EE5104 Adaptive Control Systems

AY2021/2022 Semester 2

Continuous Assessment 1

Philippe Andrin Brigger
A0248834U

Contents

1 Introduction **3**

2 Adaptive Control System Design **4**
2.1 Design Requirements 4
2.2 Controller Design 4
2.2.1 Perfect Control Input 4
2.2.2 Adaptive Control Input 7

3 Simulation Results **12**
3.1 Perfect Control Input 12
3.2 Adaptive Control Input 12
3.2.1 Square Wave Reference Signal 15
3.2.2 Sinusoidal Reference Signal 16

4 Conclusion **18**

A Appendix **20**
A.1 Simulink Model 21
A.2 Simulation Script 22

1 Introduction

This *CA1* report is part of the spring 2022 module EE5104 Adaptive Control Systems at the National University of Singapore (NUS). Adaptive control system design is a design methodology which can be used to control systems with initially uncertain or varying parameters. The control law non-linearly adapts the controller gains to match the system output to a reference model output. The task of *CA1* is to design and simulate an adaptive controller for a second order system with unknown parameters. The first part of this report details the design process of the adaptive control system. As the adaptive control approach is non-linear, and this bears the risk of signals diverging at an arbitrary speed, the first part of the report focuses on rigorously showing that the tracking error of the system converges to zero and that all signals remain bounded. Next, the report discusses the simulation results of the proposed controller and examines the effects that different design choices have on the performance of the system. Finally, the appendix contains the code used to run the simulations. In general, all vectors are printed in bold while scalars and matrices are printed as normal text throughout the report.

2 Adaptive Control System Design

2.1 Design Requirements

In *CA1*, an adaptive controller for a second order continuous-time system $\Sigma(s)$ is designed. The system $\Sigma(s)$ has the following form:

$$\Sigma(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s + b_1}{s^2 + a_1 s + a_2} \tag{1}$$

The exact coefficients of the plant are unknown and only the plant output $y(t)$ and input $u(t)$ are measurable. It is, however, known that b_0 is a negative number and that the plant zero does not lie in the right half of the s-plane. As the zeros of a plant are found by equating the plant numerator to zero, the knowledge of the plant's zero location ζ_0 constrains that b_1 must also be negative:

$$\begin{aligned} & b_0\zeta_0 + b_1 = 0 \Rightarrow \zeta_0 = -\frac{b_1}{b_0} = -\frac{\text{sign}(b_1)|b_1|}{\text{sign}(b_0)|b_0|} < 0 \\ & \Rightarrow \text{sign}(b_1) < 0, \quad \text{since} \quad \text{sign}(b_0) < 0 \end{aligned}$$

The design of the adaptive controller must make the system response meet the following two requirements:

- Boundedness of $y(t)$ and $u(t)$ $\forall\, t$
- Zero steady-state error

For instance, a strictly positive real, first order system $H_m(s)$ may fulfill these two requirements:

$$H_m(s) = \frac{1}{\tau s + 1} = \frac{1/\tau}{s + 1/\tau} \tag{2}$$

This system will be referred to as the reference model and all quantities with subscript m are quantities of the reference model. To meet the requirement of bounded input and output signals, the pole π_m of $H_m(s)$ must have a negative real part and thus the time constant τ of $H_m(s)$ must be greater than zero. Following the final value theorem [2], $H_m(s)$ will not have a steady-state error. Mathematically expressed:

- Boundedness of y(t) and u(t) iff: $\text{Re}(\pi_m) < 0 \Rightarrow \pi_m = -\dfrac{1}{\tau} \Rightarrow \underline{\tau > 0}$
- Steady-state error: Final value theorem: $\underline{y(\infty)} = \lim_{s\to 0} sY(s) = \lim_{s\to 0} s\dfrac{1}{\tau s + 1} \underline{= 0}$

The choice of the time constant τ of $H_m(s)$ determines the system response behavior. Generally speaking, the smaller τ the quicker the system response $H_m(s)$ will be, since the pole of H_m will lie farther left in the s-plane. To conclude, if a controller can be designed in a way to make the plant $\Sigma(s)$ behave the same way as $H_m(s)$, then it can be guaranteed that the system response has bounded input and output signals and that the steady-state error converges to zero. The adaptive controller essentially aims to make the system response of the uncertain plant $\Sigma(s)$ be the same as the system response of the desired reference model $H_m(s)$.

2.2 Controller Design

2.2.1 Perfect Control Input

The controller is designed to make $\Sigma(s)$ behave like $H_m(s)$. To begin with the controller design, the system described in eq. (1) can be converted from the frequency domain to the time domain via inverse Laplace transform and written in the following form:

$$R_p(\frac{d}{dt})y(t) = k_p Z_p(\frac{d}{dt})u(t) \tag{3}$$

R_p is a monic, pole polynomial of degree $n = 2$, while Z_p is a monic, zero polynomial of degree $m = 1$. Furthermore, k_p is a negative gain (recall that it is known that b_0 is negative) that ensures that both polynomials are monic. The subscript p denotes that these are quantities of the original plant $\Sigma(s)$.

$$\begin{aligned}
R_p(\frac{d}{dt}) &= \frac{d^2}{dt^2} + a_1\frac{d}{dt} + a_2 \\
Z_p(\frac{d}{dt}) &= \frac{d}{dt} + \frac{b_1}{b_0} \\
k_p &= b_0 < 0
\end{aligned}$$

Therefore, it can be said that the relative degree of the plant is $n^* = n - m = 1$. Doing the same with $H_m(s)$, and writing r for the reference signal instead of the input u, gives:

$$R_m(\frac{d}{dt})y(t) = k_m Z_m(\frac{d}{dt})r(t) \tag{4}$$

As in eq. (3), R_m and Z_m are the monic pole respectively zero polynomials of $H_m(s)$ and k_m is the gain that ensures both polynomials are monic. Thanks to the design considerations mentioned in section 2.1, it can further be said that R_m is asymptotically stable (if $\tau > 0$, then the pole has a negative real part) and a polynomial of degree one.

$$\begin{aligned}
R_m(\frac{d}{dt}) &= \frac{d}{dt} + \frac{1}{\tau} \\
Z_m(\frac{d}{dt}) &= 1 \\
k_m &= \frac{1}{\tau}
\end{aligned}$$

Next, the Diophantine identity from the lecture notes can be considered.

$$T(\frac{d}{dt})R_m(\frac{d}{dt}) = R_p(\frac{d}{dt})E(\frac{d}{dt}) + F(\frac{d}{dt}) \tag{5}$$

In eq. (5), T is an asymptotically stable and monic polynomial which, as with R_m, is a design choice. T is chosen to have order $n = 2$.

$$T = \frac{d^2}{dt^2} + t_1\frac{d}{dt} + t_2 \tag{6}$$

In order for T to be asymptotically stable, the roots $s_{1,2}$ of T must all have negative real part. As a result, t_1 and t_2 must both be positive:

$$\begin{aligned}
s_{1,2} &= \frac{-t_1 \pm \sqrt{t_1^2 - 4t_2}}{2} \Rightarrow \mathrm{Re}(s_{1,2}) < 0 \Rightarrow \underline{t_1 > 0} \\
&\Rightarrow -t_1 + \sqrt{t_1^2 - 4t_2} < 0 \Rightarrow t_1^2 - 4t_2 < t_1^2 \Rightarrow \underline{t_2 > 0}
\end{aligned}$$

Once T and R_m have both been defined, and an initial guess of R_p exists, E and F are unique. They will both be of first order, and E will be monic. By multiplying both sides of eq. (3) with E, and inserting the information obtained from the Diophanite identity in eq. (5), eq. (3) can be rewritten as follows:

$$\begin{aligned}
ER_p y = (TR_m - F)y &= k_p E Z_p u \\
\Rightarrow R_m y &= \frac{F}{T}y + k_p\frac{EZ_p}{T}u
\end{aligned}$$

If the input u is chosen appropriately, the above equation will be equivalent to eq. (4). This, in turn, means that the output y of the actual plant will behave the same as the output y_m of the reference system. To conclude, given any bounded input $u(t)$, the system will have a bounded output $y(t)$ and will not have a steady-state error. To find this appropriate input $u(t)$, let us define $\bar{F} = F/k_p$. $\bar{F}$ will still be a polynomial of first order.

$$\bar{F} = f_1 \frac{d}{dt} + f_2$$

Furthermore, let us define $\bar{G} = EZ_p = T + G_1$. Since $\bar{G}$ and T are both second order, monic polynomials, G_1 will be a first order polynomial.

$$G_1 = g_1 \frac{d}{dt} + g_2$$

These definitions yield:

$$R_m y = k_p \Big(\frac{\bar{F}}{T} y + \frac{G_1}{T} u + u \Big) \tag{7}$$

By comparing the plant output $y(t)$ of eq. (7) with the reference output $y_m(t)$ of eq. (4), the ideal control input $u^*(t)$ results as:

$$u^*(t) = -\frac{G_1}{T} u - \frac{\bar{F}}{T} y + \frac{k_m}{k_p} r \tag{8}$$

This input would make the plant output $y(t)$ respond the same as the reference model output y_m.

$$R_m y = k_p \Big(\frac{\bar{F}}{T} y + \frac{G_1}{T} u - \frac{\bar{F}}{T} y - \frac{G_1}{T} u + \frac{k_m}{k_p} r \Big) = k_m r = R_m y_m$$

Next, the influence of $\frac{G_1}{T}$ on u and $\frac{\bar{F}}{T}$ on y must be understood. First, let $\omega_u = u/T$ define a new state. With eq. (6), ω_u can be written as:

$$\ddot{\omega}_u + t_1 \dot{\omega}_u + t_2 \omega_u = u$$

Or, in state-space form:

$$\begin{bmatrix} \dot{\omega}_u \\ \ddot{\omega}_u \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix} \begin{bmatrix} \omega_u \\ \dot{\omega}_u \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{9}$$

Secondly, let $\omega_y = y/T$ define a second new state. With eq. (6) and the same considerations as for ω_u, ω_y can be written in state-space form as:

$$\begin{bmatrix} \dot{\omega}_y \\ \ddot{\omega}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix} \begin{bmatrix} \omega_y \\ \dot{\omega}_y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} y \tag{10}$$

The new states ω_u, ω_y can be inserted into the terms $-\frac{G_1}{T} u$, $-\frac{\bar{F}}{T} y$:

$$-\frac{G_1}{T} u = -G_1 \omega_u = -g_1 \dot{\omega}_u - g_2 \omega_u$$
$$-\frac{\bar{F}}{T} y = -\bar{F} \omega_y = -f_1 \dot{\omega}_u - f_2 \omega_u$$

Inserting this into eq. (8) and defining $k^* = k_m/k_p$ gives:

$$u^* = \bar{\boldsymbol{\theta}}^{*T}\bar{\boldsymbol{\omega}} = \begin{bmatrix} \text{-}f_2 & \text{-}f_1 & \text{-}g_2 & \text{-}g_1 & k^* \end{bmatrix} \begin{bmatrix} \omega_y & \dot{\omega}_y & \omega_u & \dot{\omega}_u & r \end{bmatrix}^T \tag{11}$$

$$R_m y = k_p\Big(\bar{F}\omega_y + G_1\omega_u + \bar{\boldsymbol{\theta}}^{*T}\bar{\boldsymbol{\omega}}\Big) \tag{12}$$

At this point, it can be noted that the state vector $\bar{\omega}$ is a vector of dimension $2n+1$. Furthermore, the $2n$ states $\omega_u, \dot{\omega}_u, \omega_y, \dot{\omega}_y$ form a non-minimal state space representation of the system $\Sigma(s)$. If T and R_m have been designed, and the plant is known (specifically R_p, Z_p and k_p are known), the perfect, constant controller gains $\bar{\theta}^*$ can be calculated from eq. (5) with polynomial division and coefficient comparison.

2.2.2 Adaptive Control Input

Unfortunately, the plant parameters are not known and the ideal control input u^* from eq. (11) cannot be used directly. Hence, the control parameters cannot directly be calculated with the Diophantine identity (eq. (5)). To solve this problem, time-varying (adaptive) gains $\bar{\theta}(t)$ instead of constant gains $\bar{\theta}^*$ may be used. Inserting $u = \bar{\theta}^T\bar{\omega}$ in eq. (7) gives:

$$\begin{aligned}
R_m y &= k_p\Big(\frac{\bar{F}}{T}y + \frac{G_1}{T}u + \bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\omega}}\Big)\\
&= k_p\Big(\frac{\bar{F}}{T}y + \frac{G_1}{T}u - k^*r + k^*r + \bar{\theta}^T\bar{\omega}\Big)\\
&= k_p\Big(k^*r + \bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\omega}} - \bar{\boldsymbol{\theta}}^{*T}\bar{\boldsymbol{\omega}}\Big)\\
\Rightarrow R_m y &= k_m r + k_p\bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\omega}}
\end{aligned}$$

$\bar{\boldsymbol{\Phi}} = \bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}^*$ is the controller gain error signal. The output error dynamics $e_1 = y - y_m$ are found by subtracting $R_m y_m = k_m r$ from the equation above:

$$R_m e_1 = R_m(y - y_m) = \big(k_m r + k_p\bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\omega}}\big) - \big(k_m r\big) = k_p\bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\omega}} \tag{13}$$

As eq. (13) shows, the output error dynamics solely depend on the controller gain error signal $\bar{\boldsymbol{\Phi}}$ and the states $\bar{\boldsymbol{\omega}}$. Correspondingly, the controller gain error signal $\bar{\boldsymbol{\Phi}}$ depends solely on the adaptive controller gains $\bar{\boldsymbol{\theta}}$, a design choice, and the constant, ideal controller gains $\bar{\boldsymbol{\theta}}^*$. To conclude, an output error of zero can be achieved by designing the adaptive gains $\bar{\boldsymbol{\theta}}(t)$ in a way that the product of $\bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\omega}}$ will converge to zero. Hence, the plant output will match the reference model output.
To design $\bar{\boldsymbol{\theta}}(t)$ properly, the plant and reference model are written as a non-minimal realization. The state vector used for the realization is $\boldsymbol{\omega} = [\omega_y \ \dot{\omega_y} \ \omega_u \ \dot{\omega_u}]^T$. The states $\omega_y = y/T$, $\omega_u = u/T$ are defined analogously to section 2.2.1. To obtain the non-minimal realizations, both sides of eq. (3) and eq. (4) need to be multiplied with $1/T$. For the plant results to:

$$\begin{aligned}
R_p(\frac{d}{dt})\frac{y(t)}{T} &= k_p Z_p(\frac{d}{dt})\frac{u(t)}{T}\\
\Rightarrow R_p\omega_y &= k_p Z_p\omega_u
\end{aligned}$$

Inserting the definitions of R_p, Z_p yields:

$$\begin{aligned}
(\frac{d^2}{dt^2} + a_1\frac{d}{dt} + a_2)\omega_y &= \big(b_0\frac{d}{dt} + b_1)\omega_u\\
\Rightarrow \frac{d^2}{dt^2}\omega_y &= -a_1\frac{d}{dt}\omega_y - a_2\omega_y + b_0\frac{d}{dt}\omega_u + b_1\omega_u
\end{aligned}$$

This equation can be rewritten in state space form, when considering that $y = T\omega_y = \big(T - R_p\big)\omega_y + k_p Z_p \omega_u$:

$$\begin{bmatrix}\dot{\omega_y}\\ \ddot{\omega_y}\\ \dot{\omega_u}\\ \ddot{\omega_u}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ \text{-}a_2 & \text{-}a_1 & b_1 & b_0\\ 0 & 0 & 0 & 1\\ 0 & 0 & \text{-}t_2 & \text{-}t_1\end{bmatrix}\begin{bmatrix}\omega_y\\ \dot{\omega_y}\\ \omega_u\\ \dot{\omega_u}\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\end{bmatrix}u$$

$$y = \begin{bmatrix}t_2 - a_2 & t_1 - a_1 & b_1 & b_0\end{bmatrix}\begin{bmatrix}\omega_y\\ \dot{\omega_y}\\ \omega_u\\ \dot{\omega_u}\end{bmatrix}$$

$$\Rightarrow \dot{\boldsymbol{\omega}} = A_p\boldsymbol{\omega} + B_p u$$

$$y = C_p^T\boldsymbol{\omega}$$

Similarly, the reference model can also be written as a non-minimal state space model using T:

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{m}} = A_m\boldsymbol{\omega}_{\boldsymbol{m}} + B_m r$$

$$y_m = C_m^T\boldsymbol{\omega}_{\boldsymbol{m}}$$

The objective is to have a control signal that makes the non-minimal state of the plant $\boldsymbol{\omega}$ converge to the non-minimal state $\boldsymbol{\omega}_{\boldsymbol{m}}$ of the reference model. In the following, it will be shown that y will converge to y_m and thus $e_1 = y - y_m$ will converge to zero if $\boldsymbol{\omega}$ converges to $\boldsymbol{\omega}_{\boldsymbol{m}}$. Therefore, instead of analyzing the output error e_1, the state error $\boldsymbol{e_2} = \boldsymbol{\omega} - \boldsymbol{\omega}_{\boldsymbol{m}}$ can also be used to determine the error convergence of the system. One approach to find a suitable control signal is with Lyapunov's Direct Method. Lyapunov's Direct Method says that if a scalar valued function $V(\boldsymbol{\omega}, t)$ of a system's state variables $\boldsymbol{\omega}$ that satisfies certain conditions can be found, then the equilibrium point of a (nonlinear) system will be asymptotically stable [3]. In other words, if a Lyapunov function $V(\boldsymbol{\omega}, t)$ can be found, then the system defined in eq. (13) will have an asymptotically stable equilibrium point at $\boldsymbol{\omega} = 0$. This implies that the state error e_2 will converge to zero. It will later be shown that the output error e_1 will also converge to zero. Note here that $\boldsymbol{\omega}$ is a subset of the full state vector $\bar{\boldsymbol{\omega}}$. The state that is missing in $\boldsymbol{\omega}$ is the reference signal r. Since the reference signal r is the same for both the reference model and the plant, this state will not have an influence on the error convergence of the system since . Therefore, it suffices to show that $\boldsymbol{e_2} = \boldsymbol{\omega} - \boldsymbol{\omega}_{\boldsymbol{m}}$ converges to zero since this implicitly implies that the error $\boldsymbol{e} = \bar{\boldsymbol{\omega}} - \bar{\boldsymbol{\omega}}_{\boldsymbol{m}}$ also converges to zero. Hence, finding a Lyapunov function $V(\boldsymbol{\omega}, t)$ guarantees that the state errors $\boldsymbol{e_2} = \boldsymbol{\omega} - \boldsymbol{\omega}_{\boldsymbol{m}}$ as well as $\boldsymbol{e} = \bar{\boldsymbol{\omega}} - \bar{\boldsymbol{\omega}}_{\boldsymbol{m}}$ of the system converge to zero. The four conditions that a Lyapunov function $V(\boldsymbol{\omega}, t)$ must satisfy are:

- $V(\boldsymbol{\omega}, t)$ is positive-definite
- $V(\boldsymbol{\omega}, t)$ is decrescent
- $\dot{V}(\boldsymbol{\omega}, t)$ is negative semi-definite
- $V(\boldsymbol{\omega}, t)$ is radial unbounded

As will be shown, one function which satisfies these four conditions is:

$$V(\omega, t) = \boldsymbol{e_2}^T P\boldsymbol{e_2} + \bar{\boldsymbol{\Phi}}^T\Gamma^{-1}\bar{\boldsymbol{\Phi}} \tag{14}$$

This function is a function of the non-minimal realization system states $\boldsymbol{\omega}$ and of time t since it will be shown that $\bar{\boldsymbol{\Phi}}(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}^*)$ is a function of the system states $\boldsymbol{\omega}$ and

t, and the error $\boldsymbol{e_2}$ is a function of $\boldsymbol{\omega}$ and time:

$$\begin{aligned}\bar{\boldsymbol{\Phi}} &= \bar{\boldsymbol{\Phi}}(\boldsymbol{\omega}, r, t) = \bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}^* = \bar{\boldsymbol{\theta}}(\boldsymbol{\omega}, r, t) - \bar{\boldsymbol{\theta}}^*(\text{constant}) \\ \boldsymbol{e_2} &= \boldsymbol{e_2}(\boldsymbol{\omega}, t) = \boldsymbol{\omega} - \boldsymbol{\omega_m}\end{aligned}$$

Furthermore, in eq. (14), P is the positive definite solution to the Algebraic Riccati equation $A_m^T P + PA_m = -Q$ with Q being any symmetric positive definite matrix. P exists since $\boldsymbol{A_m}$ itself is an asymptotically stable matrix (The reference model was designed to be asymptotically stable as the desired plant response should be asymptotically stable). Finally, Γ is a positive definite gain matrix which can be chosen by the engineer.
The first condition of positive-definiteness is easily proven. Per definition, a positive definite matrix M fulfills the following equation:

$$\boldsymbol{x}^T M \boldsymbol{x} > 0 \quad \forall \boldsymbol{x} \neq \mathbf{0}$$

Since P and Γ are positive definite matrices, and $\bar{\boldsymbol{\Phi}} \neq \mathbf{0}$, V must be positive definite itself. Mathematically speaking, for a function $\alpha(\|\boldsymbol{\omega}\|)$ with $\alpha(0) = 0$:

$$V(\boldsymbol{\omega}, t) \geq \alpha(\,\|\|\omega) > 0 \quad \forall t,\ \forall \boldsymbol{\omega} \neq 0,\ \text{positive definite}$$

Furthermore, it can be said that $V(\boldsymbol{\omega}, t)$ is radial unbounded because, following the lecture notes, $\alpha(\boldsymbol{\omega})$ can be chosen such that i.e.:

$$\alpha(\|\boldsymbol{\omega}\|) \to \infty \text{ as } \|\boldsymbol{\omega}\| \to \infty$$

Next, if $\dot{V}(\boldsymbol{\omega}, t)$ can be shown to be negative semi-definite, $V(\boldsymbol{\omega}, t)$ will consequently also be decrescent since a function that never grows in value is upper-bounded (which is the case if the slope of a function is negative semi-definite). This would then conclude the search for a Lyapunov function that satisfies the four conditions and would give a control law that makes eq. (13) an asymptotically stabe system. Hence, the state error $\boldsymbol{e_2}$ would converge to zero. To show negative semi-definiteness, $V(\boldsymbol{\omega}, t)$ is differentiated with respect to time:

$$\dot{V}(\boldsymbol{\omega}, t) = 2\boldsymbol{e_2}^T P \dot{\boldsymbol{e}}_{\boldsymbol{2}} + 2\bar{\boldsymbol{\Phi}}^T \Gamma^{-1} \dot{\bar{\boldsymbol{\Phi}}} \tag{15}$$

$\dot{\boldsymbol{e}}_{\boldsymbol{2}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_{\boldsymbol{m}}$ can be found by inserting a rewritten form of the control input $u = \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\omega}}$ into the non-minimal state-space description of the plant and by comparing this state-space description with the reference model state-space description.

$$\begin{aligned} u(t) &= \bar{\boldsymbol{\theta}}^T(t)\bar{\boldsymbol{\omega}}(t) = \boldsymbol{\theta}^T(t)\boldsymbol{\omega}(t) + k(t)r(t) \\ &= \big[\boldsymbol{\theta}^{*T} + \boldsymbol{\phi}^T(t)\big]\boldsymbol{\omega}(t) + \big[k^* + \phi_k(t)\big]r(t) \\ \Rightarrow \dot{\boldsymbol{\omega}} &= A_p\boldsymbol{\omega} + B_p\Big\{\big[\boldsymbol{\theta}^{*T} + \boldsymbol{\phi}^T(t)\big]\boldsymbol{\omega}(t) + \big[k^* + \phi_k(t)\big]r(t)\Big\} \\ \dot{\boldsymbol{\omega}} &= \Big\{A_p + B_p\boldsymbol{\theta}^{*T}\Big\}\boldsymbol{\omega}(t) + k^* B_p r(t) + B_p\boldsymbol{\phi}^T(t)\boldsymbol{\omega}(t) + B_p\phi_k(t)r(t) \\ \Rightarrow y_p &= C_p^T\boldsymbol{\omega} \end{aligned}$$

Since the reference model behaves the same way as the plant with perfect controller gains $\bar{\boldsymbol{\theta}}^{*T}$ (thus $\bar{\boldsymbol{\Phi}} = \bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}^* = \mathbf{0}$), the state space matrices A_m, B_m, C_m of the reference model must be the same as the plant state space matrices A_p, B_p, C_p with the perfect controller gains inserted. These matrices are derived in the equations above by inserting $u(t) = \bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\omega}}$ and splitting the equation into one part only dependent of the perfect controller gains $\bar{\boldsymbol{\theta}}^*$ and another part only dependent of the controller gain error $\bar{\boldsymbol{\Phi}}$. This gives:

$$
\begin{aligned}
A_m &= A_p + B_p \boldsymbol{\theta}^{*T} \\
B_m &= k^* B_p \\
C_m &= C_p
\end{aligned}
$$

Thanks to this, $\dot{\boldsymbol{e_2}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega_m}}$ results as:

$$
\begin{aligned}
\dot{\boldsymbol{e_2}} &= A_m \boldsymbol{\omega} + B_m r + B_p \boldsymbol{\phi}^T \boldsymbol{\omega} + B_p \phi_k r - A_m \boldsymbol{\omega_m} - B_m r \\
&= A_m \big(\boldsymbol{\omega} - \boldsymbol{\omega_m}\big) + \frac{1}{k^*} B_m \bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\omega}} \\
&= A_m \boldsymbol{e_2} + \frac{1}{k^*} B_m \bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\omega}}
\end{aligned}
$$

Furthermore, $C_m = C_p$ implies and shows that the output error e_1 must converge to zero if the state error $\boldsymbol{e_2}$ converges to zero.

$$
\begin{aligned}
& e_1 = y - y_m = C_p^T \boldsymbol{\omega} - C_m^T \boldsymbol{\omega_m} = C_m^T (\boldsymbol{\omega} - \boldsymbol{\omega_m}) \\
\Rightarrow\ & e_1 = C_m^T \boldsymbol{e_2}
\end{aligned}
$$

The new found equation for $\dot{\boldsymbol{e_2}}$ is inserted in eq. (15).

$$
\dot{V} = 2\boldsymbol{e_2}^T P A_m \dot{\boldsymbol{e_2}} + 2\boldsymbol{e_2}^T P \frac{1}{k^*} B_m \bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\omega}} + 2\bar{\boldsymbol{\Phi}}^T \Gamma^{-1} \dot{\bar{\boldsymbol{\Phi}}} \tag{16}
$$

Next, it is to be noted that a square matrix A can be split up into a symmetric part A_s and an anti-symmetric part A_{as}:

$$
A = A_s + A_{as} = \frac{1}{2}\big(A + A^T\big) + \frac{1}{2}\big(A - A^T\big)
$$

Since PA_m is a square matrix, this relationship, together with the Algebraic Riccati equation, can be exploited:

$$
\begin{aligned}
2PA_m &= 2\big([PA_m]_s + [PA_m]_{as}\big) = \big(PA_m + [PA_m]^T\big) + \big([PA_m - [PA_m]^T\big) \\
&= \big(PA_m + A_m^T P\big), \text{ since } P = P^T, \text{ and } PA_m - A_m^T P = 0 \\
\Rightarrow PA_m &= -\frac{1}{2} Q
\end{aligned}
$$

Reinserting this into eq. (16) gives the dynamics of $\dot{V}$ which is required to be negative semi-definite:

$$
\dot{V} = -\boldsymbol{e_2}^T Q \boldsymbol{e_2} + 2\boldsymbol{e_2}^T P \frac{1}{k^*} B_m \bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\omega}} + 2\bar{\boldsymbol{\Phi}}^T \Gamma^{-1} \dot{\bar{\boldsymbol{\Phi}}} \leq 0 \tag{17}
$$

It can be noted that in eq. (17) $-\boldsymbol{e_2}^T Q \boldsymbol{e_2}$ is negative semi-definite since Q is positive definite and the state error $\boldsymbol{e_2}^T \boldsymbol{e_2}$ is greater or equal to zero. This trivially makes $-\boldsymbol{e_2}^T Q \boldsymbol{e_2}$ negative semi-definite too. In order for the second and third term of $\dot{V}$ to be negative semi-definite, the dynamics of the control gain $\dot{\bar{\boldsymbol{\theta}}} = \dot{\bar{\boldsymbol{\Phi}}} + \dot{\bar{\boldsymbol{\theta}}}^* = \dot{\bar{\boldsymbol{\Phi}}}$ is chosen such that the third term cancels out the second term:

$$
\dot{\bar{\boldsymbol{\theta}}} = -\text{sign}(\frac{1}{k^*}) \Gamma \bar{\boldsymbol{\omega}} \boldsymbol{e_2}^T \frac{1}{|(k^*)|} P B_m \tag{18}
$$

This expression can be further simplified. Firstly, $\text{sign}(1/k^*) = \text{sign}(k_p/k_m) = \text{sign}(\tau b_0)$, and, since $\tau > 0$ and $b_0 < 0$ is known from section 2.2.1, $\text{sign}(1/k^*)$ must be negative. Secondly, the non-singular (or positive-definite) matrix P from the Algebraic Riccati equation is known. Furthermore, assuming the matrices A, B, C, D

of a state space description are known and the system is fully controllable and observable, then the Kalman-Yakubovich-Popov Lemma [1] states:

$$\begin{aligned} PA + A^T P &= -LL^T \\ PB - C^T &= -LW \\ D + D^T &= W^T W \end{aligned}$$

Inserting the system matrices $A_m, 1/|k^*|B_m, C_m, D_m$ into the above yields:

$$C_m = P\frac{1}{|k^*|}B_m$$

Plugging this into eq. (18) gives:

$$\dot{\bar{\boldsymbol{\theta}}} = \Gamma \bar{\boldsymbol{\omega}} \boldsymbol{e_2}^T C_m \tag{19}$$

Here, it can be noted that $\boldsymbol{e_2}^T C_m = C_m^T \boldsymbol{e_2}$, since $C_m \,\epsilon\, \mathbb{R}^{4x1}$ and $\boldsymbol{e_2} \,\epsilon\, \mathbb{R}^{4x1}$. Recalling that $\boldsymbol{e_2} = \boldsymbol{\omega} - \boldsymbol{\omega_m}$ and $C_m = C_p$, $e_2^T C_m$ can also be written as:

$$\boldsymbol{e_2}^T C_m = C_m^T \boldsymbol{e_2} = C_m^T(\boldsymbol{\omega} - \boldsymbol{\omega_m}) = y - y_m = e_1$$

To sum up, the control gains should have the following dynamics:

$$\dot{\bar{\boldsymbol{\theta}}} = \Gamma \bar{\boldsymbol{\omega}} e_1 \tag{20}$$

If the dynamics are chosen as in eq. (20), the third term of eq. (17) will cancel out the second term and make $\dot{V}$ negative semi-definite:

$$\dot{V} = -\boldsymbol{e_2}^T Q \boldsymbol{e_2} \leq 0\,, \text{ negative semi-definite}$$

Since $\dot{V}$ is negative semi-definite, it can also be implied that V must be decrescent as V will not grow over time. Mathematically, for a scalar function $\beta(\|\boldsymbol{\omega}\|)$ with $\beta(0) = 0$, this means:

$$\beta(\|\boldsymbol{\omega}\|) \geq V(\boldsymbol{\omega}, t)$$

To conclude, $V = \boldsymbol{e_2}^T P \boldsymbol{e_2} + \bar{\boldsymbol{\Phi}}^T \Gamma^{-1} \bar{\boldsymbol{\Phi}}$ fulfills the four criteria and is a Lyapunov function. This in return means that the system is asymptotically stable which implies that $\boldsymbol{e_2} = \boldsymbol{\omega} - \boldsymbol{\omega_m}$ converges to zero. For this to be possible $\|\boldsymbol{\omega}\|$ must be bounded since $\boldsymbol{\omega}$ must converge to $\boldsymbol{\omega_m}$. If a bounded input signal $r(t)$ is chosen, then $\|\bar{\boldsymbol{\omega}}\|$ will hence be bounded too. Therefore, it can be said that $\left\|\bar{\boldsymbol{\theta}}(t, \bar{\boldsymbol{\omega}})\right\|$ must be bounded too. From this, it follows that $\left\|\bar{\boldsymbol{\Phi}}(\bar{\boldsymbol{\theta}})\right\|$ is bounded because the discovery of a Lyapunov function makes the system defined in eq. (13) asymptotically stable. Since it was shown that the output error $e_1 = C_m^T \boldsymbol{e_2}$ depends on the state error $\boldsymbol{e_2}$, it can be said that the output error e_1 converges to zero because $\boldsymbol{e_2}$ converges to zero. This implies that the system does not have a steady-state error. Furthermore, this requires $\|y\|$ to be bounded. Since it can be said that $\left\|\bar{\boldsymbol{\theta}}\right\|$ and $\|\bar{\boldsymbol{\omega}}\|$ are bounded, $u(t) = \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\omega}}$ will be bounded too. In summary, the adaptive controller makes the system respond as desired from the requirements of section 2.1. The system has the following properties.

- Zero steady-state error, e_1 converges to zero, $y(t)$ converges to $y_m(t)$
- Bounded input $u(t)$ and bounded output $y(t)$
- Zero steady-state state error, $\boldsymbol{e_2}$ converges to zero, $\boldsymbol{\omega}(t)$ converges to $\boldsymbol{\omega_m}(t)$
- Bounded state signals $\|\boldsymbol{\omega}\|$, $\|\bar{\boldsymbol{\omega}}\|$
- Bounded controller gains $\left\|\bar{\boldsymbol{\theta}}\right\|$
- Bounded control gain error signal $\left\|\bar{\boldsymbol{\Phi}}\right\|$

3 Simulation Results

To evaluate the performance of the adaptive control law, different parameters t_1, t_2 for the observer polynomial T and different gain matrices Γ were used. Additionally, the plant Σ_{guess} used to guess the initial controller gains was a second order system with relative degree of one, natural frequency $\omega_n = 2\,rad/s$ and light damping $\zeta = 0.1$. Mathematically expressed, Σ_{guess} was defined as:

$$\Sigma_{guess} = \frac{-(s+1)}{s^2 + 2\zeta\omega_n s + \omega^2}\,, \quad \omega_n = 2\,\frac{rad}{s}, \zeta = 0.1 \tag{21}$$

Furthermore, the first order reference model used for the simulations had a time constant $\tau = 1s$. The resulting transfer function $H_m(s)$ used for the simulations was:

$$H_m(s) = \frac{1}{s+1}$$

The simulations were run with Matlab and Simulink. Figure 1 gives a rough overview of the control structure. A fixed time step of $T = 1e-04s$ was used to run the simulations. Such a small time step ensured that the simulation captured all of the relevant dynamics from one time step to another. Additionally, the initial observer polynomial T was $T = s^2 + 20s + 100$ while $\Gamma = \mathbb{I}$ was the start matrix for Γ.

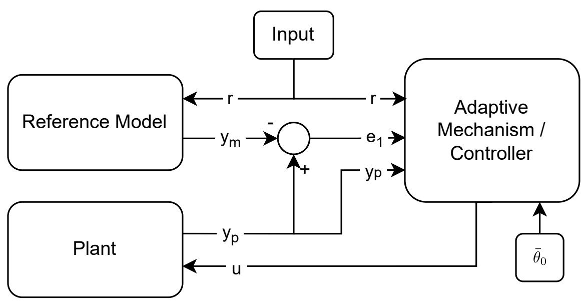


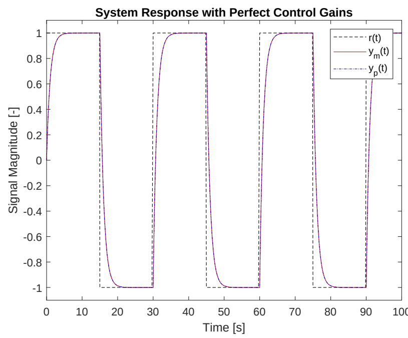
Figure 1: Control structure implemented in Simulink and Matlab.

3.1 Perfect Control Input

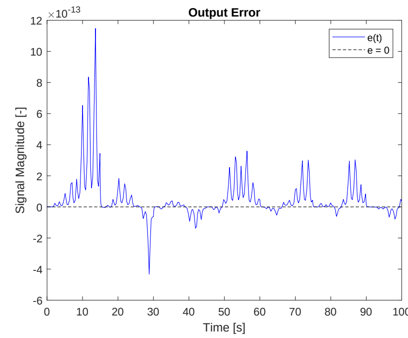
As discussed in section 2.2.1, there exists a perfect control input $u^* = \bar{\boldsymbol{\theta}}^{*T}\bar{\boldsymbol{\omega}}$. However, the perfect control gains $\bar{\boldsymbol{\theta}}^*$ can only be calculated if the system parameters b_0, b_1, a_1, a_2 are known. Since for simulation purposes the plant must be defined with certain parameters, the perfect control law can be verified by calculating the adaptive control gains with these parameters. The plots in fig. 2 show that with a perfect initial guess of the control gains $\bar{\boldsymbol{\theta}}_0$, the system response matches the reference model response. The output error is in the order of $1.0e-13$ which is due to numerical simulation errors. The plots in fig. 3 show that if the initial controller gains guess is perfect, the controller gains will not have to adapt. The slight adaptations of e.g. k^* can again be explained by numerical simulation discrepancies.

3.2 Adaptive Control Input

For the adaptive control input, the exact parameters of the true plant are unknown. The initial guess comes from the second order system defined in eq. (21). Since these gains will not match the perfect controller gains, the system response will no longer match the reference model response. Thanks to the adaptive control law,

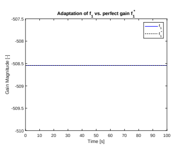


(a) Plant and reference model response to square wave reference input signal with perfect initial guess.

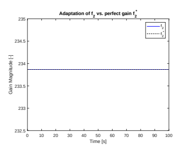


(b) Plant output error with square wave reference input signal and perfect initial guess.

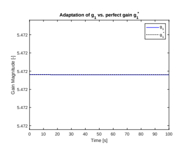
Figure 2: The plant matches the reference model if the initial controller gains are perfect.



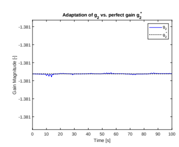
(a) Adaptation of f_1



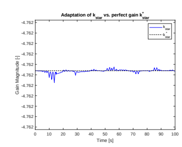
(b) Adaptation of f_2



(c) Adaptation of g_1



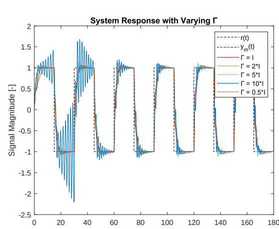
(d) Adaptation of g_2



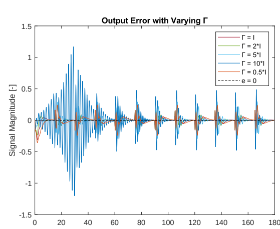
(e) Adaptation of k^*

Figure 3: The controller gains adapt negligibly if the initial guess $\bar{\boldsymbol{\theta}}_0 = \bar{\boldsymbol{\theta}}^*$ is perfect.

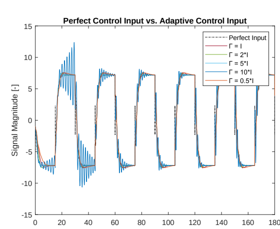
the controller gains are adapted as described in section 2.2.2, and the steady-state error converges to zero while the signals are guaranteed to be bounded. However, the design choices of T and Γ strongly influence the system performance. Ill-designed T and Γ can lead to a slow or strongly oscillatory system response and error convergence. The influence of varying Γ can be seen in fig. 4. It can be



(a) Plant output response



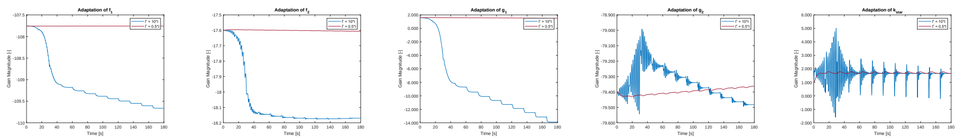
(b) Plant output error



(c) Plant input

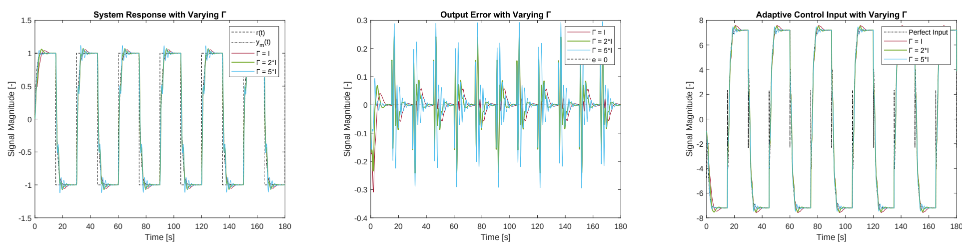
Figure 4: The plant response, plant output error and plant input with a square reference signal vary strongly depending on which gain matrix Γ is chosen.

observed that the controller provides bounded signals and ensures that the steady-state error converges to zero with any Γ as was proven in section 2.2.2. However, the system response will be quicker and more oscillatory if an aggressive Γ is chosen (which means Γ is large in magnitude) and slower and less oscillatory if a small Γ is chosen. For instance, for $\Gamma = 10 \cdot \mathbb{I}$, the response at first has an undesired, unstable behavior. This is because the gains get adapted too abruptly. On the other hand, a Γ of small magnitude performs poorly too, as the gains adapt too slowly which makes the error converge slow. Figure 5 clearly shows that the system with a large Γ adapts the gains very quickly while the system with a small Γ adapts the gains very slowly.



(a) Adaptation of f_1 (b) Adaptation of f_2 (c) Adaptation of g_1 (d) Adaptation of g_2 (e) Adaptation of k^*

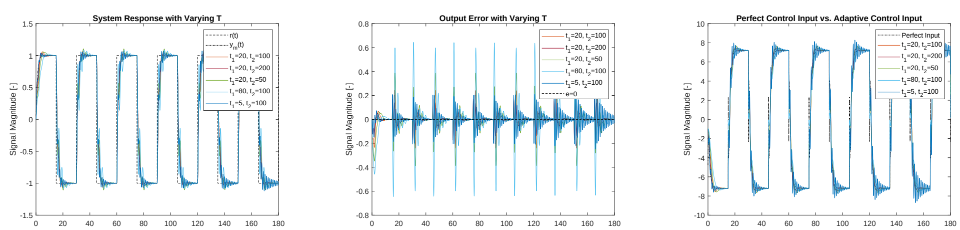
Figure 5: Difference in gain adaptation between an aggressive $\Gamma = 10 \cdot \mathbb{I}$ and small $\Gamma = 0.5 \cdot \mathbb{I}$.



(a) Plant output response (b) Plant output error (c) Plant input

Figure 6: The plant response, plant output error and plant input with a square reference signal vary strongly depending on which gain matrix Γ is chosen.

Figure 6 shows that $\Gamma = 2 \cdot \mathbb{I}$ has a quick response without too big of an overshoot and without very oscillatory behavior. Keeping $\Gamma = 2 \cdot \mathbb{I}$, the influence of a varying observer polynomial T was analyzed.



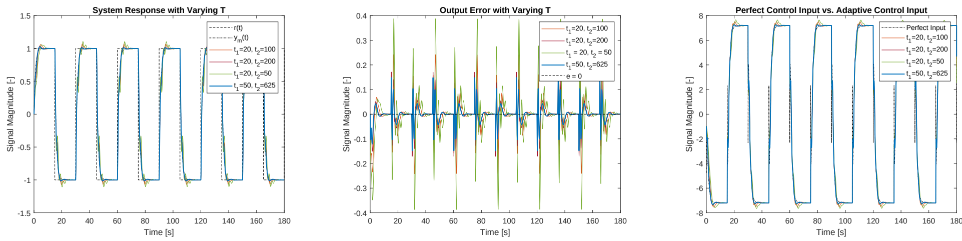
(a) Plant output response (b) Plant output error (c) Plant input

Figure 7: The plant response, plant output error and plant input with a square reference signal vary strongly depending on which observer polynomial T is chosen.

As seen in fig. 7, if the parameter t_1 of the observer polynomial is chosen too small, the system responds poorly. This makes sense, as t_1 mostly determines where the roots of the observer polynomial T lie. In general, a system converges to zero quicker the more negative the real part of its roots are. The roots of T are related to the parameters t_1 and t_2 as follows:

$$s^0_{1,2} = \frac{-t_1 \pm \sqrt{t_1^2 - 4t_2}}{2}$$

As can be seen, the negative real part is mainly dependent on t_1. A large t_1 will bring a quicker system response. Additionally, it can be said that t_2 will influence how oscillatory the system response is. A system responds oscillatory if its roots have an imaginary part. Therefore, an oscillatory system response will appear if $t_2 > t_1^2/4$ as then the roots of T will have an imaginary part. Since T has two roots, it is desirable to design both poles of T to have a large negative real part and no imaginary part. The best scenario will arise when $t_1^2 = 4t_2$. Then there will be a double root at $s_{1,2}^0 = -t_1$ with a large, negative real part. Additionally, the roots will not have an imaginary part and thus won't excite the system in an oscillatory manner.



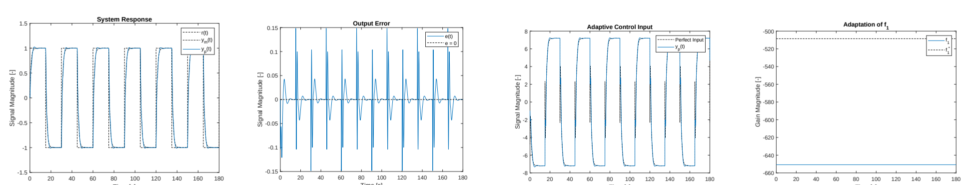
(a) Plant output response (b) Plant output error (c) Plant input

Figure 8: The plant response, plant output error and plant input with a square reference signal vary strongly depending on which observer polynomial T is chosen.

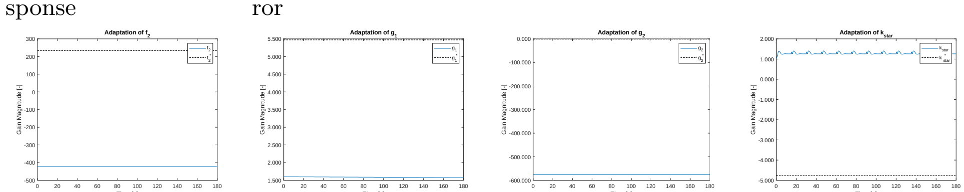
Figure 8 shows that the system response is best when t_1 has a large magnitude, and when $t_1^2 = 4t_2$ is fulfilled. To conclude, the optimal observer polynomial was set as $T = s^2 + 50s + 625$ while the optimal gain matrix was set as $\mathit{\Gamma} = 2 \cdot \mathbb{I}$.

3.2.1 Square Wave Reference Signal

The simulation results with the optimal observer polynomial $T = s^2 + 50s + 625$ and optimal gain matrix $\mathit{\Gamma} = 2 \cdot \mathbb{I}$ are plotted in fig. 9. It can be seen that the response is quick and not oscillatory. Furthermore, the input and output signals remain bounded as is desired.



(a) Plant output response (b) Plant output error (c) Plant input (d) Adaptation of f_1



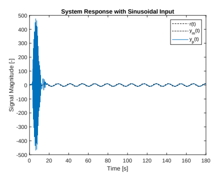
(e) Adaptation of f_2 (f) Adaptation of g_1 (g) Adaptation of g_2 (h) Adaptation of k_{star}

Figure 9: Plant output response, plant output error, plant input and controller gain adaptations with optimal observer polynomial $T = s^2 + 50s + 625$ and gain matrix $\mathit{\Gamma} = 2 \cdot \mathbb{I}$ for square wave reference signal.

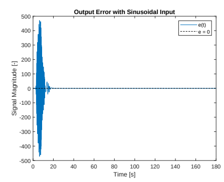
It is to be noted that the gains never converge to the perfect controller gains. This is because there are multiple sets of controller gains that can stabilize the system. Therefore, only the output error $e_1 = y - y_m$ converges to zero but not necessarily the controller gain error signal $\bar{\boldsymbol{\Phi}}$. In other words, eq. (13) converges to zero because the gain adaptation makes the product of $\bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\omega}}$ converge to zero.

3.2.2 Sinusoidal Reference Signal

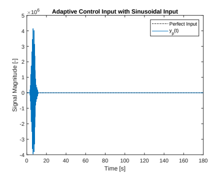
The plant was also simulated with the optimal observer polynomial and gain matrix from section 3.2.1 and a sinusoidal instead of square wave reference input signal $r(t) = 10sin(0.5t)$. Figure 10 shows that the system response fulfills the requirements. Albeit staying bounded, the control input signal at the beginning of the adaptation process is very large. A less aggressive observer polynomial and gain



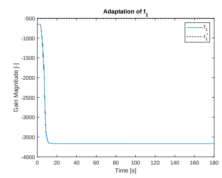
(a) Plant output response



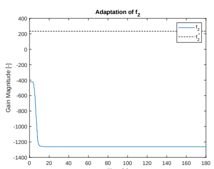
(b) Plant output error



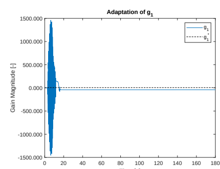
(c) Plant input



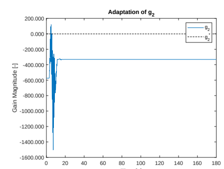
(d) Adaptation of f_1



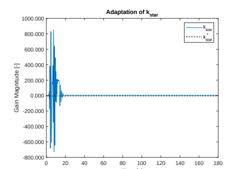
(e) Adaptation of f_2



(f) Adaptation of g_1



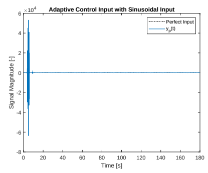
(g) Adaptation of g_2



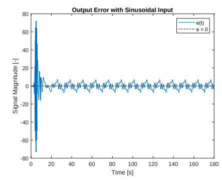
(h) Adaptation of k_{star}

Figure 10: Plant output response, plant output error, plant input and controller gain adaptations with optimal observer polynomial $T = s^2 + 50s + 625$ and gain matrix $\Gamma = 2 \cdot \mathbb{I}$ for sinusoidal reference signal.

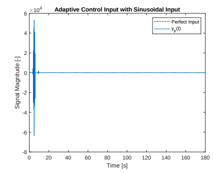
matrix are chosen to see the effects on the plant simulation with a sinusoidal reference signal. Figure 11 shows that the control input is considerably lower with the design parameters set less aggressively. It is to be noted that the output error oscillates around zero due to the time-varying sinusoidal reference signal. The controller continuously adapts the gains but always lags a bit behind the changing reference signal as it cannot anticipate the change in the reference signal. On average, the error is zero.



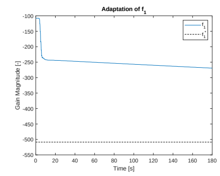
(a) Plant output response



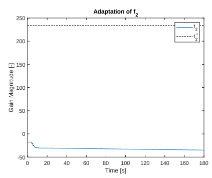
(b) Plant output error



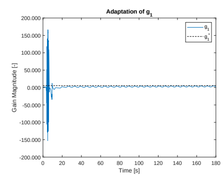
(c) Plant input



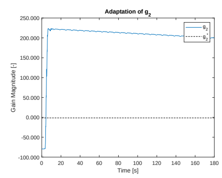
(d) Adaptation of f_1



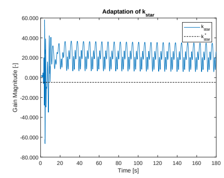
(e) Adaptation of f_2



(f) Adaptation of g_1



(g) Adaptation of g_2



(h) Adaptation of k_{star}

Figure 11: Plant output response, plant output error, plant input and controller gain adaptations with less aggressive observer polynomial $T = s^2 + 20s + 100$ and gain matrix $\Gamma = 1 \cdot \mathbb{I}$ for sinusoidal reference signal.

4 Conclusion

The adaptive control law proposed in section 2.2.2 successfully makes the steady-state error of the second order plant $\Sigma(s)$ with unknown parameters converge to zero. Furthermore, the input and output signals remain bounded. The simulation results in section 3 show that the system performance strongly depends on the design choices of the observer polynomial T and gain matrix Γ. The system response can be quicker or slower and more or less oscillatory depending on the design choices. Having roots with a large negative real part and no imaginary part for the observer polynomial T results in a quick response without much oscillation or overshoot. The same can be said for a moderately chosen Γ. In this exercise, there are no constraints on the input signals. In a real, hardware application this wouldn't further apply as all physical systems have their limits and saturations. In such a case, the design choices would need to be re-evaluated and adjusted to ensure that the control input doesn't exceed the physical limits of the system at hand.

References

[1] Brogliato, B., Maschke, B., Lozano, R., and Egeland, O. (2007). *Kalman-Yakubovich-Popov Lemma*, pages 69–176. Springer London, London.

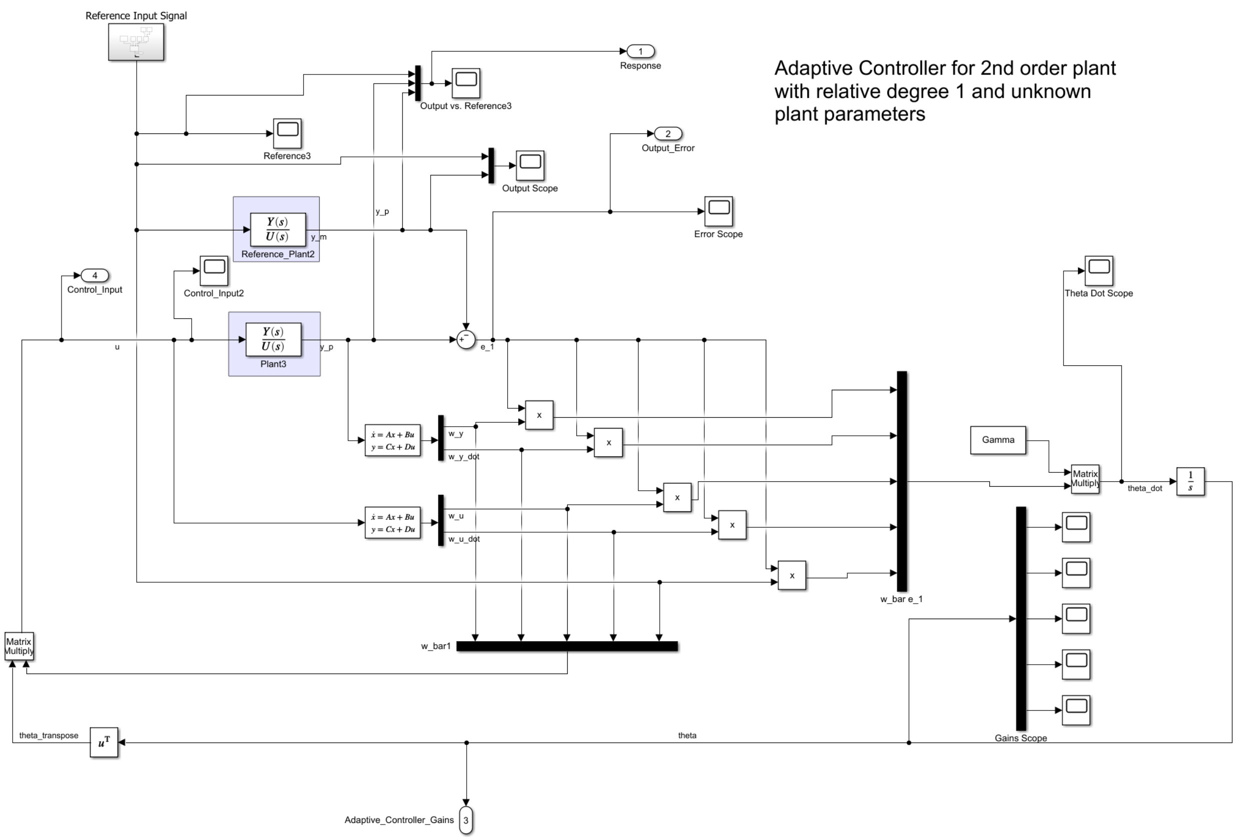
[2] Gluskin, E. (2003). Let us teach this generalization of the final-value theorem. *European Journal of Physics*, 24:591–597.

[3] Khalil, H. (2002). *Nonlinear Systems*. Pearson Education. Prentice Hall.

A Appendix

The simulations were conducted with Simulink and Matlab. The adaptive controller was designed with Simulink and enables the plant output to converge to the reference model output. The parameters were set and calculated with Matlab scripts. The *controlGains* function calculates the ideal controller gains for a given reference model, plant and observer polynomial. The *CA1* script defines the simulation parameters such as the reference model and simulation plant parameters. It calls the *CA1 Plot Script* which produces all of the plots that are included in this report.

A.1 Simulink Model



A.2 Simulation Script

CA1

Design an adaptive controller for a continuous-time system of second order, relative degree one and unknown parameters. Only input and output is measurable.

```
% Clean up
clear all
close all
clc
```

Plant parameters

Define plant and reference model parameters for simulation.

```
s = tf('s');
% Define plant in continuous time
R_p = s^2 + 0.29*s + 7.2; % Pole polynomial
k_p = -0.21; % Gain
Z_p = s-1/k_p; % Zero polynomial

disp('The plant used for simulation has the transfer function: ')
```

```
The plant used for simulation has the transfer function:
```

```
Plant = k_p*Z_p/R_p
```

```
Plant =

     -0.21 s - 1
  ------------------
  s^2 + 0.29 s + 7.2

Continuous-time transfer function.
```

```

% Reference model in continuous time
tau = 1;
R_m = s + 1/tau; % Pole polynomial
k_m = 1/tau; % Gain
Z_m = 1; % Zero polynomial

disp("The reference model used for simulation has the transfer function: ")
```

```
The reference model used for simulation has the transfer function:
```

```
Reference_Plant = k_m * Z_m/R_m
```

```
Reference_Plant =

    1
  -----
  s + 1

Continuous-time transfer function.
```

1

Simulation

Run Simulink model for simulation and plot results. 'Plots.mlx' runs the simulations with various gain matrices Γ and observer polynomials T. The initial guess for the controller gains are calculated in the function 'controlGains.mlx'.

```
run('Plots.mlx');
```

```
The guess of the plant used to calculate the initial gains has the transfer function:

Plant_guess =

      -s - 1
  ---------------
  s^2 + 0.4 s + 4

Continuous-time transfer function.

The perfect control gains from the perfect plant are:
f_1 = -508.5433
f_2 = 233.8667
g_1 = 5.4719
g_2 = -1.3810
k_star = -4.7619
```

CA1 Plot Script

This script plots the simulation results for the simulation with a perfect initial guess, and the simulation with varying Gamma as well as varying T observer polynomial. The reference signal is either a square wave or sinusoidal.

Simulation parameters

```
close all
% Reference input signal
% Choose ref: 1 = step input, 2 = sinusoidal input, 3 = square wave input
ref_signal = 3;

% Choose T
t_1 = 20.0;
t_2 = 100.0;
T = s^2 + t_1 * s + t_2;

% Choose Gamma
Gamma = 1*eye(5);

% Initial guess of controller gains from guessed plant
% Guess of plant:
% Second order system with natural frequency wn = 2 rad/s
% Very lightly damped: E.g. zeta = 0.1
% Relative degree n_star = 1, since original plant has relative degree of 1
% => One zero, two poles.
wn = 2; zeta = 0.1;
R_p_guess = s^2 + 2*zeta*wn*s + wn^2; % Pole polynomial
k_p_guess = -1; % Gain
Z_p_guess = s+1; % Zero polynomial

disp('The guess of the plant used to calculate the initial gains has the transfer function: ')
Plant_guess = k_p_guess*Z_p_guess/R_p_guess
```

Run & Plot Simulations

Simulation with perfect initial guess

```
% Colors for plots
blue = [0 0.4470 0.7410]; red = [0.6350 0.0780 0.1840]; orange = [0.8500 0.3250 0.0980];
green = [0.4660 0.6740 0.1880]; lblue = [0.3010 0.7450 0.9330]; purple = [0.4940 0.1840 0.5560];
% Run simulation with perfect initial guess
% Calculate perfect controller gains from true plant
disp('The perfect control gains from the perfect plant are: ')
[f_1, f_2, g_1, g_2, k_star] = controlGains(T, R_m, R_p, Z_p, k_m, k_p)
[f_1_0, f_2_0, g_1_0, g_2_0, k_star_0] = controlGains(T, R_m, R_p, Z_p, k_m, k_p);
simData = sim('CA1_alone.slx');
r = simData.yout{1}.Values.Data(:,1);
resp = simData.yout{1}.Values.Data(:,2);
ref = simData.yout{1}.Values.Data(:,3);
error = simData.yout{2}.Values.Data(:,1);
gains = simData.yout{3}.Values.Data(:,:);
control_input = simData.yout{4}.Values.Data(:,:);
t = simData.tout; i = 1;
ax = axes;
figure(i); i = i+1;
plot(t,r,'k--'); hold on; plot(t,ref,'-','Color',red); plot(t,resp,'-.','Color',blue); hold off;
title('System Response with Perfect Control Gains')
xlabel('Time [s]')
ylabel('Signal Magnitude [-]')
legend('r(t)','y_m(t)','y_p(t)')
ylim([-1.1 1.1]);
xlim([0, 180]);
saveas(gcf,'Perfect_Output.pdf')
figure(i); i = i+1;
plot(t,error,'Color',blue); hold on; plot(t,zeros(size(t)),'k--'); hold off;
title('Output Error')
xlabel('Time [s]')
ylabel('Signal Magnitude [-]')
xlim([0, 180]);
legend('e(t)','e = 0')
saveas(gcf,'Perfect_Error.pdf')
```

```
figure(i); i = i+1;
plot(t,-gains(:,2),'Color',blue); hold on; plot(t,f_1*ones(size(t)),'k--'); hold off;
title('Adaptation of f_1 vs. perfect gain f_1^{ *}')
xlabel('Time [s]')
ylabel('Gain Magnitude [-]')
xlim([0, 180]);
% ytickformat('%.3f')
legend('f_1','f_1^{ *}')
saveas(gcf,'Perfect_f_1.pdf')
figure(i); i = i+1;
plot(t,-gains(:,1),'Color',blue); hold on; plot(t,f_2*ones(size(t)),'k--'); hold off;
title('Adaptation of f_2 vs. perfect gain f_2^{ *}')
xlabel('Time [s]')
ylabel('Gain Magnitude [-]')
xlim([0, 180]);
% ytickformat('%.3f')
legend('f_2','f_2^{ *}')
saveas(gcf,'Perfect_f_2.pdf')
figure(i); i = i+1;
plot(t,-gains(:,4),'Color',blue); hold on; plot(t,g_1*ones(size(t)),'k--'); hold off;
title('Adaptation of g_1 vs. perfect gain g_1^{ *}')
xlabel('Time [s]')
ylabel('Gain Magnitude [-]')
xlim([0, 180]);
ytickformat('%.3f')
legend('g_1','g_1^{ *}')
saveas(gcf,'Perfect_g_1.pdf')
figure(i); i = i+1;
plot(t,-gains(:,3),'Color',blue); hold on; plot(t,g_2*ones(size(t)),'k--'); hold off;
title('Adaptation of g_2 vs. perfect gain g_2^{ *}')
xlabel('Time [s]')
ylabel('Gain Magnitude [-]')
xlim([0, 180]);
legend('g_2','g_2^{ *}')
ytickformat('%.3f')
saveas(gcf,'Perfect_g_2.pdf')
```

```
figure(i); i = i+1;
plot(t,gains(:,5),'Color',blue); hold on; plot(t,k_star*ones(size(t)),'k--'); hold off;
title('Adaptation of k_{star} vs. perfect gain k_{star}^*')
xlabel('Time [s]')
ylabel('Gain Magnitude [-]')
xlim([0, 180]);
legend('k_{star}','k_{star}^{ *}')
ytickformat('%.3f')
saveas(gcf,'Perfect_k_star.pdf')
figure(i); i = i+1;
plot(t,control_input,'Color',blue)
title('Perfect Control Input vs. Adaptive Control Input')
xlabel('Time [s]')
ylabel('Signal Magnitude [-]')
xlim([0, 180]);
saveas(gcf,'Perfect_Input.pdf')
```

Simulation with adaptive control gains

Compare the simulation results with varying Gamma

```
% Calculate the initial gains from the guessed plant
[f_1_0, f_2_0, g_1_0, g_2_0, k_star_0] = controlGains(T, R_m, R_p_guess, Z_p_guess, k_m, k_p_guess)

% Save simulation results with varying Gamma
Gamma = eye(5);
simData = sim('CA1_alone.slx');
r1 = simData.yout{1}.Values.Data(:,1);
resp1 = simData.yout{1}.Values.Data(:,2);
ref1 = simData.yout{1}.Values.Data(:,3);
error1 = simData.yout{2}.Values.Data(:,1);
gains1 = simData.yout{3}.Values.Data(:,:);
control_input_adaptive1 = simData.yout{4}.Values.Data(:,:);
t1 = simData.tout;
Gamma = 2*eye(5); % Change Gamma
simData = sim('CA1_alone.slx');
r2 = simData.yout{1}.Values.Data(:,1);
resp2 = simData.yout{1}.Values.Data(:,2);
```

```
ref2 = simData.yout{1}.Values.Data(:,3);
error2 = simData.yout{2}.Values.Data(:,1);
gains2 = simData.yout{3}.Values.Data(:,:);
control_input_adaptive2 = simData.yout{4}.Values.Data(:,:);
t2 = simData.tout;
Gamma = 5*eye(5); % Change Gamma
simData = sim('CA1_alone.slx');
r_5 = simData.yout{1}.Values.Data(:,1);
resp_5 = simData.yout{1}.Values.Data(:,2);
ref_5 = simData.yout{1}.Values.Data(:,3);
error_5 = simData.yout{2}.Values.Data(:,1);
gains_5 = simData.yout{3}.Values.Data(:,:);
control_input_adaptive_5 = simData.yout{4}.Values.Data(:,:);
t_5 = simData.tout;
Gamma = 10*eye(5); % Change Gamma
simData = sim('CA1_alone.slx');
r_10 = simData.yout{1}.Values.Data(:,1);
resp_10 = simData.yout{1}.Values.Data(:,2);
ref_10 = simData.yout{1}.Values.Data(:,3);
error_10 = simData.yout{2}.Values.Data(:,1);
gains_10 = simData.yout{3}.Values.Data(:,:);
control_input_adaptive_10 = simData.yout{4}.Values.Data(:,:);
t_10 = simData.tout;
Gamma = 0.5*eye(5); % Change Gamma
simData = sim('CA1_alone.slx');
r_05 = simData.yout{1}.Values.Data(:,1);
resp_05 = simData.yout{1}.Values.Data(:,2);
ref_05 = simData.yout{1}.Values.Data(:,3);
error_05 = simData.yout{2}.Values.Data(:,1);
gains_05 = simData.yout{3}.Values.Data(:,:);
control_input_adaptive_05 = simData.yout{4}.Values.Data(:,:);
t_05 = simData.tout;

% Plot the simulation results of varying Gamma
% Plot system responses
figure(i); i = i+1;
```

Function controlGains() to compute perfect/initial control gains

The controlGains() function computes the control gains for the adaptive control law.

Input: T, R_m, R_p, Z_p, k_m, k_p: Continuous-time transfer functions with polynomial orders as defined in report.

Output: f_1, f_2, g_1, g_2, k_star: Optimal/Initial gains for adaptive controller

```
function [f_1, f_2, g_1, g_2, k_star] = controlGains(T,R_m,R_p, Z_p, k_m, k_p)
    syms s_
    % Get polynomial coefficients from input continuous-time polynomials
    T_cell = tfdata(T); R_m_cell = tfdata(R_m);
    R_p_cell = tfdata(R_p); Z_p_cell = tfdata(Z_p);
    % Save coefficients from cell array in vector
    T_coeffs = T_cell{1}; R_m_coeffs = R_m_cell{1};
    R_p_coeffs = R_p_cell{1}; Z_p_coeffs = Z_p_cell{1};
    % Define symbolic polynomials to do calculations
    T_ = T_coeffs(1,1)*s_^2 + T_coeffs(1,2)*s_ + T_coeffs(1,3);
    R_m_ = R_m_coeffs(1,1)*s_ + R_m_coeffs(1,2);
    R_p_ = R_p_coeffs(1,1)*s_^2 + R_p_coeffs(1,2)*s_ + R_p_coeffs(1,3);
    Z_p_ = Z_p_coeffs(1,1)*s_ + Z_p_coeffs(1,2);

    % Find F, E via symbolic polynomial division.
    [F,E]=polynomialReduce(T_*R_m_,R_p_);

    % Find controller gains f_1, f_2 from coefficients of F_bar
    f = sym2poly(F/k_p);
    f_1 = f(1); f_2 = f(2);
    % Find controller gains g_1, g_2 from coefficients of G1
    G1 = vpa(simplify(E*Z_p_-T_));
    g = sym2poly(G1);
    g_1 = g(1,1); g_2 = g(1,2);
    % Calculate k_star
    k_star = k_m/k_p;
end
```

1