3.1 to find a bound for (4.21), we begin by fixing $t \geq \bar{t}_0+1$ such that $\|\tilde{\theta}(t)\| \geq \delta$. Now apply Lemma 3.1 with $\mathcal{X} := \{\tilde{\theta}(t), \tilde{\theta}(t-1), \ldots, \tilde{\theta}(\bar{t}_0+1)\}$ (notice the decreasing order). The lemma gives us a construction of basis vectors, and by reversing their order, they can be expressed as $\left\{\tilde{\theta}(i_1), \tilde{\theta}(i_2), \ldots, \tilde{\theta}(i_{\bar{n}})\right\}$ with $\bar{n} \leq n+m$, and $\bar{t}_0 + 1 \leq i_1 < \ldots < i_{\bar{n}} = t$. The lemma also provides a set of coefficients $\alpha_{(i,j)}$, but we are going to let $g_j(i)$ be a suitably defined shifted version of these coefficients such that they satisfy

$$g_j(i) = 0, \qquad i = i_j + 1, \ldots, t, \quad j = 1, \ldots, \bar{n} - 1, \tag{4.27}$$

$$|g_j(i)| \leq \left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{\bar{n}}, \quad i = 1, \ldots, i_j, \qquad j = 1, \ldots, \bar{n}, \tag{4.28}$$

and such that the approximation error

$$\bar{c}(i) := \tilde{\theta}(i) - \sum_{j=1}^{\bar{n}} g_j(i)\tilde{\theta}(i_j), \quad i = \bar{t}_0 + 1, \ldots, t \tag{4.29}$$

satisfies $\|\bar{c}(i)\| \leq \delta$. Thus, each $\tilde{\theta}(i)$ for $i = \bar{t}_0 + 1, \ldots, t$ is approximated by a linear combination of these basis vectors.

Now we apply (4.29) to (4.7) to express the prediction error in terms of these basis vectors:

$$
\begin{aligned}
e_{\hat{\theta}(i+1)}(i+1) &= w(i) - \phi(i)^\top \tilde{\theta}(i+1) \\
&= w(i) - \phi(i)^\top \sum_{j=1}^{\bar{n}} g_j(i+1)\tilde{\theta}(i_j) - \phi(i)^\top \bar{c}(i+1) \\
&= w(i) + \sum_{j=1}^{\bar{n}} g_j(i+1) \Big(w(i) - \phi(i)^\top \tilde{\theta}(i_j) - w(i) \Big) - \phi(i)^\top \bar{c}(i+1) \\
&= w(i) + \sum_{j=1}^{\bar{n}} g_j(i+1) e_{\hat{\theta}(i_j)}(i+1) - \sum_{j=1}^{\bar{n}} g_j(i+1) w(i) - \phi(i)^\top \bar{c}(i+1) \\
&= \sum_{j=1}^{\bar{n}} g_j(i+1) e_{\hat{\theta}(i_j)}(i+1) + \left(1 - \sum_{j=1}^{\bar{n}} g_j(i+1) \right) w(i) - \phi(i)^\top \bar{c}(i+1), \\
&\qquad\qquad i = \bar{t}_0, ..., t-1.
\end{aligned} \tag{4.30}
$$

This converts the problem of finding a bound for (4.21) to that of finding a bound for

$$
\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |g_j(i+1) e_{\hat{\theta}(i_j)}(i+1)|, \quad j = 1, ..., \bar{n}.
$$

To do so, we make use of Lemma 3.3, Claim 4.2, (4.27) and (4.28). We see that for each $j \in \{1, \ldots, \bar{n}\}$, there exists a projection operator $\psi_j : \{t \in \mathbf{Z} : t \geq \bar{t}_0\} \to \{0, 1\}$ satisfying

(4.26) such that

$$\begin{aligned}
&\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}(1-\psi_j(i))|g_j(i+1)e_{\hat{\theta}(i_j)}(i+1)| \\
&\quad \leq \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \sum_{i=\bar{t}_0}^{i_j-1} \lambda^{t-i-1}(1-\psi_j(i))|e_{\hat{\theta}(i_j)}(i+1)| \quad \text{(using (4.27) and (4.28))} \\
&\quad \leq \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \sqrt{\frac{1}{1-\lambda}\sum_{i=\bar{t}_0}^{i_j-1} \lambda^{i_j-i-1}(1-\psi_j(i))|e_{\hat{\theta}(i_j)}(i+1)|^2} \quad \text{(using Lemma 3.3)} \\
&\quad \leq \frac{1}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \left(\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|w(i)|^2} + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|\right) \\
&\qquad \text{(using Claim 4.2)} \\
&\quad \leq \frac{1}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \left(\sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1}|w(i)| + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|\right) \\
&\qquad \text{(using Lemma 3.3)}\,.
\end{aligned}$$

Now we create a new projection operator $\Psi : \{\bar{t}_0, \ldots, t-1\} \to \{0, 1\}$ whose support is precisely the union of the supports of $\psi_j, j \in \{1, \ldots, \bar{n}\}$:

$$\Psi(i) := 1 - \prod_{j=1}^{\bar{n}} (1-\psi_j(i));$$

since $\bar{n} \leq n+m$, it satisfies

$$\sum_{i=\bar{t}_0}^{t-1} |\Psi(i)| \leq (n+m)(\tau_D - 1), \tag{4.31}$$

and the signal

$$\hat{e}(i+1) := (1-\Psi(i)) \sum_{j=1}^{\bar{n}} g_j(i+1)e_{\hat{\theta}(i_j)}(i+1), \quad i = \bar{t}_0, ..., t-1,$$

satisfies

$$\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |\hat{e}(i+1)|$$
$$\leq \frac{n+m}{\sqrt{1-\lambda}} \left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \left(\sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |w(i)| + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\|\right). \tag{4.32}$$

Then by defining these new signals:

$$\tilde{c}(i) := \Psi(i-1) \sum_{j=1}^{\bar{n}} g_j(i)\tilde{\theta}(i_j), \quad \bar{g}(i) := \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(i)} \end{bmatrix} \left(1 + (\Psi(i-1)-1) \sum_{j=1}^{\bar{n}} g_j(i)\right),$$
$$i = \bar{t}_0 + 1, \ldots, t,$$

we can modify (4.30) to express it as

$$\begin{aligned} e_{\hat{\theta}(i+1)}(i+1) &= \hat{e}(i+1) + \Psi(i) \sum_{j=1}^{\bar{n}} g_j(i+1) e_{\hat{\theta}(i_j)}(i+1) + \left(1 - \sum_{j=1}^{\bar{n}} g_j(i+1)\right) w(i) \\ &\quad - \phi(i)^\top \bar{c}(i+1) \\ &= \hat{e}(i+1) + \Psi(i) \sum_{j=1}^{\bar{n}} g_j(i+1) \left(w(i) - \phi(i)^\top \tilde{\theta}(i_j)\right) \\ &\quad + \left(1 - \sum_{j=1}^{\bar{n}} g_j(i+1)\right) w(i) - \phi(i)^\top \bar{c}(i+1) \\ &= \hat{e}(i+1) - (\tilde{c}(i+1) + \bar{c}(i+1))^\top \phi(i) + \left(1 + (\Psi(i)-1) \sum_{j=1}^{\bar{n}} g_j(i+1)\right) w(i), \end{aligned}$$
$$i = \bar{t}_0, ..., t-1.$$

By substituting this into (4.10), we obtain:

$$\begin{aligned} \phi(i+1) &= \left(A_{\hat{\theta}(i+1)\hat{\theta}(i+1)} - \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(i+1)} \end{bmatrix} (\tilde{c}(i+1) + \bar{c}(i+1))^\top\right) \phi(i) \\ &\quad + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} h_{\hat{\theta}(i+1)} \bar{r}(i+1) + \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(i+1)} \end{bmatrix} \hat{e}(i+1) + \bar{g}(i+1) w(i), \end{aligned}$$
$$i = \bar{t}_0, ..., t-1. \tag{4.33}$$

Recall that $A_{\hat{\theta}(i+1)\hat{\theta}(i+1)}$ is a stable matrix, so this is a stable system subject to a perturbation of the sort considered in Lemma 3.2. If the perturbation is sufficiently small then the perturbed system will also be stable. Using (4.28), (4.31), $\bar{n} \le n+m$ and $\|\tilde{\theta}(\cdot)\| \le 2\|\mathcal{S}\|$, it is clear that

$$\sum_{i=\bar{t}_0}^{t-1} \|\tilde{c}(i+1)\| \le 2\|\hat{\mathcal{S}}\|(n+m)^2(\tau_D - 1)\left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m},$$

which is independent of t. Also, we know that $\|\bar{c}(i)\| \le \delta$, $i = \bar{t}_0 + 1, \dots, t$. Thus,

$$\begin{aligned}
&\sum_{i=i_1}^{i_2-1} \left\| \begin{bmatrix} \bar{b}_n \\ \bar{b}_m g_{\hat{\theta}(i+1)} \end{bmatrix} (\tilde{c}(i+1) + \bar{c}(i+1))^\top \right\| \\
&\qquad \le 2\gamma_1 \|\hat{\mathcal{S}}\|(n+m)^2(\tau_D - 1)\left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} + \delta\gamma_1(i_2 - i_1), \quad \bar{t}_0 \le i_1 < i_2 \le t.
\end{aligned}$$

Hence, this 'perturbation' is small on average. Since $A_{\hat{\theta}(t)\hat{\theta}(t)}$ has stability margin $\tilde{\lambda} < \lambda$, we can apply Lemma 3.2 and it follows that if we fix $\delta \in (0, \sigma]$ such that $\delta < \frac{\lambda - \tilde{\lambda}}{\gamma_1 \hat{\gamma}}$, then (4.33) is a stable system with margin λ. Now define

$$\gamma_3 := \gamma_1 \left(1 + (n+m)\left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}\right),$$

and observe that $\bar{g}(i) \le \gamma_3$. From Lemma 3.2, we conclude that there exists a $\gamma_4 \ge 1$ so that for every $t > \bar{t}_0$ for which $\|\tilde{\theta}(t)\| \ge \delta$, we have

$$\|\phi(t)\| \le \gamma_4 \lambda^{t-\bar{t}_0}\|\phi(\bar{t}_0)\| + \gamma_4 \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}\left(\gamma_3|w(i)| + \gamma_2\|\bar{r}(i+1)\|\right) + \gamma_4\gamma_1 \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|\hat{e}(i+1)|.$$

Now define the constants

$$\begin{aligned}
\gamma_5 &:= \gamma_3 + \gamma_1 \frac{n+m}{\sqrt{1-\lambda}}\left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}, \\
\gamma_6 &:= \bar{k}\gamma_1\gamma_4 \frac{n+m}{\sqrt{1-\lambda}}\left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}.
\end{aligned}$$

Finally, using (4.32) and knowing that $\lambda < \sqrt{\lambda}$, for every $t > \bar{t}_0$ such that $\|\tilde{\theta}(t)\| \geq \delta$ we have

$$\|\phi(t)\| \leq \gamma_4 \sqrt{\lambda}^{t-\bar{t}_0} \|\phi(\bar{t}_0)\| + \gamma_6 \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \gamma_4 \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} (\gamma_5 |w(i)| + \gamma_2 \|\bar{r}(i+1)\|). \tag{4.34}$$

Part 3: A bound on $\|\phi(t)\|$ on the whole interval

We claim that there exists positive constants c_1, c_2, c_3, c_4 such that for any $\bar{t}_0 \geq 0$,

$$\|\phi(t)\| \leq c_1 \sqrt{\lambda}^{t-\bar{t}_0} \|\phi(\bar{t}_0)\| + c_2 \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} (c_3 |w(i)| + c_4 \|\bar{r}(i+1)\|), \quad t \geq \bar{t}_0. \tag{4.35}$$

Case 1: If $\|\tilde{\theta}(i)\| < \delta$ for all $i \in \{\bar{t}_0 + 1, \ldots, t\}$, then using (4.25), clearly (4.35) is satisfied with $c_1 = \bar{\gamma}, c_2 = 0, c_3 = \bar{\gamma}\gamma_1, c_4 = \bar{\gamma}\gamma_2$.

Case 2: If $\|\tilde{\theta}(t)\| \geq \delta$, using (4.34), (4.35) is satisfied with $c_1 = \gamma_4, c_2 = \gamma_6, c_3 = \gamma_4\gamma_5, c_4 = \gamma_4\gamma_2$.

Case 3: If $\|\tilde{\theta}(t)\| < \delta$ and there exists any $\bar{t} \in \{\bar{t}_0 + 1, \ldots, t-1\}$ such that $\|\tilde{\theta}(\bar{t})\| \geq \delta$, then using (4.25) and (4.34) together, one can see that (4.35) is satisfied with $c_1 = \bar{\gamma}\gamma_4, c_2 = \bar{\gamma}\gamma_6, c_3 = \bar{\gamma}\gamma_4\gamma_5 + \bar{\gamma}\gamma_1, c_4 = \bar{\gamma}\gamma_4\gamma_2 + \bar{\gamma}\gamma_2$.

Combining each case and using the knowledge that $\bar{\gamma} \geq 1$ and $\gamma_4 \geq 1$, (4.35) holds in all cases for

$$c_1 = \bar{\gamma}\gamma_4, \quad c_2 = \bar{\gamma}\gamma_6, \quad c_3 = \bar{\gamma}\gamma_4\gamma_5 + \bar{\gamma}\gamma_1, \quad c_4 = \bar{\gamma}\gamma_4\gamma_2 + \bar{\gamma}\gamma_2.$$

Part 4: A bound on $\|W(t)\|$

Observe that (4.35) is a sort of convolution bound for $\phi(t)$, except that it contains an additional term with $W(\bar{t}_0)^{\frac{1}{2}}$. This shows that $\phi(t)$ is not uniformly exponentially stable: that is, when the exogenous inputs r and w are zero, there does not exist some $\gamma \geq 1$ and $\lambda \in (0, 1)$ such that the bound

$$\|\phi(t)\| \leq \gamma \lambda^{t-\bar{t}_0} \|\phi(\bar{t}_0)\|, \quad t \geq \bar{t}_0$$

holds uniformly for any initial condition. This is what motivates the choice to include both $\phi(t)$ and $W(t)^{\frac{1}{2}}$ in $\bar{x}(t)$.

From (3.5), we see that

$$W(t) = \lambda^{t-\bar{t}_0} W(\bar{t}_0) + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \begin{bmatrix} \phi(i) \\ y(i+1) \end{bmatrix} \begin{bmatrix} \phi(i) \\ y(i+1) \end{bmatrix}^\top, \quad t \geq \bar{t}_0.$$

Hence,

$$\begin{aligned}
\|W(t)^{\frac{1}{2}}\| &\leq \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left\| \begin{bmatrix} \phi(i) \\ y(i+1) \end{bmatrix} \right\|^2} \\
&\leq \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\|\phi(i)\| + |y(i+1)|\right)^2} \\
&\leq \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\|\phi(i)\| + \|\theta^*\|\|\phi(i)\| + |w(i)|\right)^2} \\
&\leq \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \bar{k}\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \|\phi(i)\|^2} + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |w(i)|^2} \\
&\leq \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \bar{k}\sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} \|\phi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |w(i)|, \\
&\qquad t \geq \bar{t}_0.
\end{aligned}$$

Thus, $\|W(t)^{\frac{1}{2}}\|$ is bounded by a filtered version of $\phi(t)$ with a pole at $z = \sqrt{\lambda}$, which by (4.35) is itself a convolution with a pole at $z = \sqrt{\lambda}$. Together, this would yield a convolution bound with a double-pole at $z = \sqrt{\lambda}$. Instead, by using the fact that $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, one can find a $\gamma_7 \geq 1$ such that a first order convolution bound holds:

$$\|W(t)^{\frac{1}{2}}\| \leq \gamma_7 \bar{\lambda}^{t-\bar{t}_0} \|\phi(\bar{t}_0)\| + \gamma_7 \bar{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \gamma_7 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} \left(|w(i)| + \|\bar{r}(i+1)\|\right), \quad t > \bar{t}_0.$$

By combining this with (4.35), and using the equivalence of norms:

$$\|W(t)^{\frac{1}{2}}\| \leq \|W(t)^{\frac{1}{2}}\|_F = \left\| \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \right\| \leq \sqrt{n+m+1}\, \|W(t)^{\frac{1}{2}}\|,$$

one can find a γ_8 to create the convolution bound:

$$\|\bar{x}(t)\| \leq \gamma_8 \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0)\| + \gamma_8 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} \left(|w(i)| + \|\bar{r}(i+1)\|\right), \quad t > \bar{t}_0.$$

Finally, recall that $\bar{r}(t)$ is just a vector of delayed copies of $r(t)$:

$$\|\bar{r}(i)\| \leq |r(i+1)| + |r(i+2)| + \ldots + |r(t+m)|,$$

so (4.23) follows. □

4.5 Robustness

Consider now a more complicated scenario: the plant is dependent on the time-varying parameter vector $\theta^*(t)$, subjected to an additive disturbance $w(t)$, and there are some unmodeled dynamics which enter the system via $\bar{w}(t)$:

$$y(t+1) = \phi(t)^\top \theta^*(t) + w(t) + \bar{w}(t). \tag{4.36}$$

We adopt a common model of acceptable time-variations used in adaptive control: with $\zeta > 0$ and $\eta > 0$, we let $s(\mathcal{S}, \zeta, \eta)$ denote the subset of $l_\infty(\mathbf{R}^{n+m})$ whose elements θ^* satisfy $\theta^*(t) \in \mathcal{S}$ for every $t \geq t_0$ as well as

$$\sum_{i=t_1}^{t_2-1} \|\theta^*(i+1) - \theta^*(i)\| \leq \zeta + \eta(t_2 - t_1), \quad t_2 > t_1 \geq t_0.$$

We also adopt a common model of unmodeled dynamics:

$$\begin{aligned} m(t+1) &= \beta m(t) + \beta \|\phi(t)\|, \\ \bar{w}(t) &\leq \mu m(t) + \mu \|\phi(t)\|. \end{aligned} \tag{4.37}$$

As argued in [35], this model subsumes the classical additive uncertainty, multiplicative uncertainty, and uncertainty in a coprime factorization, which is common in the robust control literature, e.g. see [57], and is commonly used in the adaptive control literature, e.g. see [24]. The only limitation is that the perturbations correspond to strictly causal terms.

We will now show that if the time-variations are slow enough and the size of the unmodeled dynamics are small enough, then the closed-loop system retains exponential stability as well as the convolution bound. Furthermore, we need not re-derive the entire stability proof: we can leverage the linear-like nature of the convolution bound proven in Theorem 4.2 to prove this result.

Theorem 4.3 *For every $\zeta > 0$, $\beta \in (0,1)$, $\lambda \in (\lambda_S, 1)$, $\tilde{\lambda} \in (\lambda_S, \lambda)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and τ_D satisfying (4.20), there exists a $\eta > 0$, $\hat{\lambda} \in (max\{\beta, \bar{\lambda}\}, 1)$ and $\gamma' \geq 1$ so that for every $\theta^* \in s(\mathcal{S}, \zeta, \eta)$, $\bar{t}_0 \geq t_0$, $\phi(\bar{t}_0) \in \mathbf{R}^{n+m}$, $W(\bar{t}_0) \in \mathbf{R}^{(n+m+1)\times(n+m+1)}$ positive semidefinite and symmetric, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, $\tau(\bar{t}_0) \in \{0, \dots, \tau_D - 1\}$, and $r, w \in l_\infty$, when the supervisory controller given by (3.4) - (3.6) and (4.6) is applied to the plant (4.36) with $\bar{w}(t)$ satisfying (4.37), the following bound holds:*

$$\left\| \begin{bmatrix} \bar{x}(t) \\ m(t) \end{bmatrix} \right\| \leq \gamma' \hat{\lambda}^{t-\bar{t}_0} \left\| \begin{bmatrix} \bar{x}(\bar{t}_0) \\ m(\bar{t}_0) \end{bmatrix} \right\| + \gamma' \sum_{i=\bar{t}_0}^{t-1} \hat{\lambda}^{t-i-1} |w(i)| + \gamma' \sum_{i=\bar{t}_0+1}^{t+m-1} \hat{\lambda}^{t-i-1} |r(i+1)|,$$
$$t \geq \bar{t}_0.$$

Proof of Theorem 4.3

It is proven in [49] that this robustness property holds for a wide class of 'partially' exponentially stable systems which admit a convolution bound - that is, systems of the form (4.1) whose overall state can be decomposed as

$$x(t) = \begin{bmatrix} \phi(t) \\ z_1(t) \\ z_2(t) \end{bmatrix},$$

with $z_2(t)$ confined to some space $\mathcal{X}$, and with $\begin{bmatrix} \phi(t) \\ z_1(t) \end{bmatrix}$ obeying a convolution bound.

Fix $\lambda \in (\lambda_S, 1)$, $\tilde{\lambda} \in (\lambda_S, \lambda)$, and $\bar{\lambda} \in (\sqrt{\lambda}, 1)$. Then, using the convolution (4.22) from Theorem 4.2, the result follows directly from applying Theorem 1 and Theorem 3 of [49] with[5]

$$
\begin{aligned}
z_1(t) &\Leftarrow \operatorname{vec}\left(W(t)^{\frac{1}{2}}\right), & z_2(t) &\Leftarrow \begin{bmatrix} \hat{\theta}(t) \\ \tau(t) \end{bmatrix}, \\
w(t) &\Leftarrow w(t), & r(t) &\Leftarrow \bar{r}(t), \\
\mathcal{X} &\Leftarrow \hat{\mathcal{S}} \times \{0, \ldots, \tau_D - 1\}. & & \qquad \square
\end{aligned}
$$

4.6 Simulation Results

Now we demonstrate in simulation the results proven in Theorems 4.1 - 4.3. The performance is compared to the adaptive controller constructed using the Projection Algorithm using the setup of [36], as it is the only other adaptive controller which has been shown to exhibit the same linear-like closed-loop behavior. In the first simulation, the controller's ability to asymptotically track an arbitrary reference signal is demonstrated. The set of admissible parameters being considered is the set

$$
\mathcal{S} = \left\{ \begin{bmatrix} a \\ b \end{bmatrix} : \, a \in [1,4], b \in [1,4] \right\},
$$

which satisfies Assumption 4.1. Observe that each model in this set is unstable. Since this is convex, we choose $\hat{\mathcal{S}} = \mathcal{S}$. The plant being considered is

$$
y(t+1) = \underbrace{\begin{bmatrix} 2 \\ 3 \end{bmatrix}}_{\theta^*}^{\top} \underbrace{\begin{bmatrix} y(t) \\ u(t) \end{bmatrix}}_{\phi(t)} + w(t)
$$

with initial condition

$$
\phi(0) = \begin{bmatrix} y(0) \\ u(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
$$

The Supervisory Controller's exponential forgetting factor λ is irrelevant as it has no impact on its performance in the nominal setting. Since the set of admissible plant transfer functions contain no zeros, the minimum dwell-time is $\tau_D = 2$. Both estimators are initialized with the parameter estimate

$$
\hat{\theta}(0) = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix}.
$$

[5] While this does not follow directly from the original version of [49], as it does not allow for $y(t+1)$ nor a general exogenous signal of the form $\bar{r}(t)$ in the controller description, the updated version does so (with minimal changes to the proof).

The Supervisory estimator is also initialized with

$$W(0) = 0, \quad \tau(0) = \tau_D - 1 = 1.$$

The reference to be tracked is

$$r(t) = \sin(0.2\pi t),$$

and the disturbance w is zero. We plot the results in Figure 4.1; with the Supervisory Controller, the tracking error goes to zero in finite time: the parameter estimates converge exactly to θ^* at $t = 4$, so the tracking error is exactly zero at $t = 5$. In contrast, the Projection Algorithm only exhibits asymptotic tracking; in this case the reference signal is sufficiently rich that the parameter estimate too will converge to θ^* asymptotically.

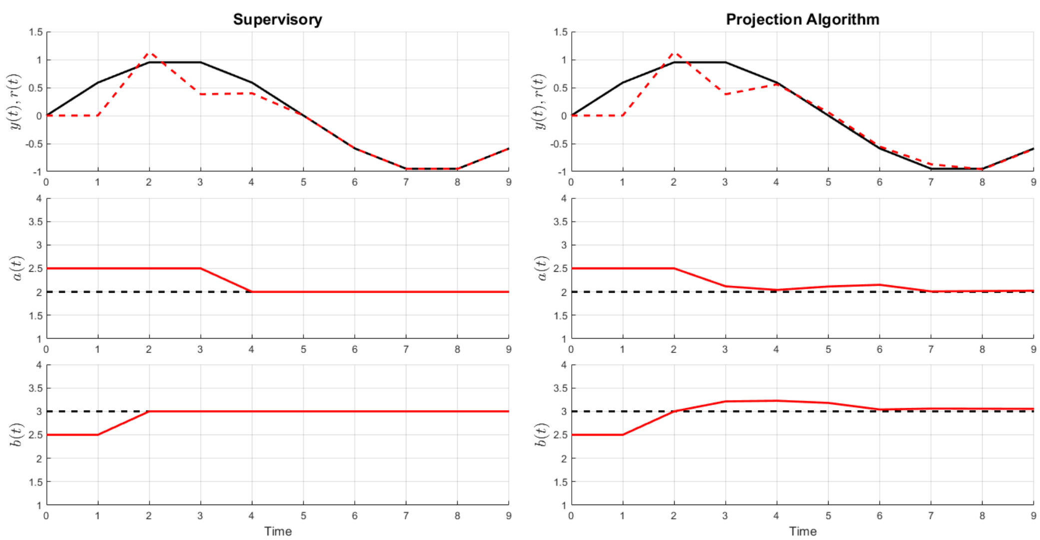

Figure 4.1: Nominal sinusoid tracking performance of Supervisory Control vs the Projection Algorithm; the parameters are dashed and the estimates are solid.

In the next simulation, the setup is precisely the same as above, except that the reference signal is now a constant $r(t) = 1$. We plot the results in Figure 4.2; the result is very similar to above, except this time the parameter estimate generated by the Projection Algorithm never converges to θ^*. Observe that this does not prevent the Projection Algorithm from driving the tracking error to zero. Note that the parameter estimates generated by the Supervisory estimator need not always converge to θ^*, however it is only in special cases where it does not happen, it is very typical for it to converge.

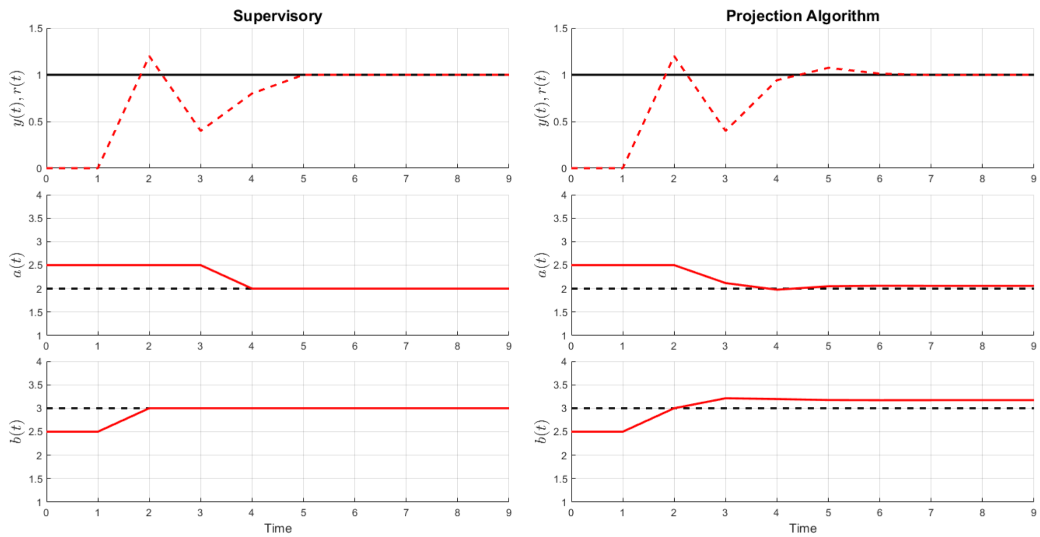

Figure 4.2: Nominal step tracking performance of Supervisory Control vs the Projection Algorithm; the parameters are dashed and the estimates are solid.

In the next simulation, the controllers are compared in the case where there is a random disturbance present and the plant parameters are time-varying. The setup is the same as above except that the reference signal is again a sinusoid:

$$r(t) = \sin(0.2\pi t),$$

the disturbance signal w is a Gaussian random signal with zero mean and standard deviation 0.1, and the plant's parameters are described by

$$a^*(t) = 2.5 + 1.5\cos(0.01\pi t), \quad b^*(t) = 2.5 + 1.5\sin(0.003\pi t).$$

The Supervisory Controller uses exponential forgetting factor $\lambda = 0.6$, which was selected experimentally. We plot the results in Figure 4.3; both systems exhibit stability, but the Supervisory Controller is considerably better: on average, the RMS of its tracking error is approximately 75% that of the Projection Algorithm. Furthermore, by **removing the dwell time**, hence allowing the estimator to change $\hat{\theta}$ at every time-step, the simulated system is still stable and the performance further improves to a RMS tracking error just 66% that of the Projection Algorithm; this has not been plotted. However, it has not been proven that the system is stable without a dwell time.

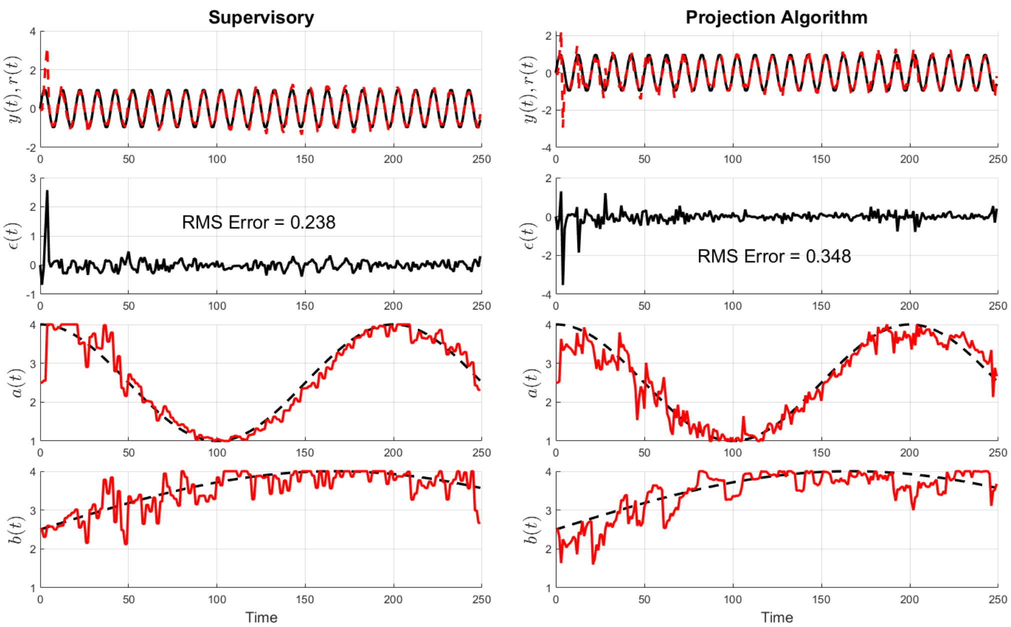

Figure 4.3: Performance in the presence of a random disturbance and time-varying parameters of Supervisory Control vs the Projection Algorithm; the parameters are dashed and the estimates are solid. Supervisory dwell-time is $\tau_D = 2$.

Next, we demonstrate Supervisory Control's ability to tolerate systems with unknown degree, delay, and sign of the high frequency gain. This is done by constructing a **non-convex** $\mathcal{S}$; this is not permitted in the setup of [36], although it is shown in [46, 47] how one can use multiple Projection Algorithm estimators to tolerate a non-convex set, however only partial results are proven, and are limited to the first order case. In this example, we consider a second order plant model of the form

$$y(t+1) = \underbrace{\begin{bmatrix} a_1^*(t) \\ a_2^*(t) \\ b_1^*(t) \\ b_2^*(t) \end{bmatrix}^\top}_{\theta^*(t)} \underbrace{\begin{bmatrix} y(t) \\ y(t-1) \\ u(t) \\ u(t-1) \end{bmatrix}}_{\phi(t)} + w(t).$$

The set of admissible parameters is constructed as the union of several subsets $\mathcal{S}_i$:

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \cup \mathcal{S}_4.$$

Let us choose $\lambda = 0.7$, $\tilde{\lambda} = 0.695$ and $\lambda_S = 0.5$, thus $0 < \lambda_S < \tilde{\lambda} < \lambda < 1$ as required. As per Remark 4.1, we must ensure that $\mathcal{S}$ is chosen such that the zeros of every admissible plant model lies within the open disk of radius λ_S. The first two subsets are defined as

$$\mathcal{S}_1 = \left\{ \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} : \ a_1 \in [-2, 2], a_2 \in [-1, 1], b_1 \in [0.5, 2], \left| \frac{b_2}{b_1} \right| \leq 0.49 \right\}$$

and

$$\mathcal{S}_2 = \left\{ \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} : \ a_1 \in [-2, 2], a_2 \in [-1, 1], b_1 \in [-2, -0.5], \left| \frac{b_2}{b_1} \right| \leq 0.49 \right\}.$$

They describe systems with a delay of one, with a positive or negative high frequency gain, respectively. The zeros of each of these models lies within the *closed* disk of radius 0.49, hence they lie within the *open* disk of radius $\lambda_S = 0.5$. This generally represents a second order system, except the case of $a_2 = b_2 = 0$ represents a first order system. The subsets

$$\mathcal{S}_3 = \left\{ \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} : \ a_1 \in [-2, 2], a_2 \in [-1, 1], b_1 = 0, b_2 \in [0.5, 2] \right\}$$

and

$$\mathcal{S}_4 = \left\{ \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} : \ a_1 \in [-2, 2], a_2 \in [-1, 1], b_1 = 0, b_2 \in [-2, -0.5] \right\}$$

describe second order systems with a delay of two, with no zeros, and with a positive or negative high frequency gain. Since each of these subsets is convex, we can choose $\hat{\mathcal{S}} = \mathcal{S}$, as this satisfies Remark 4.3.

Now consider the minimum dwell time. For any $\theta \in \hat{\mathcal{S}}$ such that $b_1 = 0$ or $b_2 = 0$, the matrix $A_{\theta\theta}$ will be deadbeat; the complexity comes about with a plant model with a zero that is not at the origin, thus we investigate the case where b_1 and b_2 are nonzero. In this case, the control law is

$$u(t) = \underbrace{\begin{bmatrix} -\frac{\hat{a}_2(t)}{\hat{b}_1(t)} & 0 & -\frac{\hat{b}_2(t)}{\hat{b}_1(t)} & 0 \end{bmatrix}}_{f_{\hat{\theta}(t)}} \phi(t-1) \underbrace{-\frac{\hat{a}_1(t)}{\hat{b}_1(t)}}_{g_{\hat{\theta}(t)}} y(t) + \frac{1}{\hat{b}_1(t)} r(t+1),$$

so the matrix of interest becomes

$$A_{\theta\theta} = \begin{bmatrix} a_1 & a_2 & b_1 & b_2 \\ 1 & 0 & 0 & 0 \\ -\frac{a_1^2+a_2}{b_1} & -\frac{a_1 a_2}{b_1} & -a_1 - \frac{b_2}{b_1} & -\frac{a_1 b_2}{b_1} \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \theta \in \mathcal{S}_1 \cup \mathcal{S}_2.$$

As expected, the eigenvalues of this matrix are $0, 0, 0, -\frac{b_2}{b_1}$, so it has stability margin $\lambda_S = 0.5$. Then we find $\hat{\gamma} = 88.5786$ satisfies

$$\|A_{\theta\theta}^k\| \leq \hat{\gamma}\lambda_S^k, \quad k \geq 0, \quad \theta \in \hat{\mathcal{S}}.$$

Hence, the dwell time condition of (4.20) becomes

$$\tau_D \geq \frac{ln(\hat{\gamma})}{ln\left(\frac{\tilde{\lambda}}{\lambda_S}\right)} = \frac{ln(88.5786)}{ln\left(\frac{0.695}{0.5}\right)} \approx 13.6163,$$

thus we choose $\tau_D = 14$. The estimator is initialized with parameter estimate

$$\hat{\theta}(0) = \begin{bmatrix} 1.5 \\ -1 \\ 0 \\ 1 \end{bmatrix}.$$

The reference signal is a sinusoid:

$$r(t) = \sin(0.2\pi t),$$

the disturbance signal w is a Gaussian random signal with zero mean and standard deviation 0.1. In the first simulation, the plant has parameters

$$\begin{aligned} a_1^*(t) &= 1.8, & a_2^*(t) &= -0.85, \\ b_1^*(t) &= 0, & b_2^*(t) &= 1; \end{aligned}$$

it is a second order, unstable system with delay of two and positive high frequency gain. We plot the results in Figure 4.4; the tracking performance is very good.

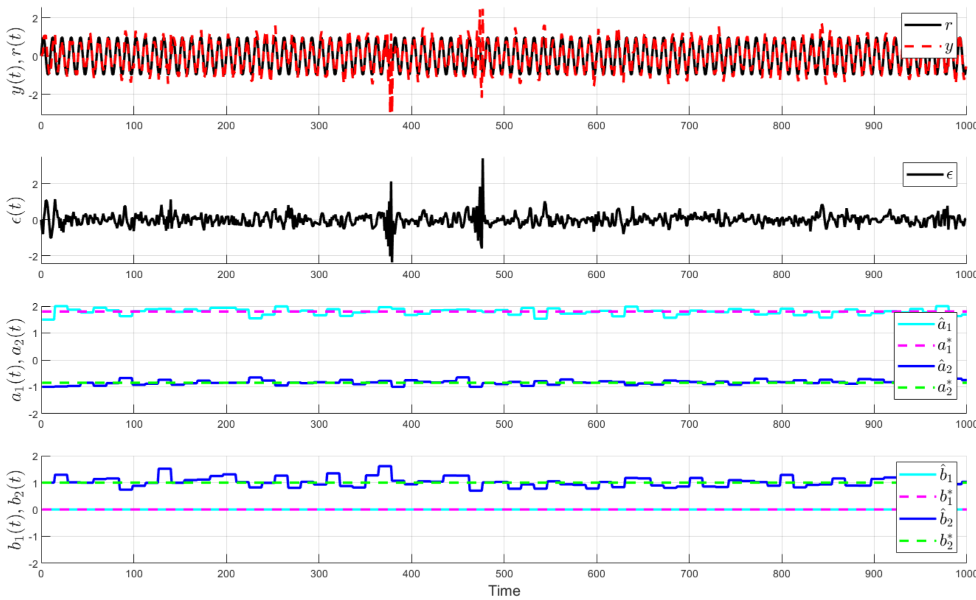

Figure 4.4: Performance of Supervisory Control with non-convex $\mathcal{S}$ and random disturbance.

In the second simulation, the plant has parameters

$$
\begin{aligned}
a_1^*(t) &= 1.8, \quad & a_2^*(t) &= -0.85, \\
b_1^*(t) &= -0.5, \quad & b_2^*(t) &= 0.1;
\end{aligned}
$$

it is a second order, unstable system with delay of one and negative high frequency gain. We plot the results in Figure 4.5; the performance is similar to above, except the initial transient is worse, since the the initial guess $\hat{\theta}(0)$ is further from the true parameters.

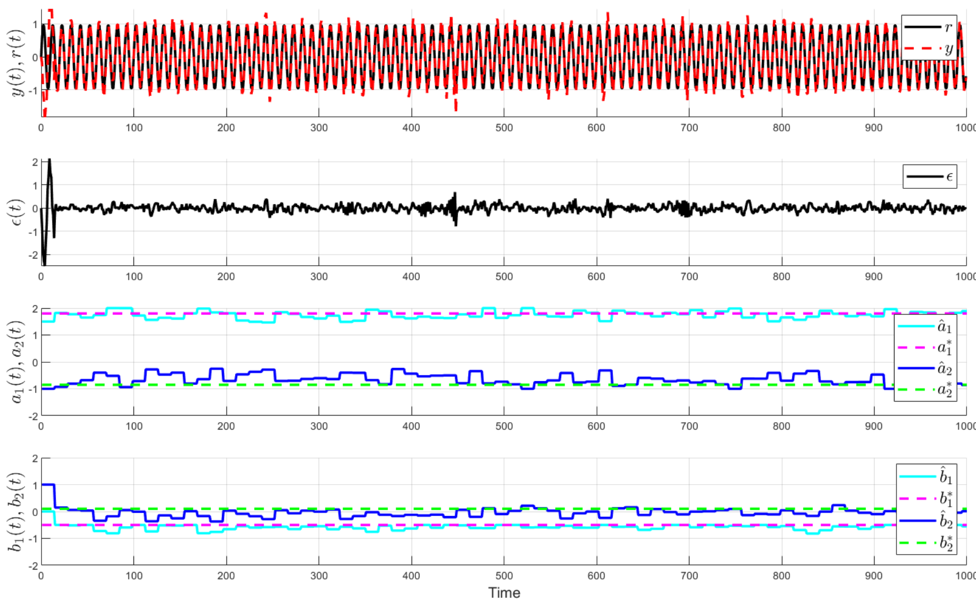

Figure 4.5: Performance of Supervisory Control with non-convex $\mathcal{S}$ and random disturbance.

Finally, we let the plant be time-varying. The plant's parameters are described by

$$\begin{aligned} a_1^*(t) &= -2\sin(0.002\pi t), \quad & a_2^*(t) &= 2\cos(0.002\pi t), \\ b_1^*(t) &= 0, & b_2^*(t) &= 1.25 - 0.75\sin(0.002\pi t); \end{aligned}$$

the plant is a time-varying second order system with delay of two and positive high frequency gain. We plot the results in Figure 4.6; the initial transient is very large because the initial guess $\hat{\theta}(0)$ is destabilizing. It is not of much interest so it has been clipped. The RMS tracking error stated on the plot is calculated using $\epsilon(t)$ for $t \geq 50$ to avoid being biased by the initial transient.

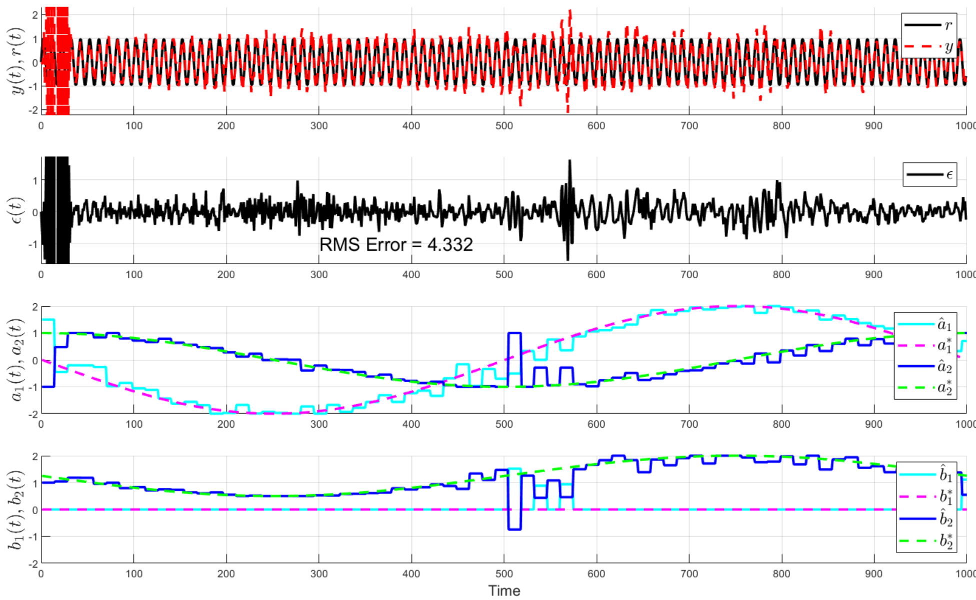

Figure 4.6: Performance of Supervisory Control with non-convex $\mathcal{S}$ and $\tau_D = 16$.

Contrast this with Figure 4.7, which is the same as above except that the dwell time constraint is removed, so $\hat{\theta}$ may change at every time-step. It has not been proven that this system is stable, however the simulated performance is excellent. Observe also that the initial transient is much better than above.

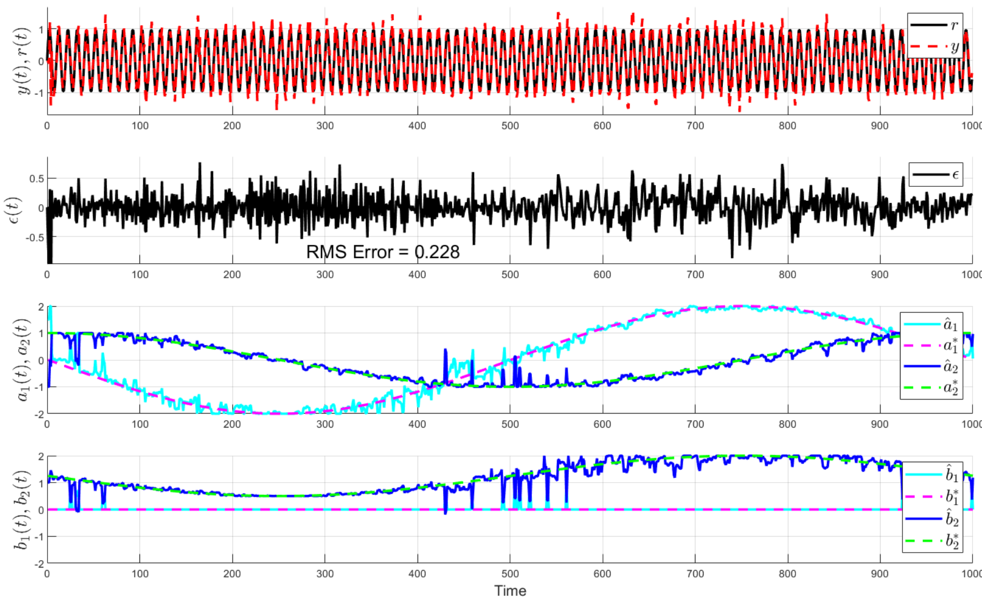

Figure 4.7: Performance of Supervisory Control with non-convex $\mathcal{S}$ and $\tau_D = 1$.

Chapter 5

Pole Placement Step Tracking with Constant Disturbance Rejection

Here we consider the step tracking problem, where the Supervisory parameter estimator defined by (3.4) - (3.6) is combined with an integral pole placement control law, as in the original papers by Morse [40, 41]. Recall that the plant model being considered is

$$y(t+1) = \underbrace{\begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ u(t) \\ \vdots \\ u(t-m+1) \end{bmatrix}^\top}_{\phi(t)} \underbrace{\begin{bmatrix} a_1^* \\ \vdots \\ a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix}}_{\theta^*} + w(t), \quad t \geq t_0. \tag{5.1}$$

The goal is to make the plant's output y track a constant reference signal r in the presence of a constant disturbance, hence to make the **tracking error** $\epsilon(t) := y(t) - r$ exponentially go to zero. This is done via the internal model principle: the plant is augmented with an integrator $\frac{z}{z-1}$, and a pole placement controller is created via certainty equivalence to stabilize the augmented plant. In this chapter, we will assume that the reference signal is constant; we shall allow the disturbance w to be time-varying since that will be leveraged later to prove robustness to time-varying parameters.

For an ordinary LTI control problem, introducing an integrator into the controller is sufficient to eliminate the influence of a constant disturbance and track a constant reference. However, if it is not filtered out, a constant disturbance will impact the parameter estimator, and thus the closed-loop performance. Hence, we use the data filter $H(z) = \frac{z-1}{z}$. The overall closed-loop system is illustrated in Figure 5.1. The prediction error (3.2) is now

$$\begin{aligned} e_{\hat{\theta}(t)}(i) &= \{h \circ y\}(i) - (\{h \circ \phi\}(i-1))^\top \hat{\theta}(t) \\ &= \Delta y(i) - \Delta\phi(i-1)^\top \hat{\theta}(t), \quad t \geq t_0, \quad i > t_0. \end{aligned}$$

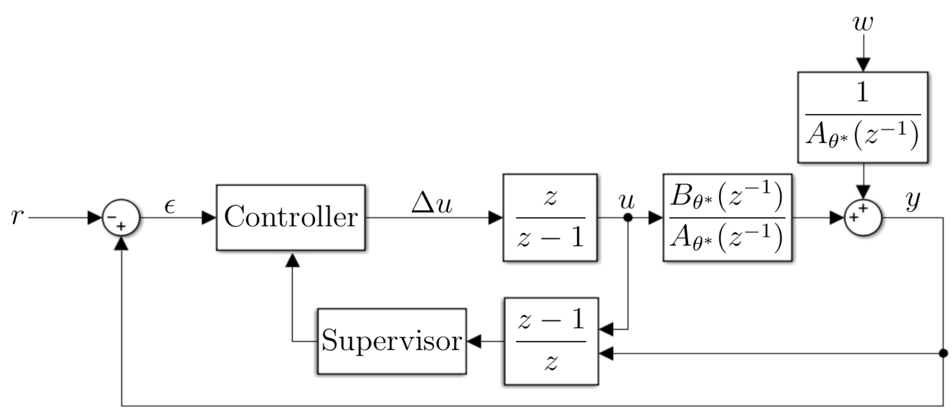

Figure 5.1: Block diagram of the estimator with pole placement controller and integrator.

Observe that the augmented plant, with input Δu and output ϵ, has the transfer function

$$\begin{aligned} &\frac{1}{1-z^{-1}} \times \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-m}}{1 - a_1^* z^{-1} - \ldots - a_n^* z^{-n}} \\ &\quad = \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-m}}{1 - (1+a_1^*)z^{-1} - (a_2^* - a_1^*)z^{-2} - \ldots - (a_n^* - a_{n-1}^*)z^{-n} + a_n z^{-n-1}}. \end{aligned}$$

Thus, by adopting the change of variables

$$\varphi(t) := \begin{bmatrix} \epsilon(t) \\ \vdots \\ \epsilon(t-n) \\ \Delta u(t) \\ \vdots \\ \Delta u(t-m+1) \end{bmatrix} \in \mathbf{R}^{n+m+1}, \quad \vartheta^* := \begin{bmatrix} 1 + a_1^* \\ a_2^* - a_1^* \\ \vdots \\ a_n^* - a_{n-1}^* \\ -a_n^* \\ b_1^* \\ \vdots \\ b_m^* \end{bmatrix} \in \mathbf{R}^{n+m+1},$$

the plant may be expressed as

$$\epsilon(t+1) = \varphi(t)^\top \vartheta^* + \Delta w(t), \quad t \geq t_0. \tag{5.2}$$

Similarly, we define a transformed version of the estimated parameter vector at time t as

$$\hat{\vartheta}(t) := \begin{bmatrix} 1 + \hat{a}_1(t) \\ \hat{a}_2(t) - \hat{a}_1(t) \\ \vdots \\ \hat{a}_n(t) - \hat{a}_{n-1}(t) \\ -\hat{a}_n(t) \\ \hat{b}_1(t) \\ \vdots \\ \hat{b}_m(t) \end{bmatrix} \in \mathbf{R}^{n+m+1}, \quad t \geq t_0.$$

As in the previous chapter, we define the **parameter estimation error** as

$$\tilde{\theta}(t) := \hat{\theta}(t) - \theta^* = \begin{bmatrix} \hat{a}_1(t) - a_1^* \\ \vdots \\ \hat{a}_n(t) - a_n^* \\ \hat{b}_1(t) - b_1^* \\ \vdots \\ \hat{b}_m(t) - b_m^* \end{bmatrix} \in \mathbf{R}^{n+m}, \quad t \geq t_0.$$

Next, define the transformed version of the parameter estimation error as $\tilde{\vartheta}(t) = \hat{\vartheta}(t) - \vartheta^*$:

$$\tilde{\vartheta}(t) = \begin{bmatrix} \hat{a}_1(t) - a_1^* \\ \hat{a}_2(t) - a_2^* - \hat{a}_1(t) + a_1^* \\ \vdots \\ \hat{a}_n(t) - a_n^* - \hat{a}_{n-1}(t) + a_{n-1}^* \\ -\hat{a}_n(t) + a_n^* \\ \hat{b}_1(t) - b_1^* \\ \vdots \\ \hat{b}_m(t) - b_m^* \end{bmatrix} \in \mathbf{R}^{n+m+1},$$

which is just a linear transformation away from $\tilde{\theta}(t)$. Specifically, this linear map is

$$R := \begin{bmatrix} 1 & & & & & & \\ -1 & 1 & & & & & \\ & \ddots & \ddots & & & & \\ & & -1 & 1 & & & \\ & & & -1 & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \in \mathbf{R}^{(n+m+1)\times(n+m)},$$

so

$$\tilde{\vartheta}(t) = R\tilde{\theta}(t).$$

This allows us to express the prediction error in another form which shall be useful:

$$\begin{aligned}
e_{\hat{\theta}(t)}(i) &= \Delta y(i) - \Delta\phi(i-1)^\top \hat{\theta}(t) \\
&= \Delta\epsilon(i) - \begin{bmatrix} \epsilon(i-1) - \epsilon(i-2) \\ \vdots \\ \epsilon(i-n) - \epsilon(i-n-1) \\ \Delta u(i-1) \\ \vdots \\ \Delta u(i-m) \end{bmatrix}^\top \begin{bmatrix} \hat{a}_1(t) \\ \vdots \\ \hat{a}_n(t) \\ \hat{b}_1(t) \\ \vdots \\ \hat{b}_m(t) \end{bmatrix} \qquad \begin{array}{l} (\text{using } \Delta y(i) = \\ \Delta y(i) - r + r = \Delta\epsilon(i)) \end{array} \\
&= \epsilon(i) - \epsilon(i-1) - \begin{bmatrix} \epsilon(i-1) \\ \vdots \\ \epsilon(i-n) \\ 0 \\ \Delta u(i-1) \\ \vdots \\ \Delta u(i-m) \end{bmatrix}^\top \begin{bmatrix} \hat{a}_1(t) \\ \vdots \\ \hat{a}_n(t) \\ 0 \\ \hat{b}_1(t) \\ \vdots \\ \hat{b}_m(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon(i-2) \\ \vdots \\ \epsilon(i-n-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}^\top \begin{bmatrix} 0 \\ \hat{a}_1(t) \\ \vdots \\ \hat{a}_n(t) \\ \hat{b}_1(t) \\ \vdots \\ \hat{b}_m(t) \end{bmatrix} \\
&= \epsilon(i) - \varphi(i-1)^\top \hat{\vartheta}(t) \qquad (5.3) \\
&= \varphi(i-1)^\top \vartheta^* + \Delta w(i-1) - \varphi(i-1)^\top \hat{\vartheta}(t) \quad (\text{using } (5.2)) \\
&= \Delta w(i-1) - \varphi(i-1)^\top R\tilde{\theta}(t), \quad t \geq t_0, \quad i > t_0. \qquad (5.4)
\end{aligned}$$

Beside Assumptions 3.1 - 3.3, which are assumed to hold, we will enforce some additional assumptions on the set of admissible parameters $\mathcal{S}$ to ensure that the pole placement control law is well behaved:

Assumption 5.1 *The polynomials $A_\theta(z^{-1})$ and $B_\theta(z^{-1})$ corresponding to each θ in $\mathcal{S}$ are coprime.*

Assumption 5.2 *The polynomial $B_\theta(z^{-1})$ corresponding to each θ in $\mathcal{S}$ is nonzero at $z = 1$.*

Assumption 5.3 *For each $\bar{n} \in \{0, \ldots, n\}$ and $\bar{m} \in \{1, \ldots, m\}$, the set of all $\theta \in \mathcal{S}$ with $A_\theta(z^{-1})$ of degree $\bar{n}$ and $B_\theta(z^{-1})$ of degree $\bar{m}$ is compact.*

Remark 5.1 *Assumption 5.1 ensures that a controller can be constructed to place all the closed-loop poles at will.*

Remark 5.2 *Assumption 5.2 ensures that the plant does not have a zero at $z = 1$, which is necessary to achieve step tracking using a bounded input.*

Remark 5.3 *Assumption 5.3 is used to ensure that the pole placement control law is uniformly bounded for all admissible plant models.*

Remark 5.4 *The components $\mathcal{S}_i$ that make up $\hat{\mathcal{S}}$ can be chosen such that each element of $\mathcal{S}_i$ satisfies Assumptions 5.1 - 5.3, so henceforth we will assume that this is the case.*

5.1 Control Law

Recall that the plant model corresponds to a system with transfer function

$$\frac{B_{\theta^*}(z^{-1})}{A_{\theta^*}(z^{-1})} = \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-m}}{1 - a_1^* z^{-1} - \ldots - a_n^* z^{-n}}.$$

Recall from Remark 3.2 that we allow some of these coefficients to be zero, thus let $\bar{n} \in \{0, \ldots, n\}$ be the degree (in z^{-1}) of the polynomial $A_{\theta^*}(z^{-1})$ and let $\bar{m} \in \{1, \ldots, m\}$ be the degree of $B_{\theta^*}(z^{-1})$. Also recall that we desire a pole placement controller with an integrator, so we augment the plant with an integrator and design a pole placement controller to stabilize it. The augmented plant has transfer function

$$\frac{1}{1-z^{-1}} \times \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-\bar{m}}}{1 - a_1^* z^{-1} - \ldots - a_n^* z^{-\bar{n}}}$$
$$= \frac{b_1^* z^{-1} + \ldots + b_m^* z^{-\bar{m}}}{1 - (1+a_1^*)z^{-1} - (a_2^* - a_1^*)z^{-2} - \ldots - (a_{\bar{n}}^* - a_{\bar{n}-1}^*)z^{-\bar{n}} + a_n z^{-\bar{n}-1}}.$$

The denominator of this system has degree $\bar{n}+1$ and the numerator has degree $\bar{m}$. Let

$$A_\theta^*(z^{-1}) := 1 + \alpha_1 z^{-1} + \ldots + \alpha_{\bar{n}+\bar{m}+1} z^{-\bar{n}-\bar{m}-1}$$

be a Schur polynomial of degree at most $\bar{n}+\bar{m}+1$ (trailing coefficients may be zero) which is the desired characteristic polynomial. Next, for each $\theta \in \hat{\mathcal{S}}$, let the polynomials

$$L_\theta(z^{-1}) := 1 - l_{\theta 1} z^{-1} - \ldots - l_{\theta \bar{m}} z^{-\bar{m}}$$

and

$$P_\theta(z^{-1}) := -p_{\theta 1} z^{-1} - \ldots - p_{\theta \bar{n}+1} z^{-\bar{n}-1}$$

satisfy the equation

$$(1-z^{-1})A_\theta(z^{-1})L_\theta(z^{-1}) + B_\theta(z^{-1})P_\theta(z^{-1}) = A^*(z^{-1}). \tag{5.5}$$

To express this polynomial equation in matrix form, let $M \in \mathbf{R}^{(\bar{n}+\bar{m}+1)\times(\bar{n}+\bar{m}+1)}$ be the matrix whose leftmost column is

$$\left.\begin{bmatrix} 1 \\ -1-a_1 \\ a_1 - a_2 \\ \vdots \\ a_{\bar{n}-1} - a_{\bar{n}} \\ a_{\bar{n}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \end{matrix}\right\} \bar{n}+1 \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \bar{m}-1 \end{matrix}\right. \in \mathbf{R}^{\bar{n}+\bar{m}+1},$$

the $(\bar{m}+1)^{\text{th}}$ column is

$$\left.\begin{bmatrix} 0 \\ b_1 \\ \vdots \\ b_{\bar{m}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} b_1 \\ \vdots \\ b_{\bar{m}} \end{matrix}}\right\} \bar{m} \\ \left.\vphantom{\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}}\right\} \bar{n} \end{matrix} \right. \in \mathbf{R}^{\bar{n}+\bar{m}+1},$$

and each of the remaining columns is the column to their left, shifted down one row, and padded with a zero. In this fashion, M is precisely the transpose of the Sylvester matrix of the polynomials $(1-z^{-1})A_\theta(z^{-1})$ and $B_\theta(z^{-1})$. Then (5.5) can be equivalently expressed as

$$M \begin{bmatrix} l_1 \\ \vdots \\ l_{\bar{m}} \\ p_1 \\ \vdots \\ p_{\bar{n}+1} \end{bmatrix} = - \begin{bmatrix} \alpha_1 + 1 + a_1 \\ \alpha_2 + a_2 - a_1 \\ \vdots \\ \alpha_{\bar{n}} + a_{\bar{n}} - a_{\bar{n}-1} \\ \alpha_{\bar{n}+1} - a_{\bar{n}} \\ \alpha_{\bar{n}+2} \\ \vdots \\ \alpha_{\bar{n}+\bar{m}+1} \end{bmatrix}.$$

A Sylvester matrix is invertible if and only if its two polynomials are coprime [2, Section 7.2]. Assumptions 5.1 and 5.2 ensure that $(1-z^{-1})A_\theta(z^{-1})$ and $B_\theta(z^{-1})$ are coprime, therefore M is nonsingular, and a unique solution exists. In addition, due to Assumption 5.3, and since this is an analytic function of $\theta \in \hat{\mathcal{S}}$ such that $A_{\theta^*}(z^{-1})$ and $B_{\theta^*}(z^{-1})$ are of degree $\bar{n}$ and $\bar{m}$, respectively, it follows that the coefficients of $L_\theta(z^{-1})$ and $P_\theta(z^{-1})$ are uniformly bounded for $\theta \in \hat{\mathcal{S}}$. Finally, using the parameter estimate $\hat{\theta}(t)$, the control law is [1]

$$\Delta u(t) = \begin{bmatrix} p_{\hat{\theta}(t)1} \\ \vdots \\ p_{\hat{\theta}(t)\bar{n}+1} \\ l_{\hat{\theta}(t)1} \\ \vdots \\ l_{\hat{\theta}(t)\bar{m}} \end{bmatrix}^\top \begin{bmatrix} \epsilon(t-1) \\ \vdots \\ \epsilon(t-\bar{n}-1) \\ \Delta u(t-1) \\ \vdots \\ \Delta u(t-\bar{m}) \end{bmatrix}, \quad t \geq t_0 + 1.$$

[1] $u(t_0)$ is given as part of the initial condition $\phi(t_0)$, so the control law only holds for $t \geq t_0 + 1$.

Notice that since $\bar{n} \le n$ and $\bar{m} \le m$, all the elements of the rightmost vector are elements of $\varphi(t-1)$. Thus, by padding the left vector with some extra zeros:

$$f_\theta := \begin{bmatrix} p_{\theta 1} & \dots & p_{\theta \bar{n}+1} & \overbrace{0 \quad \dots \quad 0}^{n-\bar{n}} & l_{\theta 1} & \dots & l_{\theta \bar{m}} & \overbrace{0 \quad \dots \quad 0}^{m-\bar{m}} \end{bmatrix},$$

we can express the control law as

$$\Delta u(t) = f_{\hat{\theta}(t)} \varphi(t-1), \quad t \ge t_0 + 1. \tag{5.6}$$

5.2 State Space Representation

We can now form an update equation for $\varphi(t)$. Start by defining, for every $k \in \mathbf{N}$,

$$\bar{A}_k := \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbf{R}^{k \times k}, \quad \bar{b}_k := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^k;$$

then begin the state-space construction with

$$\varphi(t+1) = \begin{bmatrix} \bar{A}_{n+1} & 0 \\ 0 & \bar{A}_m \end{bmatrix} \varphi(t) + \begin{bmatrix} \bar{b}_{n+1} \\ 0 \end{bmatrix} \epsilon(t+1) + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} \Delta u(t+1), \quad t \ge t_0.$$

Now incorporate the control law (5.6):

$$\varphi(t+1) = \left(\begin{bmatrix} \bar{A}_{n+1} & 0 \\ 0 & \bar{A}_m \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} f_{\hat{\theta}(t+1)} \right) \varphi(t) + \begin{bmatrix} \bar{b}_{n+1} \\ 0 \end{bmatrix} \epsilon(t+1), \quad t \ge t_0. \tag{5.7}$$

Finally, substitute in the plant dynamics from (5.2): with [2]

$$A_{\theta_1 \theta_2} := \begin{bmatrix} \bar{A}_{n+1} & 0 \\ 0 & \bar{A}_m \end{bmatrix} + \begin{bmatrix} \bar{b}_{n+1} \\ 0 \end{bmatrix} {\vartheta_1}^\top + \begin{bmatrix} 0 \\ \bar{b}_m \end{bmatrix} f_{\theta_2},$$

we end up with

$$\varphi(t+1) = A_{\theta^* \hat{\theta}(t+1)} \varphi(t) + \begin{bmatrix} \bar{b}_{n+1} \\ 0 \end{bmatrix} \Delta w(t), \quad t \ge t_0. \tag{5.8}$$

[2] Here ϑ_1 is dependent on θ_1.

Alternatively, from (5.3), substitute $\epsilon(t+1) = e_{\hat{\theta}(t+1)}(t+1) + \hat{\vartheta}(t+1)^\top \varphi(t)$ into (5.7) to yield another representation:

$$\varphi(t+1) = A_{\hat{\theta}(t+1)\hat{\theta}(t+1)}\varphi(t) + \begin{bmatrix} \bar{b}_{n+1} \\ 0 \end{bmatrix} e_{\hat{\theta}(t+1)}(t+1), \quad t \geq t_0. \tag{5.9}$$

This form is useful because $A_{\theta\theta}$ represents the closed-loop system dynamics if $\hat{\theta}(\cdot) = \theta^* = \theta \in \hat{\mathcal{S}}$. This matrix has $n+m+1$ eigenvalues. Due to the use of the certainty equivalence pole placement control law, $\bar{n} + \bar{m} + 1$ of these eigenvalues correspond to the roots of $A^*(z^{-1})$, and the remaining $n + m - \bar{n} - \bar{m}$ lie at the origin. Since $A^*(z^{-1})$ is Schur, there exists a $\lambda_S \in (0,1)$ such that all the roots of $A^*(z^{-1})$ lie within the open disc of radius λ_S. This, plus the compactness of $\mathcal{S}$ and the uniform boundedness of the control law coefficients ensures that there exists a $\hat{\gamma}$ such that

$$\|A_{\theta\theta}^k\| \leq \hat{\gamma}\lambda_S^k, \quad k \geq 0, \quad \theta \in \hat{\mathcal{S}}. \tag{5.10}$$

Thus if $\hat{\theta}$ were fixed, (5.9) becomes a stable system with input $e_{\hat{\theta}}$.

5.3 Closed-loop Stability

Before presenting the main result on stability, we return to the issue of dwell time. For any fixed $\theta \in \hat{\mathcal{S}}$, the control law ensures that $A_{\theta\theta}$ is a stable matrix; its eigenvalues determined by $A^*(z^{-1})$. Now we must impose a dwell-time constraint to ensure that the time-varying $A_{\hat{\theta}(t)\hat{\theta}(t)}$ is stable. Generally, we would require that the dwell time τ_D satisfy a requirement such as (4.20) in the d-step-ahead case. Instead, we are going to choose $A^*(z^{-1}) = 1$, so all the eigenvalues of $A_{\theta\theta}$ lie at the origin; thus

$$\|A_{\theta\theta}^{n+m+1}\| = 0, \quad \theta \in \hat{\mathcal{S}}.$$

In doing so, choosing τ_D to be $n+m+1$ is sufficient. Then, if Φ is the state transition matrix of $A_{\hat{\theta}(t)\hat{\theta}(t)}$, it shall have finite support, so for any choice $\tilde{\lambda} \in (0, \lambda)$, there exists a $\hat{\gamma} \geq 1$ such that

$$\|\Phi(t_2, t_1)\| \leq \hat{\gamma}\tilde{\lambda}^{t_2 - t_1}, \quad t_2 \geq t_1 \geq t_0.$$

Therefore, if $\hat{\theta}(t)$ is any piecewise-constant signal with dwell time at least $n+m+1$, then the system (5.9) is exponentially stable, which reveals the convolution bound, albeit in terms of internal signals rather than exogenous ones: with $\bar{t}_0 \geq t_0$,

$$\|\varphi(t)\| \leq \hat{\gamma}\tilde{\lambda}^{t-\bar{t}_0}\|\varphi(\bar{t}_0)\| + \hat{\gamma}\sum_{i=\bar{t}_0}^{t-1}\tilde{\lambda}^{t-i-1}|e_{\hat{\theta}(i+1)}(i+1)|.$$

Thus, as in Section 4.4, a stability proof relies on finding a meaningful bound on the term

$$\sum_{i=\bar{t}_0}^{t-1} \tilde{\lambda}^{t-i-1} |e_{\hat{\theta}(i+1)}(i+1)|. \tag{5.11}$$

However, this time the prediction error definition is subtly different due to the data filter.

Now we present the main stability proof of this chapter. First notice that the overall system state is uniquely defined by $\varphi(t)$, $W(t)$, $\hat{\theta}(t)$ and $\tau(t)$. Since $W(t)$ is a matrix, we vectorize it so that the overall system state vector may be expressed as

$$x(t) := \begin{bmatrix} \varphi(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \\ \hat{\theta}(t) \\ \tau(t) \end{bmatrix}.$$

Now split the state up into components $\bar{x}(t)$ and $\tilde{x}(t)$:

$$\bar{x}(t) := \begin{bmatrix} \varphi(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}, \quad \tilde{x}(t) := \begin{bmatrix} \hat{\theta}(t) \\ \tau(t) \end{bmatrix}.$$

The following theorem shows that $\bar{x}(t)$ is exponentially stable uniformly in $\tilde{x}(t)$, and admits a convolution bound.

Theorem 5.1 *For every $\lambda \in (0,1)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and with $A^*(z^{-1}) = 1$ and $\tau_D = n+m+1$, there exists a $\gamma \geq 1$ so that for every $\theta^* \in \mathcal{S}$, $\bar{t}_0 \geq t_0$, $\varphi(\bar{t}_0) \in \mathbf{R}^{n+m+1}$, $W(\bar{t}_0) \in \mathbf{R}^{(n+m+1)\times(n+m+1)}$ positive semidefinite and symmetric, $\tau(\bar{t}_0) \in \{0, \ldots, n+m\}$, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, $r \in \mathbf{R}$, and $w \in l_\infty$, when the supervisory controller given by (3.4) - (3.6) and (5.6) is applied to the plant (5.1), the following bound holds:*

$$\|\bar{x}(t)\| \leq \gamma \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0)\| + \gamma \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |\Delta w(i)|, \quad t \geq \bar{t}_0. \tag{5.12}$$

Remark 5.5 *If w is actually a constant signal, then the second term in (5.12) disappears, and $\bar{x}(t)$ (hence, $\epsilon(t)$) goes to zero in an exponential fashion.*

Proof of Theorem 5.1

Fix $\lambda \in (0,1)$ and $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and let $\theta^* \in \mathcal{S}$, $\bar{t}_0 \geq t_0$, $\varphi(\bar{t}_0) \in \mathbf{R}^{n+m+1}$, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, $W(\bar{t}_0)$ positive semidefinite and symmetric, $\tau(\bar{t}_0) \in \{0, \ldots, n+m\}$, $r \in \mathbf{R}$ and $w \in l_\infty$ be arbitrary.

To prove this, we split up time in the same fashion as Section 4.4 into those for which $\|\tilde{\theta}(t)\|$ is small and those for which it is not. Before proceeding, recall from (5.10) that $A_{\theta\theta}$ is a stable matrix. Specifically, with $A^*(z^{-1}) = 1$, it is deadbeat. So from standard linear systems theory there exists a $\sigma > 0$ and $\bar{\gamma} \geq 1$ so that the state transition matrix $\Phi_{\theta^*\hat{\theta}(t)}$ corresponding to $A_{\theta^*\hat{\theta}(t)}$ satisfies

$$\|\Phi_{\theta^*\hat{\theta}(t)}(t_2, t_1)\| \leq \bar{\gamma}\sqrt{\lambda}^{t_2 - t_1} \tag{5.13}$$

for $t_2 \geq t_1 \geq t_0$ for which $\|\tilde{\theta}(t+1)\| = \|\hat{\theta}(t+1) - \theta^*\| \leq \sigma$. With $\delta \leq \sigma$ chosen sufficiently small, we now partition the time line of $t \geq \bar{t}_0$ into two parts:

- intervals of the form $\{\underline{t}, \ldots, \bar{t}\}$ satisfying $\|\tilde{\theta}(t)\| < \delta \leq \sigma, t \in \{\underline{t}+1, \ldots, \bar{t}\}$ in which case we can obtain a bound on $\|\varphi(t)\|$ in terms of exogenous inputs and $\|\varphi(\underline{t})\|$, and
- times $t \geq \bar{t}_0$ for which $\|\tilde{\theta}(t+1)\| \geq \delta$ in which case we obtain a bound on $\|\varphi(t)\|$ in terms of the exogenous inputs and $\|\varphi(\bar{t}_0)\|$.

Part 1: A bound on $\|\varphi(t)\|$ on intervals $\{\underline{t}, \ldots, \bar{t}\}$, $t_0 \leq \underline{t} < \bar{t} < \infty$ for which $\|\tilde{\theta}(t)\| < \delta, t \in \{\underline{t}+1, \ldots, \bar{t}\}$

For intervals of this sort, (5.13) holds. It follows from (5.8) that

$$\|\varphi(t)\| \leq \bar{\gamma}\sqrt{\lambda}^{t-\underline{t}}\|\varphi(\underline{t})\| + \bar{\gamma}\sum_{i=\underline{t}}^{t-1}\sqrt{\lambda}^{t-i-1}|\Delta w(i)|, \quad t \in \{\underline{t}, \ldots, \bar{t}\}\,. \tag{5.14}$$

Part 2: A bound on $\|\varphi(t)\|$ for $\|\tilde{\theta}(t)\| \geq \delta$ and $t \geq \bar{t}_0$

Here we will obtain a bound on $\|\varphi(t)\|$ in terms of $\|\varphi(\bar{t}_0)\|$ and the exogenous inputs (this differs from Part 1). To construct the bound we analyze (5.9) on the whole interval

$\{\bar{t}_0, \ldots, t\}$. We now aim to find a bound for (5.11) in a very similar fashion as seen in Section 4.4 by making use of Lemma 3.1. But first we must define this constant

$$\bar{k} := 1 + \|\mathcal{S}\|.$$

We shall use the following preliminary result; here we use Morse's terminology of a projection operator, which is more commonly termed a characteristic function.

Claim 5.1 *For every fixed $\bar{t}_0 \geq 0$ and $t \geq \bar{t}_0$, there exists a projection operator $\psi : \{t \in \mathbf{Z} : t \geq \bar{t}_0\} \to \{0, 1\}$ that satisfies*

$$\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}(1-\psi(i))|e_{\hat{\theta}(t)}(i+1)|^2} \leq \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|\Delta w(i)|^2} + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|$$

and

$$\sum_{i=\bar{t}_0}^{\infty} |\psi(i)| \leq n + m. \tag{5.15}$$

Proof: Same as for Claim 4.2, with $\tau_D = n + m + 1$. □

Now we apply Lemma 3.1 to find a bound for (5.11). We begin by fixing $t \geq \bar{t}_0 + 1$ such that $\|\tilde{\theta}(t)\| \geq \delta$. Now apply Lemma 3.1 with $\mathcal{X} := \{\tilde{\theta}(t), \tilde{\theta}(t-1), \ldots, \tilde{\theta}(\bar{t}_0 + 1)\}$ (notice the decreasing order). The lemma gives us a construction of basis vectors, and by reversing their order, they can be expressed as $\left\{\tilde{\theta}(i_1), \tilde{\theta}(i_2), \ldots, \tilde{\theta}(i_{\bar{n}})\right\}$ with $\bar{n} \leq n + m$, and $\bar{t}_0 + 1 \leq i_1 < \ldots < i_{\bar{n}} = t$. The lemma also provides a set of coefficients $\alpha_{(i,j)}$, but we are going to let $g_j(i)$ be a suitably defined shifted version of the $\alpha_{(i,j)}$'s, such that they satisfy

$$g_j(i) = 0, \qquad i = i_j + 1, \ldots, t, \quad j = 1, \ldots, \bar{n} - 1, \tag{5.16}$$

$$|g_j(i)| \leq \left(1 + \frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{\bar{n}}, \quad i = 1, \ldots, i_j, \qquad j = 1, \ldots, \bar{n}, \tag{5.17}$$

and such that the approximation error

$$\bar{c}(i) := \tilde{\theta}(i) - \sum_{j=1}^{\bar{n}} g_j(i)\tilde{\theta}(i_j), \quad i = \bar{t}_0 + 1, \ldots, t \tag{5.18}$$

satisfies $\|\bar{c}(i)\| \leq \delta$. Thus, each $\tilde{\theta}(i)$ for $i = \bar{t}_0 + 1, \ldots, t$ is approximated by a linear combination of these basis vectors.

Now we apply (5.18) to (5.4) to express the prediction error in terms of these basis vectors:

$$\begin{aligned}
e_{\hat{\theta}(i+1)}(i+1) &= \Delta w(i) - \varphi(i)^\top R\tilde{\theta}(i+1) \\
&= \Delta w(i) - \varphi(i)^\top R \sum_{j=1}^{\bar{n}} g_j(i+1)\tilde{\theta}(i_j) - \varphi(i)^\top R\bar{c}(i+1) \\
&= \Delta w(i) + \sum_{j=1}^{\bar{n}} g_j(i+1) \left(\Delta w(i) - \varphi(i)^\top R\tilde{\theta}(i_j) - \Delta w(i)\right) - \varphi(i)^\top R\bar{c}(i+1) \\
&= \Delta w(i) + \sum_{j=1}^{\bar{n}} g_j(i+1) e_{\hat{\theta}(i_j)}(i+1) - \sum_{j=1}^{\bar{n}} g_j(i+1)\Delta w(i) - \varphi(i)^\top R\bar{c}(i+1) \\
&= \sum_{j=1}^{\bar{n}} g_j(i+1) e_{\hat{\theta}(i_j)}(i+1) + \left(1 - \sum_{j=1}^{\bar{n}} g_j(i+1)\right) \Delta w(i) - \varphi(i)^\top R\bar{c}(i+1), \\
&\qquad i = \bar{t}_0, ..., t-1.
\end{aligned} \tag{5.19}$$

This converts the problem of finding a bound for (5.11) to that of finding a bound for

$$\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |g_j(i+1) e_{\hat{\theta}(i_j)}(i+1)|, \quad j = 1, ..., \bar{n}.$$

To do so, we make use of Lemma 3.3, Claim 5.1, (5.16) and (5.17). We see that for each $j \in \{1, \ldots, \bar{n}\}$, there exists a projection operator $\psi_j : \{t \in \mathbf{Z} : t \geq \bar{t}_0\} \to \{0, 1\}$ satisfying

(5.15) such that

$$\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}(1-\psi_j(i))|g_j(i+1)e_{\hat{\theta}(i_j)}(i+1)|$$
$$\leq \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \sum_{i=\bar{t}_0}^{i_j-1} \lambda^{t-i-1}(1-\psi_j(i))|e_{\hat{\theta}(i_j)}(i+1)| \quad \text{(using (5.16) and (5.17))}$$
$$\leq \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \sqrt{\frac{1}{1-\lambda}\sum_{i=\bar{t}_0}^{i_j-1} \lambda^{i_j-i-1}(1-\psi_j(i))|e_{\hat{\theta}(i_j)}(i+1)|^2} \quad \text{(using Lemma 3.3)}$$
$$\leq \frac{1}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}\left(\sqrt{\sum_{i=\bar{t}_0}^{t-1}\lambda^{t-i-1}|\Delta w(i)|^2}+\bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|\right)$$
$$\text{(using Claim 5.1)}$$
$$\leq \frac{1}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}\left(\sum_{i=\bar{t}_0}^{t-1}\sqrt{\lambda}^{t-i-1}|\Delta w(i)|+\bar{k}\sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\|\right)$$
$$\text{(using Lemma 3.3)}\,.$$

Now we create a new projection operator $\Psi : \{\bar{t}_0, \ldots, t-1\} \to \{0, 1\}$ whose support is precisely the union of the supports of $\psi_j, j \in \{1, \ldots, \bar{n}\}$:

$$\Psi(i) := 1 - \prod_{j=1}^{\bar{n}} (1-\psi_j(i));$$

since $\bar{n} \leq n+m$, it satisfies

$$\sum_{i=\bar{t}_0}^{t-1} |\Psi(i)| \leq (n+m)^2, \tag{5.20}$$

and the signal

$$\hat{e}(i+1) := (1-\Psi(i))\sum_{j=1}^{\bar{n}} g_j(i+1)e_{\hat{\theta}(i_j)}(i+1), \quad i = \bar{t}_0, ..., t-1,$$

satisfies

$$\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} |\hat{e}(i+1)|$$
$$\leq \frac{n+m}{\sqrt{1-\lambda}} \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} \left(\sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |\Delta w(i)| + \bar{k}\sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\|\right). \tag{5.21}$$

By defining these new signals

$$\tilde{c}(i) := \Psi(i-1)\sum_{j=1}^{\bar{n}} g_j(i)\tilde{\theta}(i_j), \quad \bar{g}(i) := \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} \left(1+(\Psi(i-1)-1)\sum_{j=1}^{\bar{n}} g_j(i)\right),$$
$$i = \bar{t}_0+1, \ldots, t,$$

we can modify (5.19) to express it as

$$\begin{aligned}
e_{\hat{\theta}(i+1)}(i+1) &= \hat{e}(i+1) + \Psi(i)\sum_{j=1}^{\bar{n}} g_j(i+1)e_{\hat{\theta}(i_j)}(i+1) + \left(1-\sum_{j=1}^{\bar{n}} g_j(i+1)\right)\Delta w(i) \\
&\quad - \varphi(i)^\top R\bar{c}(i+1) \\
&= \hat{e}(i+1) + \Psi(i)\sum_{j=1}^{\bar{n}} g_j(i+1)\left(\Delta w(i) - \varphi(i)^\top R\tilde{\theta}(i_j)\right) \\
&\quad + \left(1-\sum_{j=1}^{\bar{n}} g_j(i+1)\right)\Delta w(i) - \varphi(i)^\top R\bar{c}(i+1) \quad \text{(using (5.4))} \\
&= \hat{e}(i+1) - (\tilde{c}(i+1)+\bar{c}(i+1))^\top R^\top\varphi(i) \\
&\quad + \left(1+(\Psi(i)-1)\sum_{j=1}^{\bar{n}} g_j(i+1)\right)\Delta w(i), \quad i=\bar{t}_0, ..., t-1.
\end{aligned}$$

By substituting this into (5.9) we obtain:

$$\begin{aligned}
\varphi(i+1) &= \left(A_{\hat{\theta}(i+1)\hat{\theta}(i+1)} - \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} (\tilde{c}(i+1)+\bar{c}(i+1))^\top R^\top\right)\varphi(i) \\
&\quad + \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} \hat{e}(i+1) + \bar{g}(i+1)\Delta w(i), \quad i=\bar{t}_0, ..., t-1.
\end{aligned} \tag{5.22}$$

Recall that $A_{\hat{\theta}(i+1)\hat{\theta}(i+1)}$ is a stable matrix, so this is a stable system subject to a perturbation of the sort considered in Lemma 3.2. If the perturbation is sufficiently small then the perturbed system will also be stable. Using (5.17), (5.20), $\bar{n} \leq n+m$ and $\|\tilde{\theta}(\cdot)\| \leq 2\|\mathcal{S}\|$, it is clear that

$$\sum_{i=\bar{t}_0}^{t-1} \|\tilde{c}(i+1)\| \leq 2\|\hat{\mathcal{S}}\|(n+m)^3 \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m},$$

which is independent of t. Also, we know that $\|\bar{c}(i)\| \leq \delta$, $i=\bar{t}_0+1,\ldots,t$. Thus,

$$\sum_{i=i_1}^{i_2-1} \left\| \begin{bmatrix} \bar{b}_n \\ 0 \end{bmatrix} (\tilde{c}(i+1)+\bar{c}(i+1))^\top R^\top \right\|$$
$$\leq 2\|\hat{\mathcal{S}}\|\|R\|(n+m)^3 \left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m} + \delta\|R\|(i_2-i_1), \quad \bar{t}_0 \leq i_1 < i_2 \leq t.$$

Hence, this 'perturbation' is small on average. Since $A_{\hat{\theta}(t)\hat{\theta}(t)}$ has stability margin[3] $\tilde{\lambda} < \lambda$, we can apply Lemma 3.2 and it follows that if we fix $\delta \in (0,\sigma]$ such that $\delta < \frac{\lambda-\tilde{\lambda}}{\|R\|\tilde{\gamma}}$, then (5.22) is a stable system with margin λ. Now define

$$\gamma_3 := 1+(n+m)\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m},$$

and observe that $\bar{g}(i) \leq \gamma_3$. From Lemma 3.2, we conclude that there exists a $\gamma_4 \geq 1$ so that for every $t \geq \bar{t}_0$ for which $\|\tilde{\theta}(t)\| \geq \delta$, we have

$$\|\varphi(t)\| \leq \gamma_4 \lambda^{t-\bar{t}_0}\|\varphi(\bar{t}_0)\| + \gamma_3\gamma_4 \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|\Delta w(i)| + \gamma_4 \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|\hat{e}(i+1)|.$$

Now define the constants

$$\gamma_5 := \gamma_3 + \frac{n+m}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m},$$
$$\gamma_6 := \bar{k}\gamma_4 \frac{n+m}{\sqrt{1-\lambda}}\left(1+\frac{2\|\hat{\mathcal{S}}\|}{\delta}\right)^{n+m}.$$

[3] Stability margin refers to the eigenvalues of the matrix lying within the open disk of some specified radius.

Finally, using (5.21) and knowing that $\lambda < \sqrt{\lambda}$, for every $t \geq \bar{t}_0$ such that $\|\tilde{\theta}(t)\| \geq \delta$ we have

$$\|\varphi(t)\| \leq \gamma_4 \sqrt{\lambda}^{t-\bar{t}_0} \|\varphi(\bar{t}_0)\| + \gamma_6 \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \gamma_4\gamma_5 \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |\Delta w(i)|. \tag{5.23}$$

Part 3: A bound on $\|\varphi(t)\|$ on the whole interval

We claim that there exists positive constants c_1, c_2, c_3 such that for any $\bar{t}_0 \geq 0$,

$$\|\varphi(t)\| \leq c_1 \sqrt{\lambda}^{t-\bar{t}_0} \|\varphi(\bar{t}_0)\| + c_2 \sqrt{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + c_3 \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1} |\Delta w(i)|, \quad t \geq \bar{t}_0. \tag{5.24}$$

Case 1: If $\|\tilde{\theta}(i)\| < \delta$ for all $i \in \{\bar{t}_0 + 1, \ldots, t\}$, then using (5.14), clearly (5.24) is satisfied with $c_1 = \bar{\gamma}, c_2 = 0, c_3 = \bar{\gamma}$.

Case 2: If $\|\tilde{\theta}(t)\| \geq \delta$, using (5.23), (5.24) is satisfied with $c_1 = \gamma_4, c_2 = \gamma_6, c_3 = \gamma_4\gamma_5$.

Case 3: If $\|\tilde{\theta}(t)\| < \delta$ and there exists any $\bar{t} \in \{\bar{t}_0 + 1, \ldots, t-1\}$ such that $\|\tilde{\theta}(\bar{t})\| \geq \delta$, then using (5.14) and (5.23) together, one can see that (5.24) is satisfied with $c_1 = \bar{\gamma}\gamma_4, c_2 = \bar{\gamma}\gamma_6, c_3 = \bar{\gamma}\gamma_4\gamma_5 + \bar{\gamma}$.

Combining each case and using the knowledge that $\bar{\gamma} \geq 1$ and $\gamma_4 \geq 1$, (5.24) holds in all cases for

$$c_1 = \bar{\gamma}\gamma_4, \quad c_2 = \bar{\gamma}\gamma_6, \quad c_3 = \bar{\gamma}\gamma_4\gamma_5 + \bar{\gamma}.$$

Part 4: A bound on $\|W(t)\|$

From (3.5), we see that

$$W(t) = \lambda^{t-\bar{t}_0} W(\bar{t}_0) + \sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \begin{bmatrix} \Delta\phi(i) \\ \Delta y(i+1) \end{bmatrix} \begin{bmatrix} \Delta\phi(i) \\ \Delta y(i+1) \end{bmatrix}^\top, \quad t \geq \bar{t}_0.$$

By defining the constant

$$\bar{j} := 2 + \sup_{\theta \in \mathcal{S}} \|\vartheta\|,$$

we can find

$$
\begin{aligned}
\|W(t)^{\frac{1}{2}}\| \le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left\| \begin{bmatrix} \Delta\phi(i) \\ \Delta y(i+1) \end{bmatrix} \right\|^2} \\
={}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left\| \begin{bmatrix} \Delta\epsilon(i+1) \\ \vdots \\ \Delta\epsilon(i-n+1) \\ \Delta u(i) \\ \vdots \\ \Delta u(i-m+1) \end{bmatrix} \right\|^2} \\
\le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(\left\| \begin{bmatrix} \epsilon(i+1) \\ \vdots \\ \epsilon(i-n+1) \\ \Delta u(i) \\ \vdots \\ \Delta u(i-m+1) \end{bmatrix} \right\| + \left\| \begin{bmatrix} \epsilon(i) \\ \vdots \\ \epsilon(i-n) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\| \right)^2} \\
\le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \left(\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(2\|\varphi(i)\| + |\epsilon(i+1)|\right)^2 \right)^{\frac{1}{2}} \\
\le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \left(\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1} \left(2\|\varphi(i)\| + \|\varphi(i)\|\|\vartheta^*\| + |\Delta w(i)|\right)^2 \right)^{\frac{1}{2}} \\
& \quad \text{(using (5.2))} \\
\le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \bar{j}\sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}\|\varphi(i)\|^2} + \sqrt{\sum_{i=\bar{t}_0}^{t-1} \lambda^{t-i-1}|\Delta w(i)|^2} \\
\le{}& \sqrt{\lambda}^{t-\bar{t}_0}\|W(\bar{t}_0)^{\frac{1}{2}}\| + \bar{j}\sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1}\|\varphi(i)\| + \sum_{i=\bar{t}_0}^{t-1} \sqrt{\lambda}^{t-i-1}|\Delta w(i)|, \quad t \ge \bar{t}_0.
\end{aligned}
$$

Thus, $\|W(t)^{\frac{1}{2}}\|$ is bounded by a filtered version of $\phi(t)$ with a pole at $z = \sqrt{\lambda}$, which by (5.24) is itself a convolution with a pole at $z = \sqrt{\lambda}$. Together, this would yield a convolution bound with a double-pole at $z = \sqrt{\lambda}$. Instead, by using the fact that $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, one can find a $\gamma_7 \geq 1$ such that a first order convolution bound holds:

$$\|W(t)^{\frac{1}{2}}\| \leq \gamma_7 \bar{\lambda}^{t-\bar{t}_0} \|\varphi(\bar{t}_0)\| + \gamma_7 \bar{\lambda}^{t-\bar{t}_0} \|W(\bar{t}_0)^{\frac{1}{2}}\| + \gamma_7 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |\Delta w(i)|, \quad t \geq \bar{t}_0.$$

By combining this with (5.24), and using the equivalence of norms:

$$\|W(t)^{\frac{1}{2}}\| \leq \|W(t)^{\frac{1}{2}}\|_F = \left\| \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \right\| \leq \sqrt{n+m+1}\, \|W(t)^{\frac{1}{2}}\|,$$

one can find a γ_8 to create the convolution bound:

$$\|\bar{x}(t)\| \leq \gamma_8 \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0)\| + \gamma_8 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |\Delta w(i)|, \quad t \geq \bar{t}_0. \qquad \square$$

5.3.1 A Bound on the Tracking Error

An immediate consequence of the convolution bound of Theorem 5.1 is a bound on the tracking error.

Corollary 5.1 *For every $\lambda \in (0,1)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and with $A^*(z^{-1}) = 1$ and $\tau_D = n+m+1$, there exists a $c \geq 1$ so that for every $\theta^* \in \mathcal{S}$, $\varphi(t_0) \in \mathbf{R}^{n+m+1}$, $\tau(t_0) \in \{0, \ldots, n+m\}$, $\hat{\theta}(t_0) \in \hat{\mathcal{S}}$, $r \in \mathbf{R}$, and $w \in l_\infty$, when the supervisory controller given by (3.4) - (3.6) and (5.6) is applied to the plant (5.1), the following bound holds:*

$$\sqrt{\sum_{j=t_0}^{t} |\epsilon(j)|^2} \leq c\|\bar{x}(t_0)\| + c\sqrt{\sum_{j=t_0}^{t-1} |\Delta w(j)|^2}, \quad j \geq t_0.$$

Proof: By applying Parseval's Theorem to the second term of (5.12):

$$\begin{aligned}\sqrt{\sum_{j=t_0}^{t} |\epsilon(j)|^2} &\leq \gamma \|\bar{x}(t_0)\| \sqrt{\sum_{j=t_0}^{t} \left(\bar{\lambda}^{t-\bar{t}_0}\right)^2} + \gamma \sqrt{\sum_{j=t_0}^{t} \left(\sum_{i=t_0}^{j-1} \bar{\lambda}^{t-i-1} |\Delta w(i)|\right)^2} \\ &\leq \frac{\gamma}{\sqrt{1-\bar{\lambda}^2}} \|\bar{x}(t_0)\| + \frac{\gamma}{1-\bar{\lambda}} \sqrt{\sum_{j=t_0}^{t-1} |\Delta w(j)|^2}. \qquad \square\end{aligned}$$

5.4 Robustness

Consider now a more complicated scenario: the plant is dependent on the time-varying parameter vector $\theta^*(t)$, subjected to an additive disturbance $w(t)$, and there are some unmodeled dynamics which enter the system via $\bar{w}(t)$:

$$y(t+1) = \phi(t)^\top \theta^*(t) + w(t) + \bar{w}(t). \tag{5.25}$$

We adopt a common model of acceptable time-variations used in adaptive control: with $\zeta > 0$ and $\eta > 0$, we let $s(\mathcal{S}, \zeta, \eta)$ denote the subset of $l_\infty(\mathbf{R}^{n+m})$ whose elements θ^* satisfy $\theta^*(t) \in \mathcal{S}$ for every $t \geq t_0$ as well as

$$\sum_{i=t_1}^{t_2-1} \|\theta^*(i+1) - \theta^*(i)\| \leq \zeta + \eta(t_2 - t_1), \quad t_2 > t_1 \geq t_0.$$

We also adopt a common model of unmodeled dynamics:

$$\begin{aligned} m(t+1) &= \beta m(t) + \beta\|\phi(t)\|, \\ \bar{w}(t) &\leq \mu m(t) + \mu\|\phi(t)\|. \end{aligned} \tag{5.26}$$

As argued in [35], this model subsumes the classical additive uncertainty, multiplicative uncertainty, and uncertainty in a coprime factorization, which is common in the robust control literature, e.g. see [57], and is commonly used in the adaptive control literature, e.g. see [24]. The only limitation is that the perturbations correspond to strictly causal terms.

We will now show that if the time-variations are slow enough and the size of the unmodeled dynamics are small enough, then the closed-loop system retains exponential stability as well as the convolution bound. Before presenting this result, we must first define the new vector

$$\bar{\phi}(t) := \begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ y(t-n) \\ u(t) \\ \vdots \\ u(t-m+1) \\ u(t-m) \end{bmatrix} \in \mathbf{R}^{n+m+2},$$

which is the same as $\phi(t)$, except that it contains copies of y and u one additional step into the past. The motivation for this choice will be clear later.

Theorem 5.2 *For every $\zeta > 0$, $\beta \in (0,1)$, $\lambda \in (0,1)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$, and with $A^*(z^{-1}) = 1$ and $\tau_D = n+m+1$, there exists a $\eta > 0$, $\hat{\lambda} \in (max\{\beta, \bar{\lambda}\}, 1)$ and $\gamma' \geq 1$ so that for every $\bar{t}_0 \geq t_0$, $\theta^* \in s(\mathcal{S}, \zeta, \eta)$, $\bar{\phi}(\bar{t}_0) \in \mathbf{R}^{n+m+2}$, $W(\bar{t}_0) \in \mathbf{R}^{(n+m+1)\times(n+m+1)}$ positive semidefinite and symmetric, $\hat{\theta}(\bar{t}_0) \in \hat{\mathcal{S}}$, $\tau(\bar{t}_0) \in \{0, \ldots, n+m\}$, $r \in \mathbf{R}$, and $w \in l_\infty$, when the supervisory controller given by (3.4) - (3.6) and (5.6) is applied to the plant (5.25) with $\bar{w}(t)$ satisfying (5.26), the following bound holds:*

$$\left\| \begin{bmatrix} \bar{\phi}(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \\ m(t) \end{bmatrix} \right\| \leq \gamma' \hat{\lambda}^{t-\bar{t}_0} \left\| \begin{bmatrix} \bar{\phi}(\bar{t}_0) \\ \mathrm{vec}\left(W(\bar{t}_0)^{\frac{1}{2}}\right) \\ m(\bar{t}_0) \end{bmatrix} \right\| + \gamma' \sum_{i=\bar{t}_0}^{t-1} \hat{\lambda}^{t-i-1} |w(i)| + \gamma' |r|,$$
$$t \geq \bar{t}_0.$$

Proof of Theorem 5.2

It is proven in [49] that this robustness property holds for a wide class of 'partially' exponentially stable systems which admit a convolution bound - that is, systems of the form (5.1) whose overall state can be decomposed as

$$x(t) = \begin{bmatrix} \phi(t) \\ z_1(t) \\ z_2(t) \end{bmatrix},$$

with $z_2(t)$ confined to some space $\mathcal{X}$, and with $\begin{bmatrix} \phi(t) \\ z_1(t) \end{bmatrix}$ obeying a convolution bound. In Theorem 5.1, we find a convolution bound in terms of

$$\bar{x}(t) = \begin{bmatrix} \varphi(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix},$$

but to be able to apply the results of [49], this must be converted into a convolution involving $\phi(t)$, not $\varphi(t)$. However, this actually is not sufficient, since the update equation for the matrix $W(t+1)$ is a function of both $\phi(t)$ and $\phi(t-1)$. This is why we introduced

$\bar{\phi}(t)$, as it encodes the same information as both $\phi(t)$ and $\phi(t-1)$. Now we are going to apply Theorems 1 and 3 of [49] using

$$\begin{aligned} z_1(t) &\Leftarrow \operatorname{vec}\left(W(t)^{\frac{1}{2}}\right), & z_2(t) &\Leftarrow \begin{bmatrix} \hat{\theta}(t) \\ \tau(t) \end{bmatrix}, \\ \phi(t) &\Leftarrow \bar{\phi}(t), & r(t) &\Leftarrow r, \\ \mathcal{X} &\Leftarrow \hat{\mathcal{S}} \times \{0, \dots, n+m\}. \end{aligned}$$

Observe that the plant model (5.1) may be rewritten as

$$y(t+1) = \bar{\phi}(t)^\top \begin{bmatrix} I_{n\times n} & \mathbf{0}_{n\times m} \\ \mathbf{0}_{1\times n} & \mathbf{0}_{1\times m} \\ \mathbf{0}_{m\times n} & I_{m\times m} \\ \mathbf{0}_{1\times n} & \mathbf{0}_{1\times m} \end{bmatrix} \theta^* + w(t), \quad t \geq t_0,$$

which satisfies equation (1) of [49]. The equation for $W(t+1)$ is a function of $W(t)$, $y(t+1)$, $y(t)$, $\phi(t)$ and $\phi(t-1)$, so it may be written in the form

$$W(t+1) = h_1\left(W(t), y(t+1), \bar{\phi}(t)\right).$$

Similarly, the equations for $\hat{\theta}(t+1)$ and $\tau(t+1)$ may be written as

$$\begin{bmatrix} \hat{\theta}(t+1) \\ \tau(t+1) \end{bmatrix} = h_2\left(W(t), \hat{\theta}(t), \tau(t), y(t+1), \bar{\phi}(t)\right).$$

Lastly, the control law is a function of $\hat{\theta}(t)$ and $\varphi(t)$, so it takes the form

$$u(t) = h(\hat{\theta}(t), \bar{\phi}(t), r).$$

Together, these satisfy equations (2a) - (2c) of [49].[4] All that remains before we can apply the results of that paper is to convert the convolution bound of Theorem 5.1 into one involving the term

$$\begin{bmatrix} \bar{\phi}(t) \\ \operatorname{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}.$$

[4]While the original version of [49] does not allow for $y(t+1)$ in the controller description, the updated version does so (with minimal changes to the proof).

The method for doing so is based on the technique shown in the appendix of [50]. Fix $\lambda \in (0,1)$, $\bar{\lambda} \in (\sqrt{\lambda}, 1)$. Let $\theta^* \in \mathcal{S}$ and $\bar{t}_0 \geq t_0$ be arbitrary. Then, from Theorem 5.1, there exists a $\gamma \geq 1$ such that

$$\|\bar{x}(t)\| \leq \gamma \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0)\| + \gamma \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |\Delta w(i)|, \quad t \geq \bar{t}_0. \tag{5.27}$$

We are going to first find a bound for $\bar{\phi}(t)$ in terms of $\|\bar{x}(\bar{t}_0)\|$ and the exogenous inputs. Then we convert this into the desired convolution bound involving $\begin{bmatrix} \bar{\phi}(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}$.

Step 1: A bound on $\bar{\phi}(t)$ for $t \geq \bar{t}_0 + 2\max\{n, m\} + 2$

For each $\bar{n} \in \{0, \ldots, n\}$ and $\bar{m} \in \{1, \ldots, m\}$, let $\mathcal{S}_{(\bar{n},\bar{m})}$ denote the set of all $\theta^* \in \mathcal{S}$ for which the polynomial $A_{\theta^*}(z^{-1})$ is of degree $\bar{n}$ and $B_{\theta^*}(z^{-1})$ is of degree $\bar{m}$; from Assumption 5.3 this set is compact.

Now fix $\bar{n} \in \{0, \ldots, n\}$ and $\bar{m} \in \{1, \ldots, m\}$ and let $\theta^* \in \mathcal{S}_{(\bar{n},\bar{m})}$ be arbitrary. Then the plant's transfer function can be written as

$$\frac{b_1^* z^{-1} + \ldots + b_{\bar{m}}^* z^{-\bar{m}}}{1 - a_1^* z^{-1} - \ldots - a_{\bar{n}}^* z^{-\bar{n}}},$$

which is of order $N := \max\{\bar{n}, \bar{m}\}$. Now we construct a state-space model of the plant; we choose one of dimension N which is in observable canonical form:

$$z(t+1) = \underbrace{\begin{bmatrix} a_1^* & 1 & & & \\ a_2^* & 0 & 1 & & \\ \vdots & & \ddots & \ddots & \\ a_{\bar{n}}^* & & & 0 & 1 \\ & & \mathbf{0}_{(N-\bar{n})\times N} & & \end{bmatrix}}_{=:A} z(t) + \underbrace{\begin{bmatrix} b_1^* \\ \vdots \\ b_{\bar{m}}^* \\ \mathbf{0}_{N-\bar{m}} \end{bmatrix}}_{=:B} u(t) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} w(t)$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}}_{=:C} z(t). \tag{5.28}$$

Observe that we can view u as the output of a first order system:

$$u(t+1) = u(t) + \Delta u(t+1). \tag{5.29}$$

Since $\epsilon(t) = y(t) - r$, by combining (5.28) with (5.29) we obtain the augmented $(N+1)^{\text{th}}$ order state-space system

$$\begin{bmatrix} z(t+1) \\ u(t+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A & B \\ \mathbf{0}_{1\times N} & 1 \end{bmatrix}}_{=:\bar{A}} \underbrace{\begin{bmatrix} z(t) \\ u(t) \end{bmatrix}}_{=:\bar{z}(t)} + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{=:\bar{B}_1} \Delta u(t) + \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{=:\bar{B}_2} w(t)$$

$$\epsilon(t) = \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}}_{=:\bar{C}} \begin{bmatrix} z(t) \\ u(t) \end{bmatrix} - r, \quad t \geq \bar{t}_0. \tag{5.30}$$

Since (5.28) is controllable and observable (Assumption 5.1) and does not have a zero at $z = 1$ (Assumption 5.2), it follows that $\left(\bar{C}, \bar{A}\right)$ is observable; hence there exists a unique $\bar{H}$ such that the eigenvalues of $\bar{A} + \bar{H}\bar{C}$ are all zero. It is well known that $\bar{H}$ is an analytic function of $\theta^* \in \mathcal{S}_{(\bar{n},\bar{m})}$. Now rewrite (5.30) as

$$\bar{z}(t+1) = \left(\bar{A} + \bar{H}\bar{C}\right) \bar{z}(t) + \bar{B}_1 \Delta u(t) + \bar{B}_2 w(t) - \bar{H}\epsilon(t) - \bar{H}r, \quad t \geq \bar{t}_0.$$

Noting that $\left(\bar{A} + \bar{H}\bar{C}\right)^k = 0$ for all $k \geq N+1$, the solution of the above equation is

$$\bar{z}(t) = \sum_{i=t-N-1}^{t-1} \left(\bar{A} + \bar{H}\bar{C}\right)^{t-i-1} \left(\bar{B}_1 \Delta u(i) + \bar{B}_2 w(i) - \bar{H}\epsilon(i) - \bar{H}r\right), \quad t \geq \bar{t}_0 + N + 1.$$

Since $\epsilon(i)$ and $\Delta u(i)$ are elements of $\varphi(i)$, it follows that

$$\|\bar{z}(t)\| = \sum_{i=t-N-1}^{t-1} \left\| \left(\bar{A} + \bar{H}\bar{C}\right)^{t-i-1} \right\| \left((1 + \|\bar{H}\|)\|\varphi(i)\| + |w(i)| + \|\bar{H}\||r|\right), \quad t \geq \bar{t}_0 + N + 1.$$

Now by applying the convolution bound (5.27) along with $|\Delta w(i)| \leq |w(i)| + |w(i-1)|$, it follows that there exists a $\gamma_{(\bar{n},\bar{m})} \geq \gamma$ such that the bound

$$\|\bar{z}(t)\| \leq \gamma_{(\bar{n},\bar{m})} \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0 + 1)\| + \gamma_{(\bar{n},\bar{m})} \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |w(i)| + \gamma_{(\bar{n},\bar{m})} |r|, \quad t \geq \bar{t}_0 + N + 1$$

holds uniformly for all $\theta^* \in \mathcal{S}_{(\bar{n},\bar{m})}$. By choosing γ_1 to be the maximum of all $\gamma_{(\bar{n},\bar{m})}$, it follows that the bound

$$\|\bar{z}(t)\| \leq \gamma_1 \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0 + 1)\| + \gamma_1 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |w(i)| + \gamma_1 |r|, \quad t \geq \bar{t}_0 + \max\{n, m\} + 1$$

holds uniformly for all $\theta^* \in \mathcal{S}$. Finally, since $y(t) = \bar{C}\bar{z}(t)$ and $u(t) = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \bar{z}(t)$, we can relate $\bar{\phi}$ to $\bar{z}$:

$$\begin{aligned} \|\bar{\phi}(t)\| &= \left\| \begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ y(t-n) \\ u(t) \\ \vdots \\ u(t-m+1) \\ u(t-m) \end{bmatrix} \right\| \\ &\leq \sum_{i=t-\max\{n,m\}-1}^{t} \|\bar{z}(i)\|, \quad t \geq \bar{t}_0 + \max\{n,m\} + 1. \end{aligned}$$

It follows that there exists a $\gamma_2 \geq \gamma_1$ such that

$$\|\bar{\phi}(t)\| \leq \gamma_2 \bar{\lambda}^{t-\bar{t}_0} \|\bar{x}(\bar{t}_0+1)\| + \gamma_2 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |w(i)| + \gamma_2 |r|, \quad t \geq \bar{t}_0 + 2\max\{n,m\} + 2. \tag{5.31}$$

Step 2: Creating the desired convolution bound for $t \geq \bar{t}_0$

There are three remaining steps to convert (5.31) into the desired bound. Firstly, (5.31) involves $\|\bar{x}(\bar{t}_0+1)\|$ instead of the desired term. To fix this, it is easy to see that there exists a $\gamma_3 \geq 1$ such that

$$\|\bar{x}(\bar{t}_0+1)\| \leq \gamma_3 \begin{bmatrix} \bar{\phi}(\bar{t}_0+1) \\ \operatorname{vec}\left(W(\bar{t}_0+1)^{\frac{1}{2}}\right) \end{bmatrix} + \sqrt{n+1}|r|.$$

Thus, there exists a $\gamma_4 \geq \gamma_2$ such that

$$\begin{aligned} \|\bar{\phi}(t)\| \leq \gamma_4 \bar{\lambda}^{t-\bar{t}_0+1} \left\| \begin{bmatrix} \bar{\phi}(\bar{t}_0+1) \\ \operatorname{vec}\left(W(\bar{t}_0+1)^{\frac{1}{2}}\right) \end{bmatrix} \right\| + \gamma_4 \sum_{i=\bar{t}_0}^{t-1} \bar{\lambda}^{t-i-1} |w(i)| + \gamma_4 |r|, \\ t \geq \bar{t}_0 + 2\max\{n,m\} + 2. \end{aligned}$$

The next issue is the domain for which the above equation is valid, and the starting index of $\bar{t}_0 + 1$. To remedy this, we claim that for any $p \geq 0$, there exists a $\bar{c} \geq 1$ such that

$$\|\phi(t+p)\| \leq \bar{c}\|\phi(t)\| + \bar{c}\sum_{i=0}^{p-1}|w(t+p)| + \bar{c}|r|, \quad t \geq t_0. \tag{5.32}$$

Using the compactness of $\mathcal{S}$, we have from the plant definition that $|y(t+1)| \leq \|\mathcal{S}\|\|\phi(t)\| + |w(t)|$, and from the control law there exists a $\bar{c}_1$ such that $u(t+1) \leq \bar{c}_1\|\phi(t)\| + \bar{c}_1|r|$. Thus,

$$\|\phi(t+1)\| \leq (1 + \|\mathcal{S}\| + \bar{c}_1)\|\phi(t)\| + |w(t)| + \bar{c}_1|r|, \quad t \geq t_0.$$

Iterating this reveals (5.32). Similarly, we use

$$\begin{aligned}
\|W(\bar{t}_0+1)^{\frac{1}{2}}\| &\leq \gamma\|W(\bar{t}_0)^{\frac{1}{2}}\| + \left\|\begin{bmatrix} \Delta\phi(\bar{t}_0) \\ \Delta y(\bar{t}_0+1) \end{bmatrix}\right\| \\
&\leq \gamma\|W(\bar{t}_0)^{\frac{1}{2}}\| + |y(\bar{t}_0+1)| + 2\|\bar{\phi}(\bar{t}_0)\| \\
&\leq \gamma\|W(\bar{t}_0)^{\frac{1}{2}}\| + (2 + \|\mathcal{S}\|)\|\bar{\phi}(\bar{t}_0)\| + |w(\bar{t}_0)|.
\end{aligned}$$

It follows that there exists a $\gamma_5 \geq \gamma_4$ such that

$$\|\bar{\phi}(t)\| \leq \gamma_5\bar{\lambda}^{t-\bar{t}_0}\left\|\begin{bmatrix} \bar{\phi}(\bar{t}_0) \\ \mathrm{vec}\left(W(\bar{t}_0)^{\frac{1}{2}}\right) \end{bmatrix}\right\| + \gamma_5\sum_{i=\bar{t}_0}^{t-1}\bar{\lambda}^{t-i-1}|w(i)| + \gamma_5|r|, \quad t \geq \bar{t}_0.$$

Finally, the LHS of the above inequality is wrong: to fix this, use

$$\left\|\begin{bmatrix} \bar{\phi}(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}\right\| \leq \|\bar{\phi}(t)\| + \|\bar{x}(t)\|,$$

along with (5.27). Thus, there exists a $\gamma_6 \geq \gamma_5$ such that

$$\left\|\begin{bmatrix} \bar{\phi}(t) \\ \mathrm{vec}\left(W(t)^{\frac{1}{2}}\right) \end{bmatrix}\right\| \leq \gamma_6\bar{\lambda}^{t-\bar{t}_0}\left\|\begin{bmatrix} \bar{\phi}(\bar{t}_0) \\ \mathrm{vec}\left(W(\bar{t}_0)^{\frac{1}{2}}\right) \end{bmatrix}\right\| + \gamma_6\sum_{i=\bar{t}_0}^{t-1}\bar{\lambda}^{t-i-1}|w(i)| + \gamma_6|r|,$$
$$t \geq \bar{t}_0.$$

This is the desired convolution bound, so Theorem 5.2 follows by applying Theorems 1 and 3 of [49]. $\square$

5.5 Simulation Results

Now we demonstrate in simulation the results proven in Theorems 5.1 - 5.2. The performance is compared to the adaptive controller constructed using the Projection Algorithm using the setup of [48], as it is the only other adaptive controller which has been shown to exhibit the same linear-like closed-loop behavior. Consider the second order plant

$$y(t+1) = \underbrace{\begin{bmatrix} a_1^* \\ a_2^* \\ b_1^* \\ b_2^* \end{bmatrix}}_{\theta^*}^{\top} \underbrace{\begin{bmatrix} y(t) \\ y(t-1) \\ u(t) \\ u(t-1) \end{bmatrix}}_{\phi(t)} + w(t),$$

with parameters belonging to the set

$$\mathcal{S} = \left\{ \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} : a_1 \in [-2, 0], a_2 \in [-3, -1].b_1 \in [-1, 0], b_2 \in [-5, -3] \right\},$$

which satisfies Assumptions 5.1 - 5.3. Observe that each model in this set is unstable and non-minimum phase, hence the d-step-ahead method cannot be used. Since $\mathcal{S}$ is convex, we choose $\hat{\mathcal{S}} = \mathcal{S}$. Both estimators use data filter

$$H(z) = \frac{z-1}{z},$$

initial parameter guess

$$\hat{\theta}(0) = \begin{bmatrix} -1 \\ -2 \\ -0.5 \\ -4 \end{bmatrix},$$

and both controllers use a pole placement certainty equivalence control law with integrator to place the closed-loop poles at the origin. The Supervisory Controller's minimum dwell time is $\tau_D = n + m + 1 = 5$. The Supervisory estimator is also initialized with

$$W(0) = 0, \quad \tau(0) = \tau_D - 1 = 4.$$

The plant's initial condition is

$$y(0) = 0, \quad y(-1) = 0, \quad u(0) = 0, \quad u(-1) = 0.$$

The reference signal is the constant setpoint $r = 2$. The plant parameters are

$$\theta^* = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1.5 \\ -0.75 \\ -3 \end{bmatrix},$$

and the Supervisory Controller uses exponential forgetting factor $\lambda = 0.9$. The disturbance w is set to have constant magnitude: $|w(t)| = 0.5$, but with its sign changing every 100 steps. We plot the results in Figure 5.2; the parameter estimates for Supervisory Control lock on at just $t = 6$ and never change again, so thenceforth its response is just that of an LTI pole placement controller. In contrast, the Projection Algorithm's estimate continues to jump around.

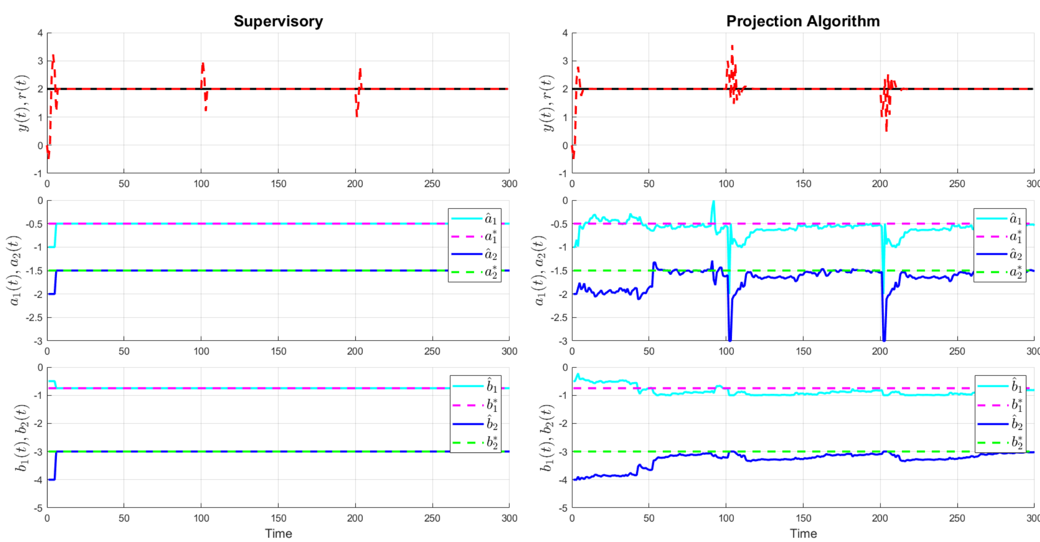

Figure 5.2: Performance of Supervisory Control vs the Projection Algorithm in the presence of a piecewise constant disturbance.

The setup of the next simulation is the same as above, except now the disturbance signal w is a Gaussian random signal with standard deviation 0.1, whose mean switches between 0.5 and -0.5 every 100 steps. We plot the results in Figure 5.3; in this case the tracking performance of the two estimators is very similar.

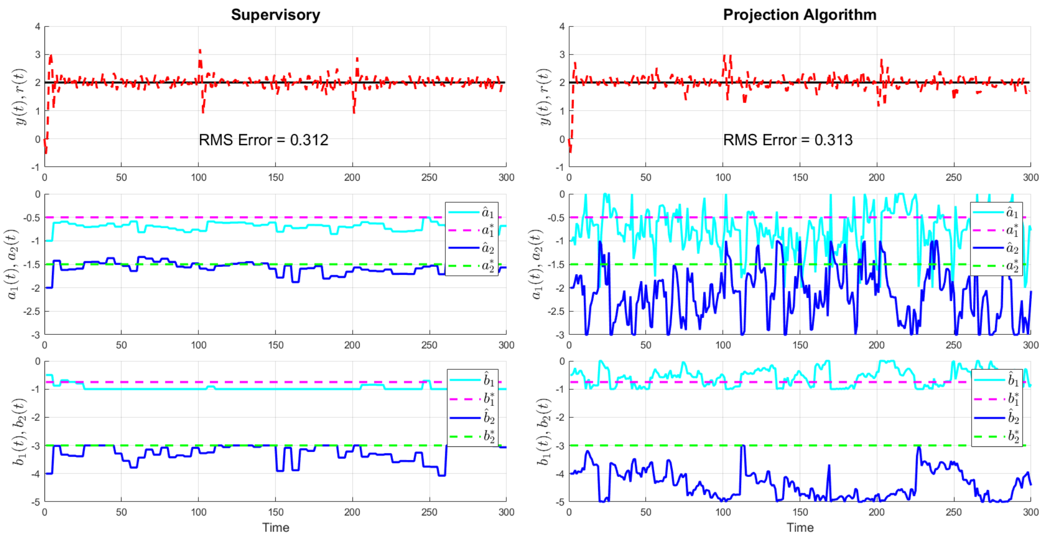

Figure 5.3: Performance of Supervisory Control vs the Projection Algorithm in the presence of a random disturbance with piecewise constant mean.

The next simulation illustrates the robustness of the two estimators to time-varying parameters. The setup is the same as above, except that the plant parameters are time varying:

$$
\begin{aligned}
a_1^*(t) &= -1 + \sin(0.005\pi t), & a_2^*(t) &= -2 - \sin(0.004\pi t), \\
b_1^*(t) &= -0.5 + 0.5\sin(0.01\pi t), & b_2^*(t) &= -4 + \sin(0.007\pi t).
\end{aligned}
$$

The Supervisory Controller now uses exponential forgetting factor $\lambda = 0.75$, which was selected experimentally. Generally, a higher λ improves its rejection of random disturbances, but a lower λ allows for better tracking of time-varying parameters. We plot the results in Figure 5.4. Qualitatively, the parameter estimates for the Supervisory Controller are less chaotic than that of the Projection Algorithm, and the tracking performance is improved.

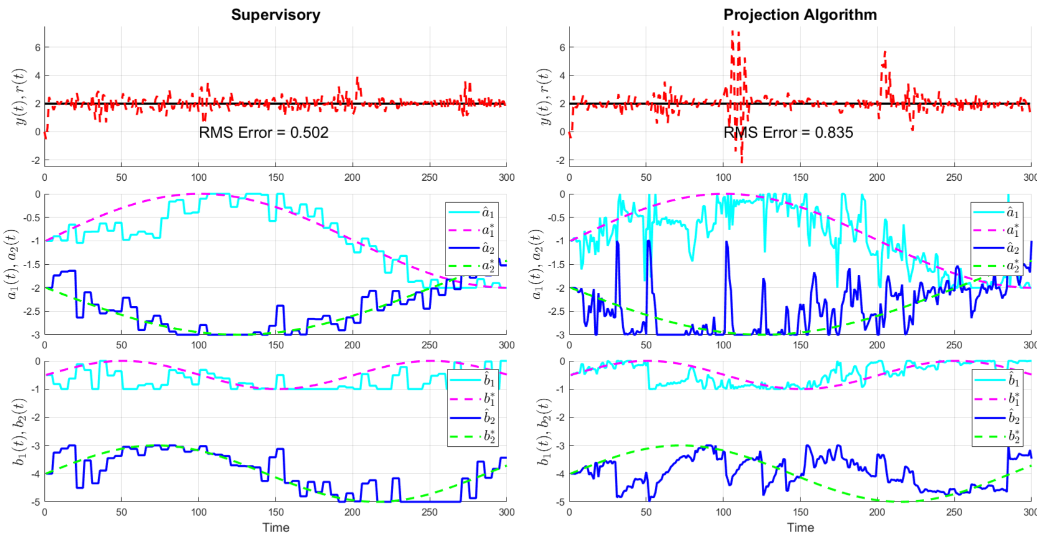

Figure 5.4: Performance of Supervisory Control vs the Projection Algorithm in the presence of a random disturbance with piecewise constant mean and time-varying parameters.

The final simulation, Figure 5.5, is precisely the same as above, except that we let the Supervisory estimator's dwell time be one, so $\hat{\theta}$ is permitted to change at every step. It has not been proven that the system is stable in this case, however it works very well in simulation; the tracking performance is superior than of Figure 5.4.

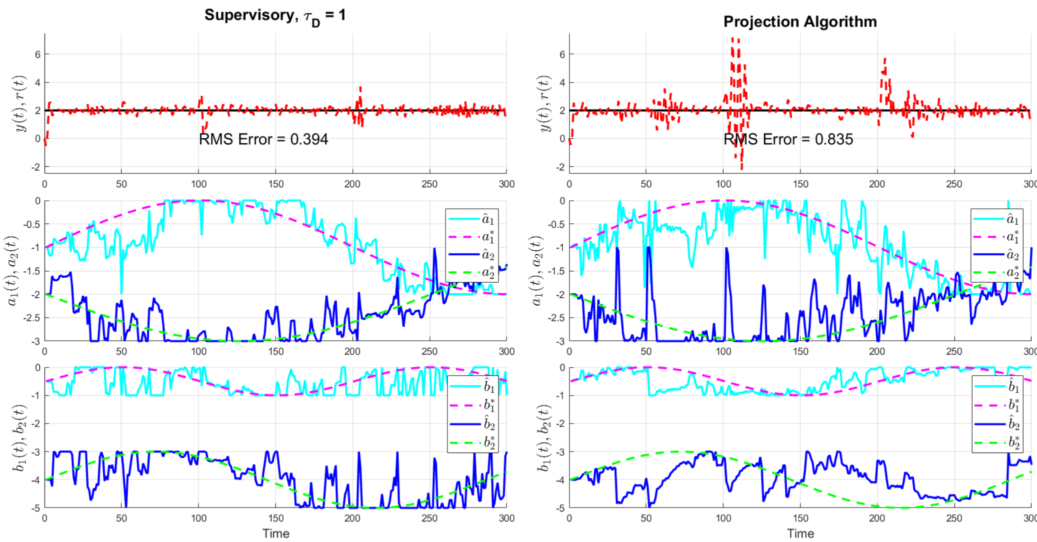

Figure 5.5: Performance of Supervisory Control without a dwell time vs the Projection Algorithm in the presence of a random disturbance with piecewise constant mean and time-varying parameters.

Chapter 6

Conclusions and Future Work

The main contribution of this thesis is in demonstrating that Supervisory Control yields a closed-loop adaptive system with 'linear-like' properties - namely that part of the system state is exponentially stable, and the influence of the exogenous inputs is bounded by a linear convolution. It is argued in Chapter 2 that this property is not shared with the majority of competing parameter estimation routines, thus the adaptive system constructed with the Supervisory estimator has superior robustness qualities than the other methods. This thesis applies the Supervisory Control method in two contexts: the d-step-ahead problem of tracking an arbitrary reference signal in Chapter 4, and the step tracking problem of tracking a constant reference in the presence of a constant disturbance in Chapter 5. In each case it is shown that the closed-loop system admits a convolution bound which holds uniformly for all admissible initial conditions. This powerful result is then leveraged in a modular fashion to show that the system is robust to a degree of time-varying parameters and unmodeled dynamics.

Recall that this is not the first time that an adaptive system has been shown to have this linear-like property; it is shown in e.g. [35, 36] that the 'ideal' Projection Algorithm estimator also yields such a convolution bound. However, Supervisory Control has a number of benefits over the Projection Algorithm. Firstly, Supervisory Control tolerates a non-convex parameter space $\mathcal{S}$ very naturally, whereas the Projection Algorithm as in [35, 36] requires that $\mathcal{S}$ be convex, although the approach may be applied to a non-convex set at the expense of greater complexity: this is done in e.g. [46, 47, 48, 50] by breaking up $\mathcal{S}$ into several convex subregions and then running in parallel a separate parameter estimator for each subregion. Secondly, in the d-step-ahead Supervisory Control approach, the order and relative degree of the plant need not be known exactly, only an upper bound need be known, and the sign of the high frequency gain need not be known, which are advantages over [35]. Another advantage is that in many situations, Supervisory Control has superior performance in simulation. Lastly, Supervisory Control permits plants which are nonlinearly dependent on their parameter vector: that is, systems of the form $y(t+1) = \phi(t)^\top f(\theta^*) + w(t)$. The advantages of this are discussed in [40], with an example of such a system in Section X therein. For the sake of simplicity, this thesis has been entirely restricted to linearly parameterized systems, but with a simple extension, all the results seen here should apply to the more general case too.

With both setups, in the case of no disturbance, constant parameters, and no unmodeled dynamics, we find a crisp bound on the tracking error. However, when the disturbance is nonzero, nothing has been proven in the d-step-ahead case, whereas with step tracking it has been shown that the energy of the tracking error is proportional to the initial condition and the energy of the disturbance. Therefore, an avenue for future work would be to find some sort of useful bound on the tracking error in the d-step-ahead context in the presence of a disturbance.

Finally, in the pole placement context, by deliberately choosing to place the poles at the origin, the dwell time may be minimized to 'only' $n + m + 1$. However, in the d-step-ahead context, the dwell-time may need to be much longer, depending on how slow are the zeros in the set of admissible plant models. All general stability proofs to date rely on the dwell time constraint, however it has never been proven that it is necessary. The simulation results suggest that the system is well behaved without a dwell time, in fact the performance is greatly improved by removing it. Recall that without the dwell time, the estimator simplifies to being the weighted least-squares estimator of Section 2.4, except the optimization is performed over the set $\hat{\mathcal{S}}$. The author suspects that the system will be stable without the dwell time and has attempted to prove so, but without success; it would be worthwhile to further investigate the nature of Supervisory Control without the dwell time constraint.

References

[1] B.D.O Anderson, T.S. Brinsmead, F. De Bruyne, J.P. Hespanha, D. Liberzon, and A.S. Morse. Multiple model adaptive control, part 1: finite controller coverings. In *International Jounral of Robust and Nonlinear Control*, volume 10(11-12), pages 909–929, 2000.

[2] P. J. Antsaklis and A. N. Michel. *Linear Systems*. McGraw-Hill, New York, 1997.

[3] D. Borrelli and A.S. Morse. Discrete-time supervisory control of families of linear set-point controllers. In *IEEE Transactions on Automatic Control*, volume 42(11), pages 1500–1515, 1997.

[4] D. Borrelli, A.S. Morse, and E. Mosca. Discrete-time supervisory control of families of 2-dof linear set-point controllers. In *IEEE Transactions on Automatic Control*, volume 44(1), pages 178–181, 1999.

[5] C. A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, New York, 1975.

[6] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky. Adaptive control and the nasa x-15-3 flight revisited. In *IEEE Control Systems Magazine*, volume 30(3), pages 32–48, 2010.

[7] C.E. Rohrs et al. Robustness of continuous-time adaptive control algorithms in the presence of unmodelled dynamics. In *IEEE Transactions on Automatic Control*, volume 30(9), pages 881–889, 1985.

[8] R.H. Middleton et al. Design issues in adaptive control. In *IEEE Transactions on Automatic Control*, volume 33(1), pages 50–58, 1988.

[9] A. Feuer and A.S. Morse. Adaptive control of single-input, single-output linear systems. In *IEEE Transactions on Automatic Control*, volume 23(4), pages 557–569, 1978.

[10] M. Fu and B.R. Barmish. Adaptive stabilization of linear systems via switching control. In *IEEE Transactions on Automatic Control*, volume 31(12), pages 1097–1103, 1986.

[11] G.C. Goodwin, P.J. Ramadge, and P.E. Caines. Discrete time multivariable control. In *IEEE Transactions on Automatic Control*, volume 25(3), pages 449–456, 1980.

[12] G.C. Goodwin and K.S. Sin. *Adaptive Filtering Prediction and Control*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1984.

[13] W. M. Haddad and V. Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, 2008.

[14] J. P. Hespanha, D. Liberzon, and A. S. Morse. Hysteresis-based switching algorithms for supervisory control of uncertain system. In *Automatica*, volume 39(2), pages 263–272, 2003.

[15] J.P. Hespanha. Tutorial on supervisory control. Lecture Notes for the workshop *Control using Logic and Switching* for the 40th Conf. on Decision and Contr., Orlando, Florida, 2001. Available at http://www.ece.ucsb.edu/~hespanha/published.

[16] J.P. Hespanha. Supervisory control of nonlinear systems. In *Proc. of the 2007 European Contr. Conf.*, 2007.

[17] J.P. Hespanha, D. Liberzon, and A.S. Morse. Logic-based switching control of a nonholonomic system with parametric modeling uncertainty. In *Systems and Control Letters*, volume 38(3), pages 167–177, 1999.

[18] J.P. Hespanha, D. Liberzon, and A.S. Morse. Supervision of integral-input-to-state stabilizing controllers. In *Automatica*, volume 38(8), pages 1327–1335, 2002.

[19] J.P. Hespanha, D. Liberzon, A.S. Morse, B.D.O Anderson, T.S. Brinsmead, and F. De Bruyne. Multiple model adaptive control, part 2: switching. In *International Journal of Robust and Nonlinear Control*, volume 11(5), pages 479–496, 2001.

[20] J. Hockerman-Frommer, S.R. Kulkarni, and P.J. Ramadge. Controller switching based on output prediction errors. In *IEEE Transactions on Automatic Control*, volume 43(5), pages 596–607, 1998.

[21] P.A. Ioannou and K.S. Tsakalis. A robust direct adaptive controller. In *IEEE Transactions on Automatic Control*, volume 31(11), pages 1033–1043, 1986.

[22] G. Kreisselmeier. Adaptive control of a class of slowly time-varying plants. In *Systems and Control Letters*, volume 8, pages 97–103, 1986.

[23] G. Kreisselmeier and B.D.O. Anderson. Robust model reference adaptive control. In *IEEE Transactions on Automatic Control*, volume 31(2), pages 127–133, 1986.

[24] G. Kreisselmeier and B.D.O. Anderson. Robust model reference adaptive control. In *IEEE Transactions on Automatic Control*, volume 31(2), pages 127–133, 1986.

[25] C.J. Lalumiere. First order supervisory adaptive control. `https://github.com/CraigLalumiere/First-Order-Supervisory-Adaptive-Control`, 2022.

[26] Y. Li and H-F Chen. Robust adaptive pole placement for linear time-varying systems. In *IEEE Transactions on Automatic Control*, volume 41(5), pages 714–719, 1996.

[27] I.M.Y. Mareels and R.R. Bitmead. Nonlinear dynamics in adaptive control: Chaotic and periodic stabilization. In *IFAC Proceedings Volumes*, volume 20(2), pages 419–424, 1987.

[28] I.M.Y. Mareels and R.R. Bitmead. Adaptive dual control methods: An overview,. In *IFAC Proceedings Volumes*, volume 28(13), pages 67–72, 1995.

[29] R.H. Middleton and G.C. Goodwin. Adaptive control of time-varying linear systems. In *IEEE Transactions on Automatic Control*, volume 33(2), pages 150–155, 1988.

[30] D.E. Miller. On necessary assumptions in discrete-time model reference adaptive control. In *International Journal of Adaptive Control and Signal Processing*, volume 10(6), pages 589–602, 1996.

[31] D.E. Miller. Classical discrete-time adaptive control revisited: Exponential stabilization. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1975–1980, 2017.

[32] D.E. Miller. A parameter adaptive controller which provides exponential stability: The first order case. In *Systems and Control Letters*, volume 103, pages 23–31, 2017.

[33] D.E. Miller and E.J. Davison. An adaptive controller which provides lyapunov stability. In *IEEE Transactions on Automatic Control*, volume 34(6), pages 599–609, 1989.

[34] D.E. Miller and E.J. Davison. A new approach to model reference adaptive control. In *IEEE Transactions on Automatic Control*, volume 48(5), pages 743–757, 2003.

[35] D.E. Miller and M.T. Shahab. Classical pole placement adaptive control revisited: linear-like convolution bounds and exponential stability. In *Math Control Signals Syst*, volume 30(4):19, 2018.

[36] D.E. Miller and M.T. Shahab. Adaptive tracking with exponential stability and convolution bounds using vigilant estimation. In *Math. Control Signals Syst*, volume 32, pages 241–291, 2020.

[37] D.E. Miller and M.T. Shahab. The inherent robustness of a new approach to adaptive control. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pages 510–515, 2020.

[38] D.E. Miller and M.T. Shahab. Linear-like properties arise naturally in the adaptive control setting. In *International Journal of Adaptive Control and Signal Processing*, volume 35(6), pages 965–990, 2021.

[39] A.S. Morse. Global stability of parameter-adaptive control systems. In *IEEE Transactions on Automatic Control*, volume 25(3), pages 433–439, 1980.

[40] A.S. Morse. Supervisory control of families of linear set-point controillers - part 1: Exact matching. In *IEEE Transactions on Automatic Control*, volume 41(10), pages 1413–1431, 1996.

[41] A.S. Morse. Supervisory control of families of linear set-point controillers - part 2: Robustness. In *IEEE Transactions on Automatic Control*, volume 42(11), pages 1500–1515, 1997.

[42] E. Mosca, F. Capecchi, and A. Casavola. Designingpredictors for mimo switchingsupervisory control. In *International Journal of Robust and Nonlinear Control*, volume 15(3), pages 265–286, 2001.

[43] S.M. Naik, P.R. Kumar, and B.E. Ydstie. Robust continuous-time adaptive control by parameter projection. In *IEEE Transactions on Automatic Control*, volume 37(2), pages 182–197, 1992.

[44] K.S. Narendra and Y.H. Lin. Stable discrete adaptive control. In *IEEE Transactions on Automatic Control*, volume 25(3), pages 456–461, 1980.

[45] K.S. Narendra, Y.H. Lin, and L.S. Valavani. Stable adaptive controller design, part ii: Proof of stability. In *IEEE Transactions on Automatic Control*, volume 25(3), pages 440–448, 1980.

[46] Hassaan Ali Qazi. Adaptive control of a first-order system providing linear-like behaviour and asymptotic tracking. Master's thesis, University of Waterloo, 2021.

[47] M.T. Shahab and D.E. Miller. Multi-estimator based adaptive control which provides exponential stability: The first-order case. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2223–2228, 2018.

[48] M.T. Shahab and D.E. Miller. Adaptive set-point regulation using multiple estimators. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 84–89, 2019.

[49] M.T. Shahab and D.E. Miller. A convolution bound implies tolerance to time-variations and unmodelled dynamics, 2019. arXiv:1910.02112.

[50] M.T. Shahab and D.E. Miller. Asymptotic tracking and linear-like behavior using multi-model adaptive control. In *IEEE Transactions on Automatic Control*, volume 67(1), pages 203–219, 2022.

[51] K.S. Tsakalis and P.A. Ioannou. Adaptive control of linear time-varying plants: A new model reference controller structure. In *IEEE Transactions on Automatic Control*, volume 34(10), pages 1038–1046, 1989.

[52] L. Vu and D. Liberzon. Supervisory control of uncertain linear time-varying systems. In *IEEE Transactions on Automatic Control*, volume 56(1), pages 27–42, 2011.

[53] C. Wen. A robust adaptive controller with minimal modifications for discrete time-varying systems. In *IEEE Transactions on Automatic Control*, volume 39(5), pages 987–991, 1994.

[54] C. Wen and D.J. Hill. Global boundedness of discrete-time adaptive control using parameter projection. In *Automatica*, volume 28(2), pages 1143–1158, 1992.

[55] B.E. Ydstie. Stability of discrete-time mrac revisited. In *Systems and Control Letters*, volume 13, pages 429–43, 1989.

[56] B.E. Ydstie. Transient performance and robustness of direct adaptive control. In *IEEE Transactions on Automatic Control*, volume 37(8), pages 1091–1105, 1992.

[57] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, Englewood Cliffs, New Jersey, USA, 1996.

[58] K. J. Åström and B. Wittenmark. *Adaptive Control.* Dover Publications, New York, second edition, 2013.

APPENDICES

Appendix A

Proof of Lemma 3.1

With $\delta > 0$ a positive constant and $\mathcal{X}$ a list of vectors $x_1, x_2, \ldots, x_m \in \mathbf{R}^n$ whose first element satisfies $\|x_1\| \geq \delta$, the method for constructing the basis set is as follows. Begin by defining the integer $i_1 := 1$. Now let $i = 1$. Trivially, choosing $\alpha_{(1,1)} = 0$ satisfies

$$\|x_1 - \alpha_{(1,1)} x_{i_1}\| = 0 \leq \delta.$$

Now increment i, and consider the minimizer of

$$\min_{\mu \in \mathbf{R}} \|x_i - \mu x_{i_1}\|^2. \tag{A.1}$$

This is a quadratic function, hence it is strictly convex, so a unique minimizer exists. Assume that the minimum is no greater than δ^2, then choose

$$\alpha_{(i,1)} = \arg\min_{\mu \in \mathbf{R}} \|x_i - \mu x_{i_1}\|^2,$$

thus,

$$\|x_i - \alpha_{(i,1)} x_{i_1}\| \leq \delta.$$

Continue iterating until $i = m$ or the minimum of (A.1) is larger than δ^2. If the former, then $\bar{n} = 1$ and the construction is complete. If the latter, then let $i_2 = i$, $\alpha_{(i_2,1)} = 0$, $\alpha_{(i_2,2)} = 1$, hence

$$\|x_{i_2} - \alpha_{(i_2,1)} x_{i_1} - \alpha_{(i_2,2)} x_{i_2}\| = 0 \leq \delta.$$

Continue incrementing i, and consider the minimizer of

$$\min_{\mu \in \mathbf{R}^2} \left\| x_i - \begin{bmatrix} x_{i_1} & x_{i_2} \end{bmatrix} \mu \right\|^2. \tag{A.2}$$

So long as the minimum is no more than δ^2, let

$$\begin{bmatrix} \alpha_{(i,1)} \\ \alpha_{(i,2)} \end{bmatrix} = \arg\min_{\mu \in \mathbf{R}^2} \left\| x_i - \begin{bmatrix} x_{i_1} & x_{i_2} \end{bmatrix} \mu \right\|^2,$$

thus,

$$\left\| x_i - \sum_{j=1}^{2} \alpha_{(i,j)} x_{i_j} \right\| \leq \delta.$$

Continue iterating until $i = m$ or the minimum of (A.2) is larger than δ^2. If the former, then $\bar{n} = 2$ and construction is complete. If the latter, then let $i_3 = i$, $\alpha_{(i_3,1)} = 0$, $\alpha_{(i_3,2)} = 0$ and $\alpha_{(i_3,3)} = 1$... continue until termination.

In doing so, for each $i \in \{1, \ldots, m\}$, we find a set of coefficients $\alpha_{(i,j)}$ such that x_i may be approximated by a subset of the basis vectors. Now by extending the domain of the coefficients by defining

$$\alpha_{(i,j)} = 0, \quad i = 1, \ldots, i_j - 1, \quad j = 2, \ldots, \bar{n},$$

then we can write

$$\left\| x_i - \sum_{j=1}^{\bar{n}} \alpha_{(i,j)} x_{i_j} \right\| \leq \delta. \tag{A.3}$$

Finally, to prove the coefficients are bounded, first let

$$y_{(i,k)} := \sum_{j=1}^{k} \alpha_{(i,j)} x_{i_j}, \quad i = 1, \ldots m, \quad k = 1, \ldots, \bar{n}. \tag{A.4}$$

We claim that

$$|\alpha_{(i,k)}| \leq \frac{\|y_{(i,k)}\|}{\delta}, \quad i = 1, \ldots m, \quad k = 1, \ldots, \bar{n}. \tag{A.5}$$

This clearly holds for $k = 1$ because then $\|y_{(i,1)}\| = |\alpha_{(i,1)}| \|x_{i_1}\|$, and we have asserted that $\|x_{i_1}\| \geq \delta$. Clearly (A.5) also holds if $\alpha_{(i,k)} = 0$. Suppose $\alpha_{(i,k)} \neq 0$; then

$$y_{(i,k)} = \alpha_{(i,k)} \left(x_{i_k} + \sum_{j=1}^{k-1} \frac{\alpha_{(i,j)}}{\alpha_{(i,k)}} x_{i_j} \right).$$

The construction of basis vectors ensures that the bracketed term has magnitude greater than δ, thus (A.5) is true. Next, write

$$\|y_{(i,\bar{n})}\| \leq \|y_{(i,\bar{n})} - x_i\| + \|x_i\|.$$

From (A.3), $\|y_{(i,\bar{n})} - x_i\| \leq \delta$, hence

$$\frac{\|y_{(i,\bar{n})}\|}{\delta} \leq 1 + \frac{\|\mathcal{X}\|}{\delta}. \tag{A.6}$$

From (A.4) we have $y_{(i,k-1)} = y_{(i,k)} - \alpha_{(i,k)} x_{i_k}$, $k = 2, \ldots, \bar{n}$, thus

$$\begin{aligned}
\frac{\|y_{(i,k-1)}\|}{\delta} &\leq \frac{\|y_{(i,k)}\|}{\delta} + \frac{|\alpha_{(i,k)}| \|x_{i_k}\|}{\delta} \\
&\leq \frac{\|y_{(i,k)}\|}{\delta} + \frac{\|y_{(i,k)}\| \|x_{i_k}\|}{\delta^2} \quad \text{(using (A.5))} \\
&\leq \left(1 + \frac{\|\mathcal{X}\|}{\delta}\right) \frac{\|y_{(i,k)}\|}{\delta}, \quad k = 2, \ldots, \bar{n}.
\end{aligned}$$

Iterating this and using (A.5) and (A.6) yields the desired result:

$$|\alpha_{(i,k)}| \leq \frac{\|y_{(i,k)}\|}{\delta} \leq \left(1 + \frac{\|\mathcal{X}\|}{\delta}\right)^{\bar{n}+1-k}, \quad k = 1, \ldots, \bar{n}. \qquad \square$$

Appendix B

Proof of Claim 4.1

In proving this claim, we use the following lemma.

Lemma B.1 *For any* $t \geq 1$, $\lambda \in \mathbf{R}$, *and signals* $x : \{1, \ldots, t\} \times \{1, \ldots, t\} \to \mathbf{R}$ *and* $y : \{1, \ldots, t\} \to \mathbf{R}$ *that satisfy*

$$\sum_{i=1}^{t} \lambda^{t-i} |x(t,i)|^2 \leq \sum_{i=1}^{t} \lambda^{t-i} |y(i)|^2 \tag{B.1}$$

and

$$\sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 \leq \sum_{i=1}^{j} \lambda^{j-i} |x(j+1,i)|^2, \quad j = 1, \ldots, t-1, \tag{B.2}$$

the following bound holds:

$$\sum_{i=1}^{t} \lambda^{t-i} |x(i,i)|^2 \leq \sum_{i=1}^{t} \lambda^{t-i} |y(i)|^2.$$

Proof:

$$
\begin{aligned}
\sum_{j=1}^{t} \lambda^{t-j} |x(j,j)|^2 &= \sum_{j=1}^{t} \lambda^{t-j} \underbrace{\left(\sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 - \sum_{i=1}^{j-1} \lambda^{j-i} |x(j,i)|^2 \right)}_{|x(j,j)|^2} \\
&= \sum_{i=1}^{t} \lambda^{t-i} |x(t,i)|^2 + \sum_{j=1}^{t-1} \lambda^{t-j} \sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 - \sum_{j=2}^{t} \lambda^{t-j} \sum_{i=1}^{j-1} \lambda^{j-i} |x(j,i)|^2 \\
&= \sum_{i=1}^{t} \lambda^{t-i} |x(t,i)|^2 + \sum_{j=1}^{t-1} \lambda^{t-j} \sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 \\
&\quad - \sum_{j=1}^{t-1} \lambda^{t-j} \sum_{i=1}^{j} \lambda^{j-i} |x(j+1,i)|^2 \\
&\leq \sum_{i=1}^{t} \lambda^{t-i} |x(t,i)|^2 + \sum_{j=1}^{t-1} \lambda^{t-j} \sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 \\
&\quad - \sum_{j=1}^{t-1} \lambda^{t-j} \sum_{i=1}^{j} \lambda^{j-i} |x(j,i)|^2 \quad \text{(using (B.2))} \\
&\leq \sum_{i=1}^{t} \lambda^{t-i} |y(i)|^2 \quad \text{(using (B.1))} \qquad \square
\end{aligned}
$$

Proof of Claim 4.1: Apply Lemma B.1 with

$$
t \Leftarrow t - t_0, \quad x(j,i) \Leftarrow e_{\hat{\theta}(j+t_0)}(i + t_0), \quad y(i) \Leftarrow w(i + t_0 - 1). \qquad \square
$$

Appendix C

Matlab Code

Here we provide Matlab code for simulating a first order system using the techniques demonstrated in this thesis. It may also be found online at [25]. The first section is the main loop which runs the simulation, the following sections are additional classes used by the main loop, including both the Supervisory Estimator and the Projection Algorithm estimator, as well as both types of control laws used in this thesis.

C.1 Main Loop

```
n = 250; % how long to run simulation
paramRange = [1, 4; 1, 4]; % the set S
Na = 1; %1st order
Nb = 1;

% Constant Parameters
% a = [1;1];
% b = [2;1];

% Time-Varying Parameters
a = 2.5 + 1.5*cos(0.01*pi*(0:n-1));
b = 2.5 + 1.5*sin(0.003*pi*(0:n-1));

% Disturbance
```

```
w = 0.1*randn(n,1);

% initial condition for plant; rightmost column is t_0, others
    are previous history
ic = zeros(2,2);
plant = linSystem(ic, a, b);

%% Estimator Selection
initialGuess = mean(paramRange, 2); % initial guess for
   theta_hat
estimator = SupervisoryEstimator(paramRange, initialGuess,
   0.6, 1, false);
% Last 3 params are lambda, dwell time, data filter on/off
% estimator = ProjectionEstimator(paramRange, initialGuess);

%% Certainty Equivalence Controller Selection
controller = dStepAhead(size(a,1), size(b,1));
% controller = integralPolePlacement(size(a,1), size(b,1));

%% Reference
r = sin(0.2*pi*(0:n+10));

%% Main Loop
for t = 1:n-1
  plant.update(w(t)); % find next y using phi and w
  theta_hat = estimator.estimate(plant);
  u = controller.control(theta_hat, plant, r, t+1);
  plant.next_input(u);
end
```

C.2 Plant Model

```
classdef linSystem<handle
  properties
    phi
    transferfunction
```

```
    a
    b
    Na
    Nb
    t
    t_offset
  end
  methods
    function obj = linSystem(ic, a, b)
      obj.phi = ic;
      obj.a = a;
      obj.b = b;
      obj.Na = size(a,1);
      obj.Nb = size(b,1);
      obj.transferfunction = tf(b(:, 1)', [1 -a(:, 1)'], 1); %
          True plant tf
      obj.t = 1;
      obj.t_offset = size(ic, 2) - 1;
    end
    function update(obj, w)
      if (size(obj.a,2) == 1) % if constant parameters
        theta = [obj.a; obj.b];
      else % if time-varying parameters
        theta = [obj.a(:,obj.t); obj.b(:,obj.t)];
        obj.transferfunction = tf(obj.b(:, obj.t)', [1 -obj.a
            (:, obj.t)'], 1); % True plant tf
      end
      A = diag(ones(obj.Na+obj.Nb-1,1), -1);
      A(obj.Na+1, :) = NaN;
      A(1,:) = theta';
      B = [1; zeros(obj.Na + obj.Nb - 1, 1)];
      obj.phi(:, end+1) = A*obj.phi(:,end) + B*w;

      obj.t = obj.t+1;
    end

    function next_input(obj, u)
      obj.phi(obj.Na+1, end) = u;
```

```
    end

    function print_phi(obj)
      fprintf("  t,  y,    u\n");
      fprintf("----------------------\n");
      fprintf("%3d, %8.4f, %8.4f \n", [(1-obj.t_offset: size(
         obj.phi, 2) - obj.t_offset); obj.phi(1, :); obj.phi(
         obj.Na+1, :)]);
    end
  end
end
```

C.3 Supervisory Estimator

```
classdef SupervisoryEstimator < handle
  properties
    theta_hat
    paramRange
    W
    lambda
    dwellTimer
    dwellTime
    filter
  end
  methods
    function obj = SupervisoryEstimator(p_range, initialGuess,
        lambda, dwellTime, filter)
      obj.paramRange = p_range;
      obj.theta_hat = initialGuess;
      obj.W = zeros(length(initialGuess)+1, length(
         initialGuess)+1);
      obj.lambda = lambda;
      obj.dwellTime = dwellTime;
      obj.dwellTimer = dwellTime-1;
      obj.filter = filter;
    end
```

```
function t_hat = estimate(obj, plant)
  if obj.filter % disturbance annihilation filter
    y = plant.phi(1, end) - plant.phi(1, end-1);
    phi = plant.phi(:, end-1) - plant.phi(:, end-2);
  else
    y = plant.phi(1, end);
    phi = plant.phi(:, end-1);
  end

  obj.W(:,:,end+1) = obj.lambda*obj.W(:,:,end) + [phi;y]*[
     phi' y];
  w = obj.W(:,:,end);

  if (obj.dwellTimer > 0) % if dwelling
    obj.dwellTimer = obj.dwellTimer - 1;
    obj.theta_hat(:, end+1) = obj.theta_hat(:, end);
  else % if not dwelling
    H = 2*[w(1,1) w(2,1); w(2,1) w(2,2)];
    f = [-2*w(3,1); -2*w(3,2)];
%       offset = w(3,3);
    A = [-1 0; 1 0; 0 -1; 0 1];
    b = [-obj.paramRange(1, 1); obj.paramRange(1, 2); -obj
       .paramRange(2, 1); obj.paramRange(2, 2)];

                              % Optimize for positive HF
                                 gain
    options = optimoptions('quadprog','Display','off');
    x = quadprog(H,f,A,b,[],[],[],[],[],options);

    b = [-obj.paramRange(1, 1); obj.paramRange(1, 2); obj.
       paramRange(2, 2); -obj.paramRange(2, 1)];
                              % Optimize for negative HF
                                 gain
    x2 = quadprog(H,f,A,b,[],[],[],[],[],options);

                              % Choose the best of the two
    if (0.5*x2'*H*x2 + f'*x2 < 0.5*x'*H*x + f'*x)
```

```
        x = x2;
      end

      if (0.5*x'*H*x + f'*x < 0.5*obj.theta_hat(:, end)'*H*
          obj.theta_hat(:, end) + f'*obj.theta_hat(:, end))
        obj.theta_hat(:, end+1) = x;
        obj.dwellTimer = obj.dwellTime - 1; % Resume
            dwelling
      else
        obj.theta_hat(:, end+1) = obj.theta_hat(:, end);
      end
    end
    t_hat = obj.theta_hat(:, end);
   end
  end
end
```

C.4 Projection Algorithm Estimator

```
classdef ProjectionEstimator < handle
  properties
    theta_hat
    paramRange
    filter
  end
  methods
    function obj = ProjectionEstimator(p_range, initialGuess,
       filter)
      obj.paramRange = p_range;
      obj.theta_hat = initialGuess;
      obj.filter = filter;
    end
    function t_hat = estimate(obj, plant)
      if obj.filter % disturbance annihilation filter
        y = plant.phi(1, end) - plant.phi(1, end-1);
        phi = plant.phi(:, end-1) - plant.phi(:, end-2);
```

```
      else
        y = plant.phi(1, end);
        phi = plant.phi(:, end-1);
      end
      e = y - phi'*obj.theta_hat(:, end); % prediction error

      if (norm(phi)^2 == 0) % phi = 0, do nothing
        obj.theta_hat(:, end+1) = obj.theta_hat(:, end);
      else
        obj.theta_hat(:, end+1) = obj.project( obj.theta_hat
           (:, end) + phi/(norm(phi)^2)*e );
      end
      t_hat = obj.theta_hat(:, end);
    end

    function y = project(obj, theta)
      y = max(min(theta, obj.paramRange(:,2)), obj.paramRange
         (:,1));
    end
  end
end
```

C.5 d-Step-Ahead Control Law

```
classdef dStepAhead<handle
  properties
    Na
    Nb
  end
  methods
    function obj = dStepAhead(Na, Nb)
      obj.Na = Na;
      obj.Nb = Nb;
    end
    function u = control(obj, theta, plant, r, t)
      A = diag(ones(obj.Na+obj.Nb-1,1), -1);
```

```
      A(obj.Na+1, :) = NaN;
      A(1,:) = theta';
      phi = plant.phi(:,end);
      d = 1;

      while(theta(obj.Na+d) == 0)
        d = d+1;
        phi(obj.Na+1) = 0;
        phi = A*phi;
      end
      u = r(t+d);

      for i = 1 : obj.Na
        u = u - theta(i)*phi(i);
      end
      for i = obj.Na+d+1 : obj.Na+obj.Nb
        u = u - theta(i)*phi(i);
      end
      u = u/theta(obj.Na+d);
    end
  end
end
```

C.6 Pole Placement Control Law

```
classdef integralPolePlacement <handle
  properties
    Na
    Nb
  end
  methods
    function obj = integralPolePlacement(Na, Nb)
      obj.Na = Na;
      obj.Nb = Nb;
    end
    function u = control(obj, theta, plant, r, t)
```

```
      varphi = zeros(obj.Na+obj.Nb+1, 1);
      varphi(obj.Na+2:end) = plant.phi(obj.Na+1:end,end-1) -
         plant.phi(obj.Na+1:end,end-2);

      R = r(1)*ones(obj.Na+1, 1);
      for i = 1:obj.Na+1
        try
          R(i) = r(t-i);
        catch
        end
      end
      varphi(1:obj.Na+1) = [plant.phi(1,end-1); plant.phi(1:
         obj.Na,end-2)] - R;

      syms q 'real';

      A = 1-q.^(1:obj.Na)*theta(1:obj.Na);
      B = q.^(1:obj.Nb)*theta(obj.Na+1:end);
      l = sym('l_', [obj.Nb, 1], 'real');
      p = sym('p_', [obj.Na+1, 1], 'real');
      L = 1-q.^(1:obj.Nb)*l;
      P = -q.^(1:obj.Na+1)*p;

      charpoly = (1-q)*A*L + B*P;
      S = solve(coeffs(charpoly, q) == [1 zeros(1, obj.Na+obj.
         Nb+1)]);
      result = [subs(p, S); subs(l, S)];

      last_u = plant.phi(plant.Na+1,end-1);
      delta_u = result'*varphi;
      u = last_u + delta_u;
    end
  end
end
```