NATIONAL CHENG KUNG UNIVERSITY

ADAPTIVE CONTROL FINAL PROJECT

REAL-TIME PARAMETER ESTIMATION AND PREDICTION-BASED CONTROL FOR AIRCRAFT PITCH DYNAMICS

INSTRUCTOR: Dr. Ming-Shaung Ju

STUDENT: Yang-Rui Li

STUDENT ID: P48104011 (DAA)

July 3, 2021

Adaptive Control Final Project
Real-Time Parameter Estimation and Prediction-Based Control for Aircraft Pitch Dynamics

Yang-Rui Li*
Intelligent Embedded Control Laboratory
Department of Aeronautics and Astronautics
National Chang Kung University
Taiwan, Taiwan

July 3, 2021

Abstract

In this work, real-time parameter estimation and prediction-based control for the aircraft pitch dynamics is proposed. The presented recursive least-squares (RLS) estimator based on integral operator enables real-time estimating the parameters of the system, and it also provides a state prediction model. Introducing the integral operator has the benefit that the RLS estimator is low-sensitivity to noisy signals. Based on predicted state info, the controller performance can be further improved. The simulation result reveals that the effectiveness of the proposed control scheme.

Keywords: Real-time parameter estimation, RLS estimator, integral operator, prediction model, prediction-based control.

1 Introduction

In this work, real-time parameter estimation and prediction-based control for the aircraft pitch dynamics is proposed. For most flight vehicles, the ope-loop plant is unstable in general. It means that the parameters and stabilizing control feedback should be conducted at the same time. The main difficulty of the closed-loop excitation is that if the modes of the system cannot be well excited, there will be a relatively large error between the estimated parameters and the actual parameters, which will

*Email: p48104011@gs.ncku.edu.tw

affect the performance of the controller. In addition, the measured noise will also affect the performance of parameter estimation/control feedback. For these reasons, the RLS estimator based on the integral operator is proposed. Apart from the real-time estimating the system parameters, and also inhibiting the influence of measured noised on the parameter estimates. Furthermore, a state prediction model based on the RLS estimator is proposed, which can estimate more accurate and smooth state information, one enables the performance of the controller to be further improved. Compare the proposed prediction-based controller scheme with the traditional proportional-integral (PI) controller, the simulation result reveals the outstanding performance of the proposed controller for trajectory tracking/real-time parameter estimation.

The organization of this article is summarized as follows. In Section 2, the dynamic equation equations are introduced. Based on the system structure, the RLS estimator based on integral operator and the corresponding state prediction model are proposed in Section 3. The design of the prediction-based controller is presented in Section 4. For comparison, the traditional output feedback PI controller is applied in Section 5. The comparitive simulation is performed in Section 6. Finally, the conclusion and future work are made in Section 7.

2 Dynamics Equation of Aircraft Pitch Dynamics

The state-space model for aircraft pitching motion is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \tag{1}$$

where $\mathbf{x}(t) = [N_z, \dot{\theta}, \delta_e]^T$ is the state vector; u is the input signal of the aileron servo; and the system matrices are

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & -a \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} b_1 \\ 0 \\ a \end{bmatrix} \tag{2}$$

This model is called short-period dynamics. The parameters of the model given depend on the operating conditions, which can be described in terms of Mach number and altitude. Consider the aircraft is operated on the Mach number 0.5 and the altitude 5000 feet. The nominal parameters, true parameters, and true eigenvalues are provided in Table 1.

Table 1: True and nominal parameters of the system.

Parameter	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a	b
Nominal Value	-0.4948	8.705	48.075	0.1324	-0.4256	-5.695	7	-48.89
True Value	-0.9896	17.41	96.15	0.2648	-0.8512	-11.39	14	-97.78
True e-value	$\lambda_1 = -3.07$,		$\lambda_2 = 1.23$,		$\lambda_3 = -14$.			

It can be found that the system is unstable due to the unstable mode $\lambda = 1.23$. Therefore, the parameter estimation should be performed in a closed-loop to ensure that the system is excited in a stable region. In this project, we consider the case that all the true parameters of the system are unknown and only the nominal values are available. In the following section, the RLS estimator based on the integral operator is proposed to estimate the unknown parameters and to predict the system states.

3 Parameter Estimation via Recursive Least-Squares Algorithm

Consider the regressive equation of the form

$$y(k) = \boldsymbol{\varphi}^T(k)\boldsymbol{\theta} \tag{3}$$

where $y(k) \in \mathbb{R}^1$ is the equation output; $\boldsymbol{\varphi}(k) \in \mathbb{R}^n$ is the regressor vector; and $\boldsymbol{\theta} \in \mathbb{R}^n$ is the unknown parameters to be estimated. The recursive least-squares (RLS) estimator for (3) is given by [1]

$$\begin{aligned} \hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\left(y(k) - \boldsymbol{\varphi}^T(k)\hat{\boldsymbol{\theta}}(k-1)\right) && (n \times 1) \\ \mathbf{K}(k) &= \mathbf{P}(k-1)\boldsymbol{\varphi}(k)\left(1 + \boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)\right)^{-1} && (n \times 1) \\ \mathbf{P}(k) &= \left(\mathbf{I}_n - \mathbf{K}(k)\boldsymbol{\varphi}^T(k)\right)\mathbf{P}(k-1) && (n \times n) \end{aligned} \tag{4}$$

In which, $\hat{\boldsymbol{\theta}}$ denotes the estimate of $\boldsymbol{\theta}$ and the initial condition is $\hat{\boldsymbol{\theta}}(0) = \hat{\boldsymbol{\theta}}_0$, $\mathbf{P}(0) = \mathbf{P}_0$.

To construct the RLS estimator, the system dynamics (1) should be rewritten as the regressive form, which is

$$\begin{aligned} \dot{x}_1(t) &= \boldsymbol{\varphi}_1^T(t)\boldsymbol{\theta}_1 \\ \dot{x}_2(t) &= \boldsymbol{\varphi}_2^T(t)\boldsymbol{\theta}_2 \\ \dot{x}_3(t) &= \varphi_3^T(t)\theta_3 \end{aligned} \tag{5}$$

where

$$\begin{aligned} \boldsymbol{\varphi}_1^T(t) &= \begin{bmatrix} u(t) & x_1(t) & x_2(t) & x_3(t) \end{bmatrix} & \boldsymbol{\theta}_1 &= \begin{bmatrix} b & a_{11} & a_{12} & a_{13} \end{bmatrix}^T \\ \boldsymbol{\varphi}_2^T(t) &= \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix} & \boldsymbol{\theta}_2 &= \begin{bmatrix} a_{21} & a_{22} & a_{23} \end{bmatrix}^T \\ \varphi_3^T(t) &= \begin{bmatrix} -x_3(t) + u(t) \end{bmatrix} & \theta_3 &= \begin{bmatrix} a \end{bmatrix} \end{aligned} \tag{6}$$

Based on the measured info $[x_1(t), x_2(t), x_3(t)]$ and $[\dot{x}_1(t), \dot{x}_2(t), \dot{x}_3(t)]$. The system parameters $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and θ_3 can be easily identified. However, the state derivatives $[\dot{x}_1(x), \dot{x}_2(t), \dot{x}_3(t)]$ cannot be acquired by measurement sensors in general. One of the methods is to estimate these quantities by numerical differentiation. However, the numerical differentiation will amplify the measured noise and result in the in-precise parameter estimation. As a reason, the numerical differentiation should be avoided. In what follows, an integral operator is introduced to conduct the parameter estimation.

3.1 RLS Estimator with Integral Operator

Applying the Laplace transform for (5) gives

$$\begin{aligned} sX_1(s) - x_1(0) &= \begin{bmatrix} U(s) & X_1(s) & X_2(s) & X_3(s) \end{bmatrix} \boldsymbol{\theta}_1 \\ sX_2(s) - x_2(0) &= \begin{bmatrix} X_1(s) & X_2(s) & X_3(s) \end{bmatrix} \boldsymbol{\theta}_2 \\ sX_3(s) - x_3(0) &= \begin{bmatrix} -X_3(s) + U(s) \end{bmatrix} \theta_3 \end{aligned} \tag{7}$$

where s is the Laplace operator and $U(s) = \mathscr{L}\{u(t)\}$, $X_i(s) = \mathscr{L}\{x_i(t)\}$, $i = 1,2,3$.

Introducing the integral operator $1/(s+\lambda)$ yields

$$\begin{aligned} X_1^{1f}(s) &= \begin{bmatrix} \phi_u^{1f}(s) & \phi_{x_1}^{1f}(s) & \phi_{x_2}^{1f}(s) & \phi_{x_3}^{1f}(s) \end{bmatrix} \boldsymbol{\theta}_1 \\ X_2^{1f}(s) &= \begin{bmatrix} \phi_{x_1}^{1f}(s) & \phi_{x_2}^{1f}(s) & \phi_{x_3}^{1f}(s) \end{bmatrix} \boldsymbol{\theta}_2 \\ X_3^{1f}(s) &= \begin{bmatrix} -\phi_{x_3}^{1f}(s) + \phi_u^{1f}(s) \end{bmatrix} \theta_3 \end{aligned} \tag{8}$$

where $\lambda \geq 0$ is the integral factor, and

$$\phi_u^{1f}(s) = \frac{U(s)}{s+\lambda}; \quad \phi_{x_i}^{1f}(s) = \frac{X_i(s)}{s+\lambda}; \quad X_i^{1f}(s) = \frac{sX_i(s) - x_i(0)}{s+\lambda}; \quad i = 1,2,3. \tag{9}$$

Reformulating $X_i^{1f}(s)$ to eliminating $sX_i(s)$ produces

$$\begin{aligned} X_i^{1f}(s) &= \frac{sX_i(s) - x_i(0)}{s+\lambda} \\ &= \frac{(s+\lambda)X_i(s) - \lambda X_i(s) - x_i(0)}{s+\lambda} \\ &= X_i(s) + Y_{i1}(s), \quad Y_{i1}(s) = \frac{-\lambda X_i(s) - x_i(0)}{s+\lambda}, \quad i = 1,2,3. \end{aligned} \tag{10}$$

Inverse Laplace transform for (8) results in

$$\begin{aligned} x_1(t) + y_{11}(t) &= \begin{bmatrix} \phi_u^{1f}(t) & \phi_{x_1}^{1f}(t) & \phi_{x_2}^{1f}(t) & \phi_{x_3}^{1f}(t) \end{bmatrix} \boldsymbol{\theta}_1 \triangleq \boldsymbol{\varphi}_{1f}^T(t)\boldsymbol{\theta}_1 \\ x_2(t) + y_{21}(t) &= \begin{bmatrix} \phi_{x_1}^{1f}(t) & \phi_{x_2}^{1f}(t) & \phi_{x_3}^{1f}(t) \end{bmatrix} \boldsymbol{\theta}_2 \triangleq \boldsymbol{\varphi}_{2f}^T(t)\boldsymbol{\theta}_2 \\ x_3(t) + y_{31}(t) &= \begin{bmatrix} -\phi_{x_3}^{1f}(t) + \phi_u^{1f}(t) \end{bmatrix} \theta_3 \triangleq \varphi_{3f}(t)\theta_3 \end{aligned} \tag{11}$$

where

$$\begin{aligned} \boldsymbol{\varphi}_{1f}^T(t) &= \begin{bmatrix} \phi_u^{1f}(t) & \phi_{x_1}^{1f}(t) & \phi_{x_2}^{1f}(t) & \phi_{x_3}^{1f}(t) \end{bmatrix} \\ \boldsymbol{\varphi}_{2f}^T(t) &= \begin{bmatrix} \phi_{x_1}^{1f}(t) & \phi_{x_2}^{1f}(t) & \phi_{x_3}^{1f}(t) \end{bmatrix} \\ \varphi_{3f}(t) &= \begin{bmatrix} -\phi_{x_3}^{1f}(t) + \phi_u^{1f}(t) \end{bmatrix} \end{aligned} \tag{12}$$

and

$$\begin{aligned} \dot{\phi}_u^{1f}(t) &= -\lambda\phi_u^{1f}(t) + u(t), & \phi_u^{1f}(0) &= 0 \\ \dot{\phi}_{x_i}^{1f}(t) &= -\lambda\phi_{x_i}^{1f}(t) + x_i(t), & \phi_{x_i}^{1f}(0) &= 0 \\ \dot{y}_{i1}(t) &= -\lambda y_{i1} - \lambda x_i(t), & y_{i1}(0) &= -x_i(0), \quad i = 1,2,3. \end{aligned} \tag{13}$$

Hence, it can be seen that the regressive formula (11) is in terms of measurable data $x_i(t)$, $i = 1,2,3$. The RLS estimators can then be used to estimate the system parameters $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and θ_3. Taking the first equation in (11) as an example, letting $y(k) = x_1(k) + y_{11}(k)$ and $\boldsymbol{\varphi}^T(k) = \boldsymbol{\varphi}_{1f}^T(k)$, the estimates $\hat{\boldsymbol{\theta}}_1 = [\hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{14}]^T$ can then be obtained by the RLS estimator (4). In the same manners, the estimates $\hat{\theta}_2$ and $\hat{\theta}_3$ can be also acquired.

3.2 Prediction Model Based on RLS Estimator

From the proposed regressive formula (11), the following predicted model based on RLS estimator is established:

$$\begin{aligned} x_{1p}(t) &= -y_{11}(t) + \boldsymbol{\varphi}_{1f}^T(t)\hat{\boldsymbol{\theta}}_1 \\ x_{2p}(t) &= -y_{21}(t) + \boldsymbol{\varphi}_{2f}^T(t)\hat{\boldsymbol{\theta}}_2 \\ x_{3p}(t) &= -y_{31}(t) + \varphi_{3f}(t)\hat{\theta}_3 \end{aligned} \tag{14}$$

where $x_{1p}(t)$, $x_{2p}(t)$, and $x_{3p}(t)$ represent the predicted states of $x_1(t)$, $x_2(t)$, and $x_3(t)$, respectively; and the estimates $\hat{\boldsymbol{\theta}}_1$, $\hat{\boldsymbol{\theta}}_2$, $\hat{\theta}_3$ are obtained from RLS estimators. The author believes that $x_{ip}(t)$ can be also interpreted as the "estimated" state because the input and measurable states info are used in (14).

In what follows, the state-feedback control with integral action based on estimated parameters and the predicted states, namely prediction-based control, is presented.

4 Prediction-Based Control

Consider the linear system described by (1). Let $y(t) = x_1(t)$ be the control objective to be tracked. Define the tracking error as $e(t) = \mathbf{C}\mathbf{x}(t) - r(t)$, where $\mathbf{C} = [1, 0, 0]$. The state-feedback control law with integral action is given by

$$u(t) = -\mathbf{F}\mathbf{x}(t) - F_f z(t) \tag{15}$$

where $\mathbf{F} \in \mathbb{R}^{1\times 3}$ is the state-feedback gain matrix; $F_f \in \mathbb{R}^1$ is the integral gain; and

$$\dot{z}(t) = e(t) = \mathbf{C}\mathbf{x}(t) - r(t) \tag{16}$$

The term $-\mathbf{F}\mathbf{x}(t)$ can be seen as used for stabilizing the state variables, as for $-F_f z(t)$ driving the specified state variable to the desired one.

Combining (1) and (16) produces the augmented system as

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{3\times 1} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -1 \end{bmatrix} r(t) \tag{17}$$

Substituting (15) into (17) follows that

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{z}(t) \end{bmatrix} = \left(\begin{bmatrix} \mathbf{A} & \mathbf{0}_{3\times 1} \\ \mathbf{C} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{F} & F_f \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -1 \end{bmatrix} r(t) \tag{18}$$

or, in short,

$$\dot{\mathbf{X}}(t) = \left(\mathbf{A}_{aug} - \mathbf{B}_{aug}\mathbf{F}_{aug}\right)\mathbf{X}(t) + \mathbf{\Gamma} r(t) \tag{19}$$

where

$$\mathbf{X}(t) = \begin{bmatrix}\mathbf{x}(t)\\ z(t)\end{bmatrix}; \quad \mathbf{A}_{aug} = \begin{bmatrix}\mathbf{A} & \mathbf{0}_{3\times1}\\ \mathbf{C} & 0\end{bmatrix}; \quad \mathbf{B}_{aug} = \begin{bmatrix}\mathbf{B}\\ 0\end{bmatrix}; \quad \mathbf{\Gamma} = \begin{bmatrix}\mathbf{0}_{3\times1}\\ -1\end{bmatrix}; \quad \mathbf{F}_{aug} = \begin{bmatrix}\mathbf{F} & F_f\end{bmatrix} \tag{20}$$

Thus, it has to design the augmented state-feedback gain matrix $\mathbf{F}_{aug}$ to guarantee the closed-loop system matrix $\mathbf{A}_{aug,c} = \mathbf{A}_{aug} - \mathbf{B}_{aug}\mathbf{F}_{aug}$ is Hurwitz. And then we have

$$x_1(t) \to r(t), \quad x_2(t) \to 0, \quad x_3(t) \to 0 \quad \text{as} \quad t \to \infty \tag{21}$$

Because the DC-gain of $r(t) \to x_1(t)$ is

$$\mathbf{G}_{aug}(s) = \mathbf{C}_{aug}\left(s\mathbf{I}_4 - \mathbf{A}_{aug,c}\right)^{-1}\mathbf{\Gamma} \quad \to \quad \mathbf{G}_{aug}(0) = \mathbf{C}_{aug}\left[-\mathbf{A}_{aug,c}\right]^{-1}\mathbf{\Gamma} = 1 \tag{22}$$

Notice that $\mathbf{C}_{aug} = [1, 0, 0, 0]$.

Based on the estimated parameters and states, the prediction-based state-feedback (PSB) controller is given by

$$u(t) = -\mathbf{F}\mathbf{x}_p(t) - F_f z_p(t) \tag{23}$$

where $\mathbf{x}_p = [x_{1p}, x_{2p}, x_{2p}]^T$; $\dot{z}_p = x_{1p}(t) - r(t)$; and $(\mathbf{F}, F_f)$ are designed based on estimated system matrices. The overall control scheme is provided by Figure 1.

In the following simulations, the integral factor $\lambda = 15$ is considered. Based on the parameter estimates, place the estimated closed-loop poles at -10, -12, -13, -14. The nominal values are used as the initial guess of the RLS estimator, and $\mathbf{P}(0) = 10^4\,\mathbf{I}_n$ is applied.

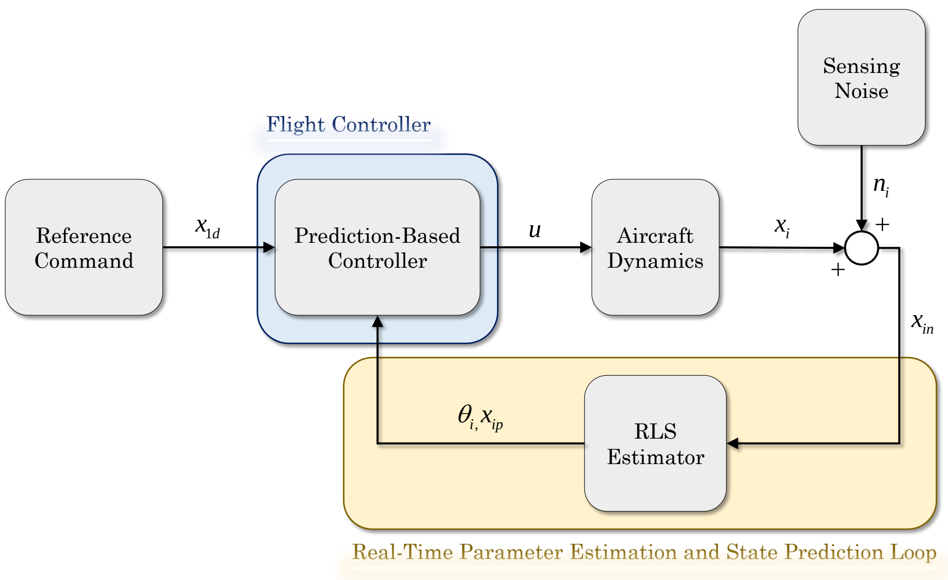


Figure 1: Control block diagram of the prediction-based control.

5 Proportional-Integral Controller

To compare the performance of proposed prediction-based state-feedback controller and the traditional controller, the PI controller is designed in this section. Consider the PI controller of the form:

$$G_c(s) = k_p + k_i \frac{1}{s} \tag{24}$$

where $k_p > 0$ and $k_i > 0$ to be designed. In the sense of controller design, the nominal values of the system will be used in designing the controller so that the desired nominal closed-loop performance can be achieved. Based on Matlab's PID tuner toolbox, the control gains (k_p, k_i) are designed as

$$k_p = -0.8956; \quad k_i = -0.0425 \tag{25}$$

The corresponding nominal and true closed-loop systems are

$$T_d(s) = \frac{43.787(s + 0.04746)(s^2 + 0.5423s + 7.148)}{(s + 51.21)(s + 0.04853)(s^2 + 0.446s + 5.977)} \tag{26}$$

and

$$T(s) = \frac{87.575(s + 0.04746)(s^2 + 1.085s + 28.59)}{(s + 102.5)(s + 0.04851)(s^2 + 0.9s + 23.91)} \tag{27}$$

respectively. It can be found that the true closed-loop system $T(s)$ is stable in this case. It should be noticed the $T(s)$ is not always stable by using the nominal values to design the controller.

The following simulations will compare the tracking performance for the proposed PSB controller and the PI controller.

6 Numerical Simulation

6.1 Command Smoothing Strategy

To avoid the aggressive commands result in the control oscillation/structure resonance, the command prefilter is designed to eliminate the high-frequency command inputs. Consider the following transfer function:

$$G_f(s) = \frac{1}{(\tau s + 1)^n} \tag{28}$$

where $\tau = 1/(2\pi F_c)$, F_c is cut-off frequency. The smooth command input is given by

$$r_f(s) = G_s(s) r(s) \tag{29}$$

where $r(s)$ and $r_f(s)$ are the original and smooth command input, respectively. Figure 2 shows the profile of these commands. In which, $n = 5$ and $F_c = 0.5$ (Hz) are used.

In the other words, smooth command $r_f(t)$ will be used as the desired command input $x_{1d}(t)$. The control objective is to drive $x_1(t) \to x_{1d}(t)$, the simulation results of the two controllers are shown as follows.

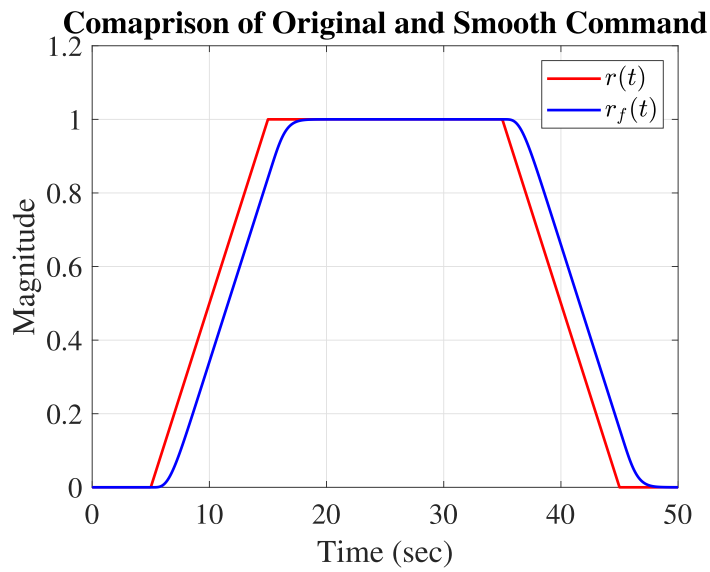


Figure 2: Comparison of original command $r(t)$ and smooth command $r_f(t)$.

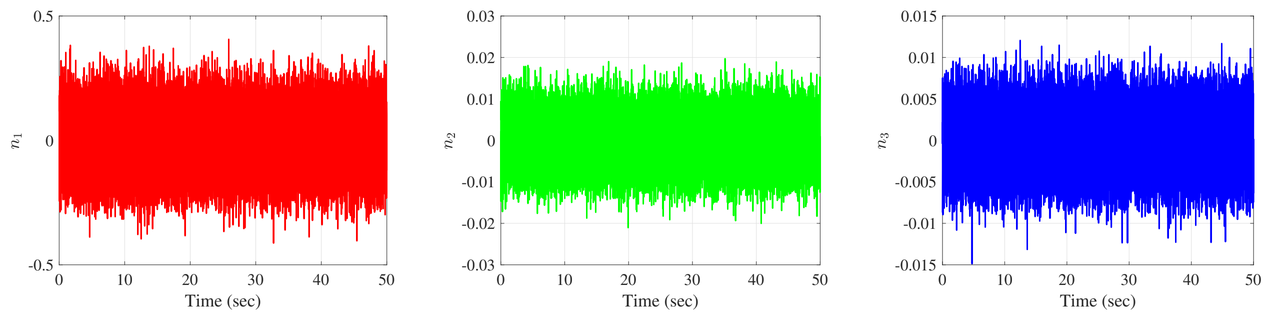
6.2 Simulation Results

In this project, we consider the measured noise, which is Gaussian white noise with normal distribution $\mathcal{N}(0,\sigma^2)$, where σ^2 is the variance of the noise. Figure 3 displays the evolution of measured noise.

The simulation results for the PI and the PSB controllers are shown in Figure 4–Figure 7.

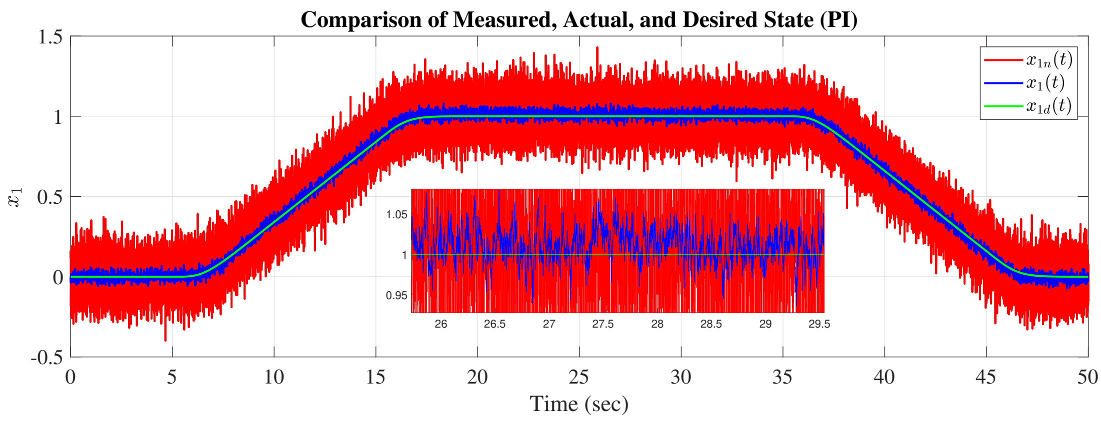
Figure 4 compares the tracking performance for PI controller and PSB controller. It can be seen that the state chattering occurs for the PI controller but for the PSB controller does not. For the PSB controller, the predicted states very close to the actual states. The same observation can be found in Figure 5. Therefore, it can deduce that the performance of the proposed prediction-based control scheme is better than traditional PI control.

The evolution of the control signal for PI and PSB controller is shown in Figure 6. Apparently, the chattering of the control signal is induced due to the noisy state feedback in the case of the PI controller. In contrast, by using the predicted states in the feedback process, the chattering of the control can be significantly improved, and the tracking performance is also increased.

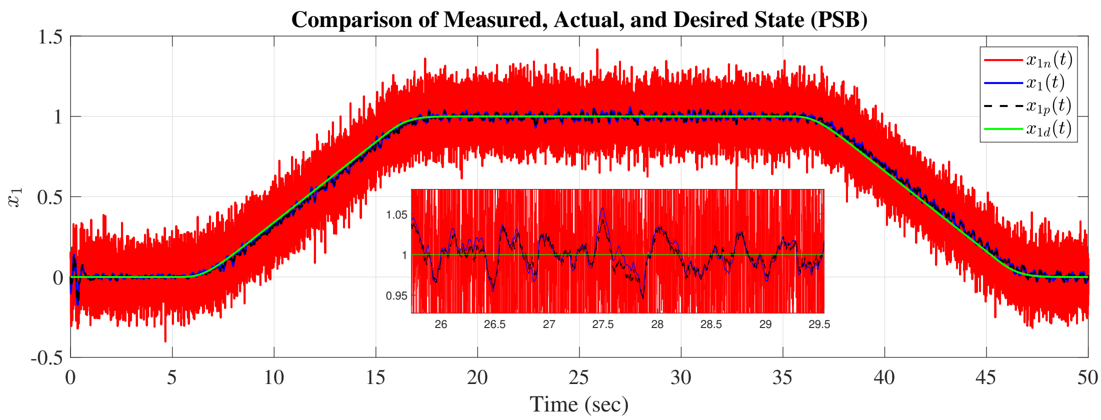


(a) Noise of $x_1(t)$ ($\sigma = 0.1$). (b) Noise of $x_2(t)$ ($\sigma = 0.005$). (c) Noise of $x_3(t)$ ($\sigma = 0.003$).

Figure 3: Evolution of measured noise.

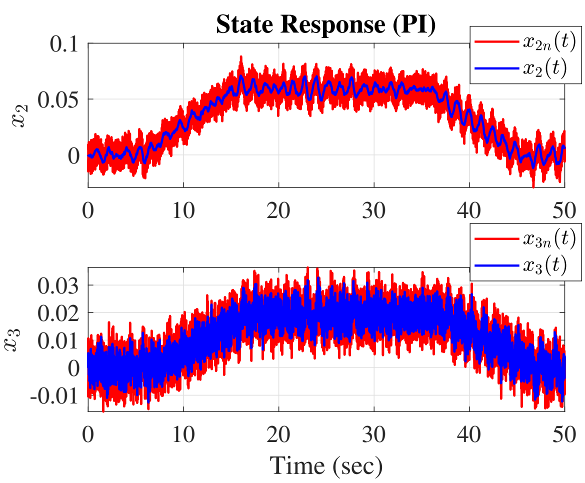


(a) PI controller.

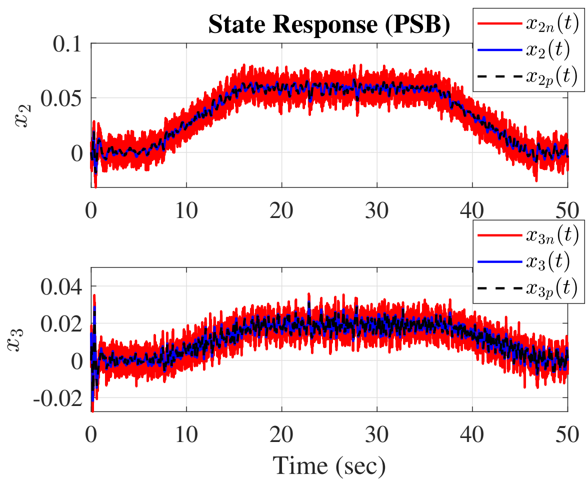


(b) PSB controller.

Figure 4: Comparison of measured, actual, and desired state for PI controller and PSB controller.

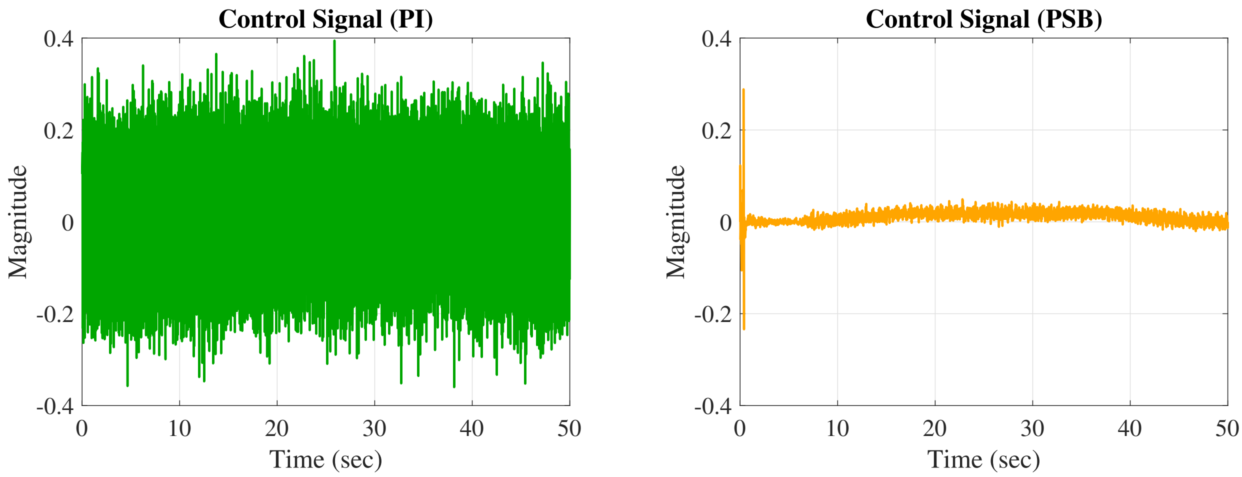


(a) PI controller.



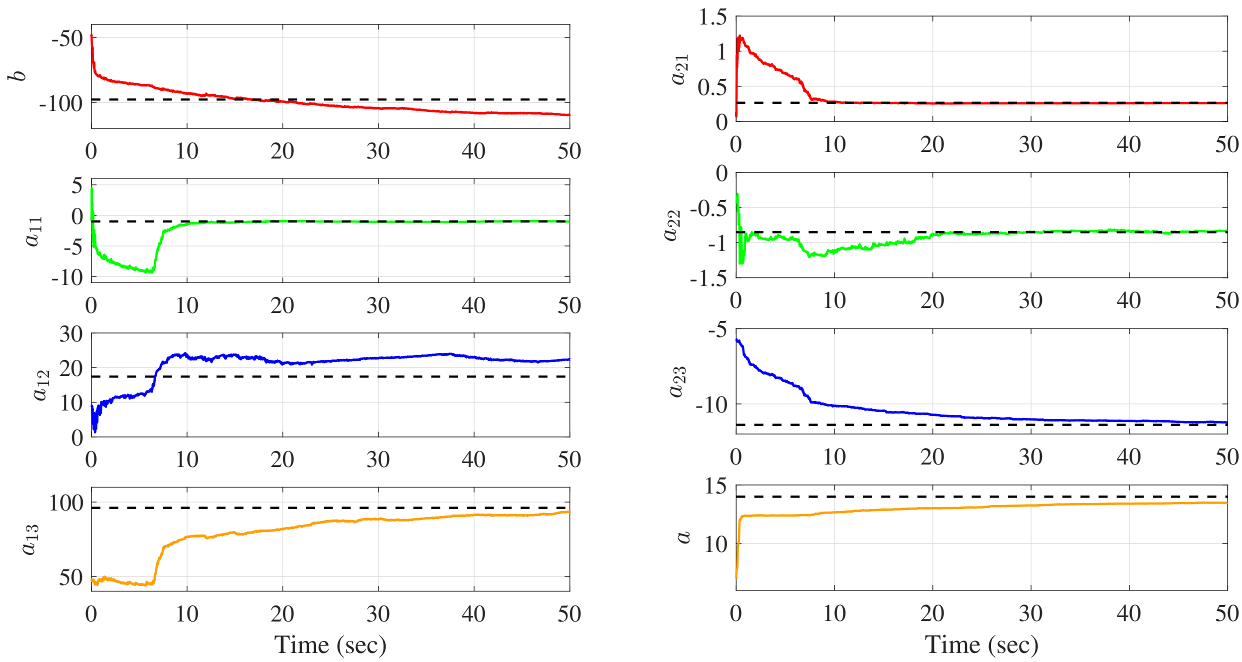
(b) PSB controller.

Figure 5: State response for PI and PSB controller.



(a) PI controller. (b) PSB controller.

Figure 6: Evolution of control signal for PI and PSB controller.



(a) b, a_{11}, a_{12}, a_{13}. (b) a_{21}, a_{22}, a_{23}, a.

Figure 7: Evolution of parameter estimates.

The evolution of parameter estimates are shown in Figure 7. The final values of the estimated parameters are (the parentheses display the true value and absolute relative error)

$$
\begin{array}{llll}
b = -109.6195 & (-97.78)(12.1\%); & a_{21} = 0.2609 & (0.2648)(1.48\%) \\
a_{11} = -1.0067 & (-0.9896)(1.73\%); & a_{22} = -0.8338 & (-0.8512)(2.05\%) \\
a_{12} = 22.3809 & (17.41)(28.55\%); & a_{23} = -11.2413 & (-11.39)(1.31\%) \\
a_{13} = 93.3992 & (96.15)(2.86\%); & a = 13.4838 & (14)(3.69\%)
\end{array} \tag{30}
$$

It can be found that the estimated values converge to the near of the true values.

7 Conclusion and Future Work

In this project, propose a real-time parameter estimation and prediction-based control scheme. By means of introducing the integral operator, the performance of parameter estimates is increased. The corresponding prediction model also gives smooth predicted states such that the control chattering can be avoided. The simulation result indicates that the proposed prediction-based controller is better than the traditional PI controller. The benefits and effectiveness of real-time parameter estimation prediction-based control have been revealed.

Regarding this research, some future work needs to be completed. The design of the more complicated nonlinear dynamic model should be taken into consideration. The technique such as unknown input reconstruction/disturbance observer should be studied so that the RLS estimator has accurate parameter estimates. The experimental study should be performed to validate the theoretical study.

References

[1] K. J. Åström and B. Wittenmark. *Adaptive control*. Addison-Wesley, 2 edition, 1995.

MATLAB Scripts and SIMULINK Block Diagram

The simulation of this work is carried out by the SIMULINK simulator. The block diagram and MATLAB scripts are shown as follows.

SIMULINK Block Diagram

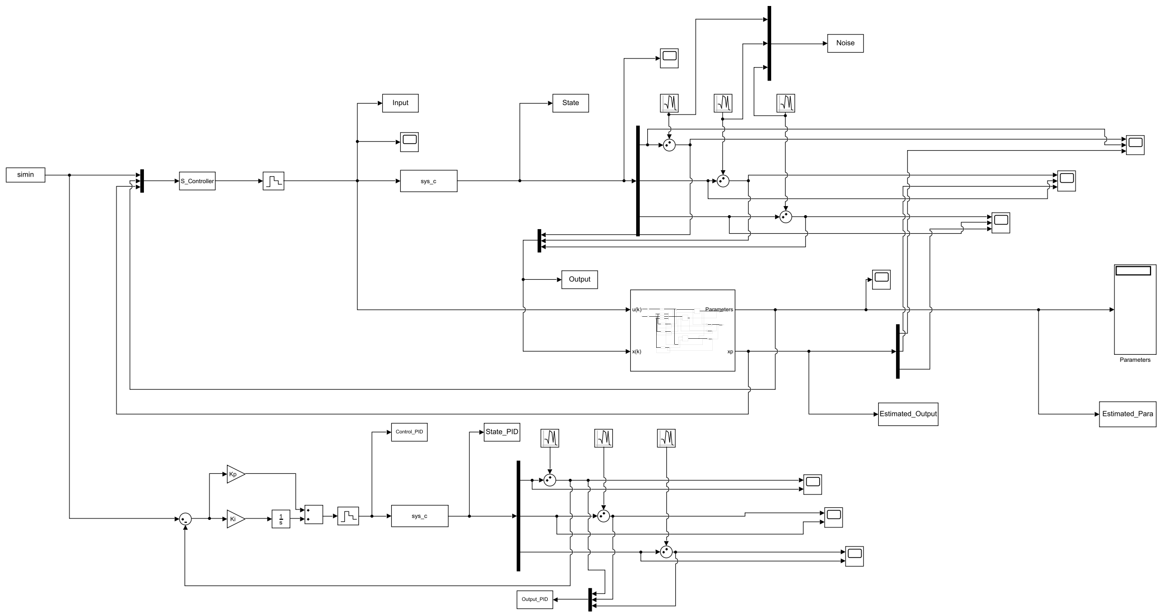


Figure 8: SIMULINK block diagram.

Main_Aircraft_Cmd.m

```
clear; clc; close all;

Ts     = 0.001;
T_end = 50;
t      = (0:Ts:T_end)';

y = y_ramp(5,15,25,Ts,1);
y = [y; flipud(y(1:end-1))];

s = tf('s');
Fc  = 0.5;
tau = 1/(2*pi*Fc);
G = 1/(tau*s+1)^5;

```

```
yf = lsim(G,y,t);

save('Data\Cmd','yf','t');
```

Main_Aircraft_RLS_Control.m

```
clear; clc; close all;

addpath('Subroutines');

cmd = load('Data\Cmd.mat');
pid = load('Data\PID_nominal.mat');

t  = cmd.t;
yd        = cmd.yf;
StepSize = 0.001;
Ts        = t(2)-t(1);
T_zoh     = t(2)-t(1);
T_end     = t(end);

% Reference Command
simin.time                = t;
simin.signals.values      = yd;
simin.signals.dimensions = 1;

% System Parameters
[Ac,Bc,X_True] = Aircraft_Para(1); C = eye(3); D = zeros(3,1);
sys_c = ss(Ac,Bc,C,D);
u_max = 20;
u_min = 20;
A_aug = [Ac zeros(3,1); 1 0 0 0];
B_aug = [Bc; 0];
[K_aug,S,E] = lqr(A_aug,B_aug,eye(4),10000000);

% Integral operator
lambda   = 15;
sys_f_1 = ss(-lambda,-lambda,1,0);  sys_f_1 = c2d(sys_f_1,Ts);
```

```
sys_f_2 = ss(-lambda,1,1,0); sys_f_2 = c2d(sys_f_2,Ts);

P_cov = 1e+4;
X0 = 0.5.*X_True;

% Moise level
N1 = (1*0.1)^2;
N2 = (0.05*0.1)^2;
N3 = (0.03*0.1)^2;

% Control Gains
para.Pole = [-10 -12 -13 -14];

Kp = pid.G_PID.Kp;
Kd = pid.G_PID.Kd;
Ki = pid.G_PID.Ki;
Tf = pid.G_PID.Tf;

simout = sim('Sim_Aircrft_RLS_Control');

save('Data\Sim_Data');
```