Improved High-Order Model Free Adaptive Control
Mathematical Proof and Simulation
Machine Learning Meets Control Theory Mid-Term

Ibrahim Elshenhapy (300389386)

July 17, 2023

Contents

1 Introduction **1**

2 Mathematical Proof **1**
2.1 Model free adaptive control . . . 1
2.1.1 Derive the associated control equations For MFAC . . . 1
2.1.2 Derive the recursive pseudo-partial derivative equations For MFAC . . . 2
2.2 Improved High-Order Model free adaptive control . . . 3
2.2.1 Derive the associated control equations For IHMFAC (Similar to section 2.1.1 with minor edits) . . . 3
2.2.2 Derive the recursive pseudo-partial derivative equations For IHMFAC . . . 4

3 MATLAB Simulation **5**
3.1 My results summarizing the results of the simulation cases . . . 5

1 Introduction

Model free adaptive control (MFAC) is a data-driven control method that does not require knowledge of the system dynamics. MFAC has been shown to be effective for controlling systems with time-varying, parameter uncertainties, or unknown disturbances. In this paper, an improved high-order MFAC (IHOMFAC) method is proposed. Where IHOMFAC takes more previous information into account in the parameter estimation algorithm, which improves the control performance. Simulation results confirm that IHOMFAC has superior control performance than the traditional high-order MFAC.

2 Mathematical Proof

2.1 Model free adaptive control

2.1.1 Derive the associated control equations For MFAC

Starting with the control input cost function

$$J(u(t), \alpha_{t,i}) = |y^*(t+1) - y(t+1)|^2 + \lambda \left| u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right|^2 \tag{1}$$

Using Equation of the nonlinear system

$$\Delta y(t+1) = \phi(t) \Delta u(t) \tag{2}$$

Get y(t+1) in terms of u(t)

$$y(t+1) = \phi(t)u(t) - \phi(t)u(t-1) + y(t) \tag{3}$$

Substitute (3) in (1)

$$J(u(t),\alpha_{t,i}) = |y^*(t+1) - (\phi(t)u(t) - \phi(t)u(t-1) + y(t))|^2 + \lambda \left| u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right|^2 \tag{4}$$

Partial Derivative the cost function with respect to u(t)

$$\frac{\partial J(u(t),\alpha_{t,i})}{\partial u(t)} = 2\left(y^*(t+1) - (\phi(t)u(t) - \phi(t)u(t-1) + y(t))\right)(-\phi(t)) + 2\lambda \left(u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right) \tag{5}$$

According to the optimal condition

$$\frac{1}{2}\frac{\partial J(u(t),\alpha_{t,i})}{\partial u(t)} = 0 \tag{6}$$

$$\frac{1}{2}(2\left(y^*(t+1) - (\phi(t)u(t) - \phi(t)u(t-1) + y(t))\right)(-\phi(t)) + 2\lambda \left(u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right)) = 0$$

$$u(t)(\lambda + \phi(t)^2) = \phi(t)y^*(t+1) + \phi(t)^2 u(t-1) - \phi(t)y(t)) + \lambda \left(\sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right)$$

Reorder the equation and add ρ where ρE[0,1] is an artificially added factor to make the controller more flexible.

$$u(t)(\lambda + \phi(t)^2) = \phi(t)^2 u(t-1) + \lambda \left(\sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right) + \rho\phi(t)(y^*(t+1) - y(t))$$

The MFAC control law can be rewritten

$$u(t) = \frac{\phi(t)^2}{\lambda + \phi(t)^2} u(t-1) + \frac{\lambda}{\lambda + \phi(t)^2} \sum_{i=1}^{l} \alpha_{t,i} u(t-i) + \frac{\rho\phi(t)}{\lambda + \phi(t)^2}(y^*(t+1) - y(t)) \tag{7}$$

It's only possible to use the estimate value for ϕ to be $\hat{\phi}$

$$u(t) = \frac{\hat{\phi}(t)^2}{\lambda + \hat{\phi}(t)^2} u(t-1) + \frac{\lambda}{\lambda + \hat{\phi}(t)^2} \sum_{i=1}^{l} \alpha_{t,i} u(t-i) + \frac{\rho\hat{\phi}(t)}{\lambda + \hat{\phi}(t)^2} e(t) \tag{8}$$

Where e(t) is the error $= y^*(t+1) - y(t)$

2.1.2 Derive the recursive pseudo-partial derivative equations For MFAC

Starting from estimating the parameter ϕ(t). Consider the following cost function

$$J(\hat{\phi}(t)) = \left| y(t) - y(t-1) - \hat{\phi}(t)\Delta u(t-1) \right|^2 + \mu \left| \hat{\phi}(t) - \hat{\phi}(t-1) \right|^2 \tag{9}$$

Take the derivative of this cost function with respect to $\hat{\phi}$(t).

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 2(y(t) - y(t-1) - \hat{\phi}(t)\Delta u(t-1))(-\Delta u(t-1)) + 2\mu(\hat{\phi}(t) - \hat{\phi}(t-1)) \tag{10}$$

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 2(y(t) - y(t-1)) - 2\mu\hat{\phi}(t) + 2\mu\hat{\phi}(t-1) - 2\hat{\phi}(t)\Delta u(t-1)$$

Set it equal to zero

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 0 \tag{11}$$

$$0 = 2(y(t) - y(t-1)) - 2\mu\hat{\phi}(t) + 2\mu\hat{\phi}(t-1) - 2\hat{\phi}(t)\Delta u(t-1)$$

$$\hat{\phi}(t) = \frac{\Delta y(t)\Delta u(t-1) + \mu\hat{\phi}(t-1)}{\mu + \Delta u(t-1)^2} \tag{12}$$

Consider incorporating ideas from the Kaczmarz projection algorithm where the current state is equal to the pervious state added to function (So add then subtract the pervious estimate)

$$\hat{\phi}(t) = \hat{\phi}(t-1) - \hat{\phi}(t-1) + \frac{\Delta y(t)\Delta u(t-1) + \mu\hat{\phi}(t-1)}{\mu + \Delta u(t-1)^2} \tag{13}$$

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{(-\hat{\phi}(t-1))(\mu + \Delta u(t-1)^2) + \Delta y(t)\Delta u(t-1) + \mu\hat{\phi}(t-1)}{\mu + \Delta u(t-1)^2}$$

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{(-\hat{\phi}(t-1))(\Delta u(t-1)^2) + \Delta y(t)\Delta u(t-1)}{\mu + \Delta u(t-1)^2}$$

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{\Delta u(t-1)(\Delta y(t) - \Delta u(t-1)\hat{\phi}(t-1))}{\mu + \Delta u(t-1)^2} \tag{14}$$

Use η where η is the step sequence to make the algorithm more general.

$$\hat{\phi}(t) = \hat{\phi}(t-1) + \frac{\eta\Delta u(t-1)}{\mu + \Delta u(t-1)^2}(\Delta y(t) - \Delta u(t-1)\hat{\phi}(t-1)) \tag{15}$$

2.2 Improved High-Order Model free adaptive control

2.2.1 Derive the associated control equations For IHMFAC (Similar to section 2.1.1 with minor edits)

Starting with the control input cost function as used in equation 1

$$J(u(t), \alpha_{t,i}) = |y^*(t+1) - y(t+1)|^2 + \lambda \left| u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right|^2 \tag{16}$$

Using Equation of the nonlinear system but as it only possible to use the estimate value as used in equation 2

$$\Delta y(t+1) = \hat{\phi}(t)\Delta u(t) \tag{17}$$

Get y(t+1) in terms of u(t)

$$y(t+1) = \hat{\phi}(t)u(t) - \hat{\phi}(t)u(t-1) + y(t) \tag{18}$$

Substitute (18) in (16)

$$J(u(t), \alpha_{t,i}) = |y^*(t+1) - (\hat{\phi}(t)u(t) - \hat{\phi}(t)u(t-1) + y(t))|^2 + \lambda \left| u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i) \right|^2 \tag{19}$$

Partial Derivative the cost function with respect to u(t)

$$\frac{\partial J(u(t), \alpha_{t,i})}{\partial u(t)} = 2\left(y^*(t+1) - (\hat{\phi}(t)u(t) - \hat{\phi}(t)u(t-1) + y(t))\right)(-\hat{\phi}(t)) + 2\lambda\left(u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i)\right) \tag{20}$$

According to the optimal condition

$$\frac{1}{2}\frac{\partial J(u(t), \alpha_{t,i})}{\partial u(t)} = 0 \tag{21}$$

$$\frac{1}{2}(2\left(y^*(t+1) - (\hat{\phi}(t)u(t) - \hat{\phi}(t)u(t-1) + y(t))\right)(-\hat{\phi}(t)) + 2\lambda\left(u(t) - \sum_{i=1}^{l} \alpha_{t,i} u(t-i)\right)) = 0$$

$$u(t)(\lambda + \hat{\phi}(t)^2) = \hat{\phi}(t)y^*(t+1) + \hat{\phi}(t)^2 u(t-1) - \hat{\phi}(t)y(t)) + \lambda\left(\sum_{i=1}^{l} \alpha_{t,i} u(t-i)\right)$$

Reorder the equation and add ρ where ρE(0,1] is an artificially added factor to make the controller more flexible.

$$u(t)(\lambda + \hat{\phi}(t)^2) = \hat{\phi}(t)^2 u(t-1) + \lambda\left(\sum_{i=1}^{l} \alpha_{t,i} u(t-i)\right) + \rho\hat{\phi}(t)(y^*(t+1) - y(t))$$

The Improved high-order IHMFAC control law can be rewritten

$$u(t) = \frac{\hat{\phi}(t)^2}{\lambda + \hat{\phi}(t)^2} u(t-1) + \frac{\lambda}{\lambda + \hat{\phi}(t)^2} \sum_{i=1}^{l} \alpha_{t,i} u(t-i) + \frac{\rho \hat{\phi}(t)}{\lambda + \hat{\phi}(t)^2} e(t) \tag{22}$$

Where e(t) is the error $= y^*(t+1) - y(t)$

2.2.2 Derive the recursive pseudo-partial derivative equations For IHMFAC

Starting from estimating the parameter ϕ(t). Consider the following cost function.

$$J\left(\hat{\phi}(t), \beta_{t,i}\right) = |y(t) - y(t-1) - \hat{\phi}(t)\Delta u(t-1)|^2 + \mu \left| \hat{\phi}(t) - \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) \right|^2 \tag{23}$$

where $\beta_{t,i}$ is a weight factor, and $\sum_{i=1}^{l} \beta_{t,i}$=1, which represents different degrees of influence of the state of the previous moment on the current state.

Take the derivative of this cost function with respect to $\hat{\phi}$(t).

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 2(y(t) - y(t-1) - \hat{\phi}(t)\Delta u(t-1))(-\Delta u(t-1)) + 2\mu(\hat{\phi}(t) - \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)) \tag{24}$$

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 2(y(t) - y(t-1)) - 2\mu\hat{\phi}(t) + 2\mu \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) - 2\hat{\phi}(t)\Delta u(t-1)$$

Set it equal to zero

$$\frac{dJ(\hat{\phi}(t))}{d\hat{\phi}(t)} = 0 \tag{25}$$

$$0 = 2(y(t) - y(t-1)) - 2\mu\hat{\phi}(t) + 2\mu \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) - 2\hat{\phi}(t)\Delta u(t-1)$$

$$\hat{\phi}(t) = \frac{\Delta y(t)\Delta u(t-1) + \mu \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)}{\mu + \Delta u(t-1)^2} \tag{26}$$

Consider incorporating ideas from the Kaczmarz projection algorithm where the current state is equal to the pervious state added to function (So add then subtract the pervious estimate ($\sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)$))

$$\hat{\phi}(t) = \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) - \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) + \frac{\Delta y(t)\Delta u(t-1) + \mu \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)}{\mu + \Delta u(t-1)^2} \tag{27}$$

$$\hat{\phi}(t) = \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) + \frac{(-\sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i))(\mu + \Delta u(t-1)^2) + \Delta y(t)\Delta u(t-1) + \mu \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)}{\mu + \Delta u(t-1)^2}$$

$$\hat{\phi}(t) = \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) + \frac{(-\sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i))(\Delta u(t-1)^2) + \Delta y(t)\Delta u(t-1))}{\mu + \Delta u(t-1)^2}$$

$$\hat{\phi}(t) = \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) + \frac{\Delta u(t-1)(\Delta y(t) - \Delta u(t-1) \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i)}{\mu + \Delta u(t-1)^2} \tag{28}$$

Use η where η is the step sequence to make the algorithm more general.

$$\hat{\phi}(t) = \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) + \frac{\eta \Delta u(t-1)}{\mu + \Delta u(t-1)^2} \left(\Delta y(t) - \Delta u(t-1) \sum_{i=1}^{l} \beta_{t,i} \hat{\phi}(t-i) \right) \tag{29}$$

Use μ as small positive value such that the denominator is non-zero value

3 MATLAB Simulation

You can find the code used for simulation (.m file)here. As it made using Matlab, To run the code you will need to open it in Matlab and no additional setup needed.

3.1 My results summarizing the results of the simulation cases

The study focuses on improving the control performance of a nonlinear system with variable characteristics. The two algorithms compared, high-order MFAC and IHOMFAC, aim to track a desired trajectory effectively. The simulation results show the advantages of the IHOMFAC algorithm over the standard MFAC algorithm. Where The IHOMFAC algorithm provides better tracking performance, especially when there are sudden changes in the desired trajectory.

In Figure 1, the tracking performance of both algorithms is shown, with the desired output indicated by the black line. It is observed that both algorithms can achieve good control performance, but the IHOMFAC algorithm shows superior control effectiveness. However, there is significant fluctuation in the response when there are step changes in the reference trajectory, which intensifies over time. This indicates that the tracking performance is not ideal under dramatic changes in expectations.

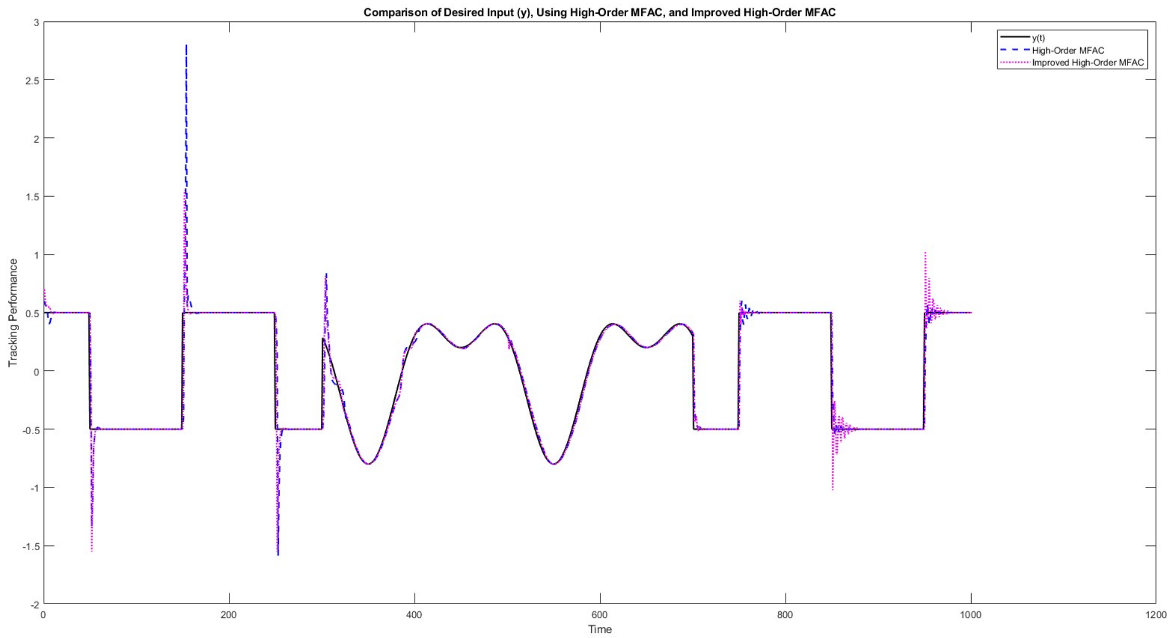

Figure 1: The tracking performance

The improved algorithm incorporates more control knowledge by utilizing previous time data in both the control input and parameter estimation. This additional information enhances the flexibility and adaptability of the control law. By considering past observations, the IHOMFAC algorithm can better capture the system dynamics and make more informed control decisions. This leads to reduced overshoot, minimized fluctuation, and faster tracking speed, as observed in the simulation results.

To highlight the advantages of the improved higher-order algorithm, local enlarged graphs for the time intervals 785-820 and 920-1000 are provided in Figures 2 and 3, respectively. It is evident that the IHOMFAC algorithm exhibits smaller overshoot, reduced fluctuation, and faster tracking speed compared to the standard MFAC algorithm.

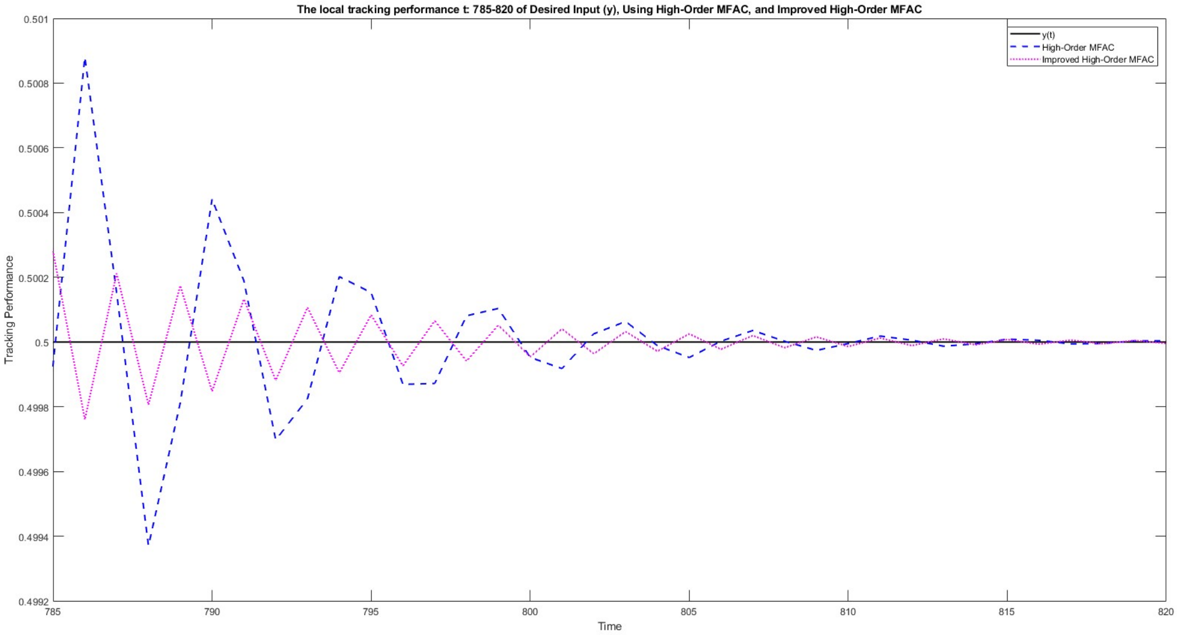

Figure 2: The local tracking performance t: 785-820

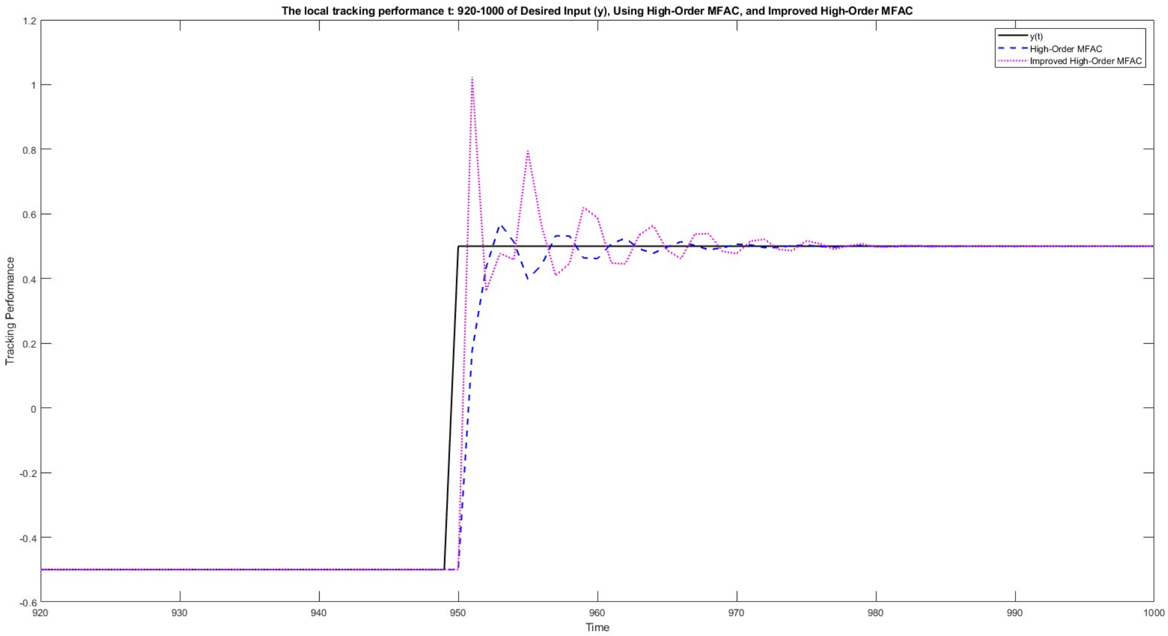

Figure 3: The local tracking performance t: 920-1000

Another noteworthy aspect is the comparison of the estimated parameter values and control inputs. Analyzing these quantities provides insights into the behavior and performance of the control algorithms. It allows for a deeper understanding of how the algorithms respond to changes in the system and how they adapt their control actions accordingly.

To understand the reasons behind the performance changes, the Parameterized Polynomial Derivative (PPD) estimated values and control inputs are shown in Figures 4 and 5, respectively. Figure 4 reveals the change in the PPD estimated values, which directly affect the control input and output. Figure 5 further illustrates the relationship between the control input and output.

Figure 5 further illustrates the relationship between the control input and output.

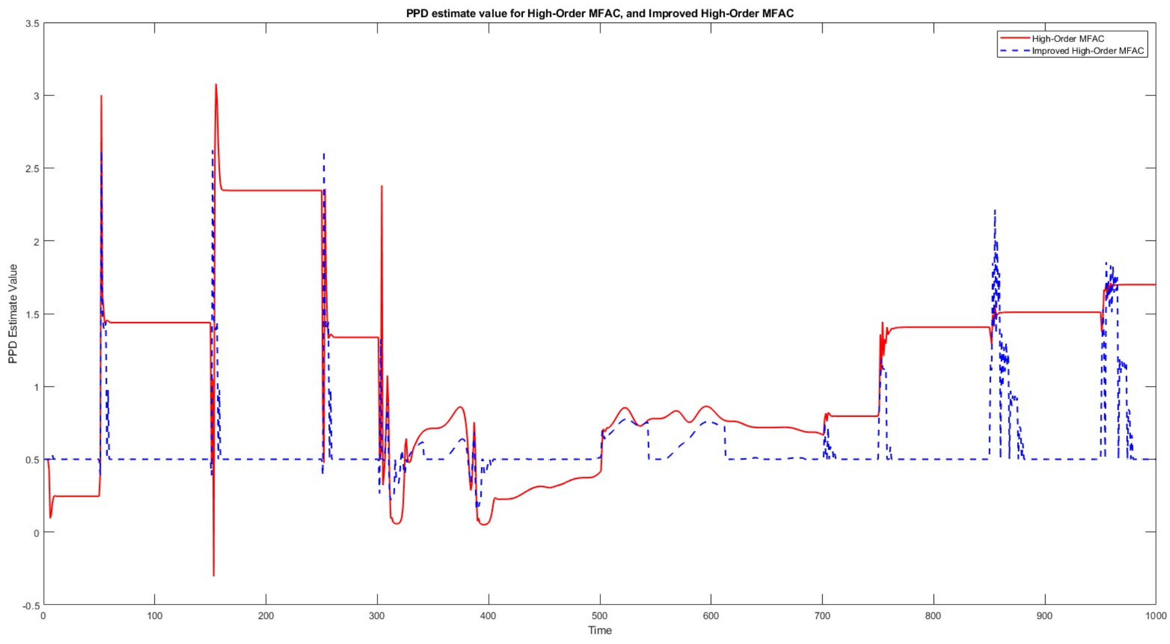

Figure 4: PPD estimated value

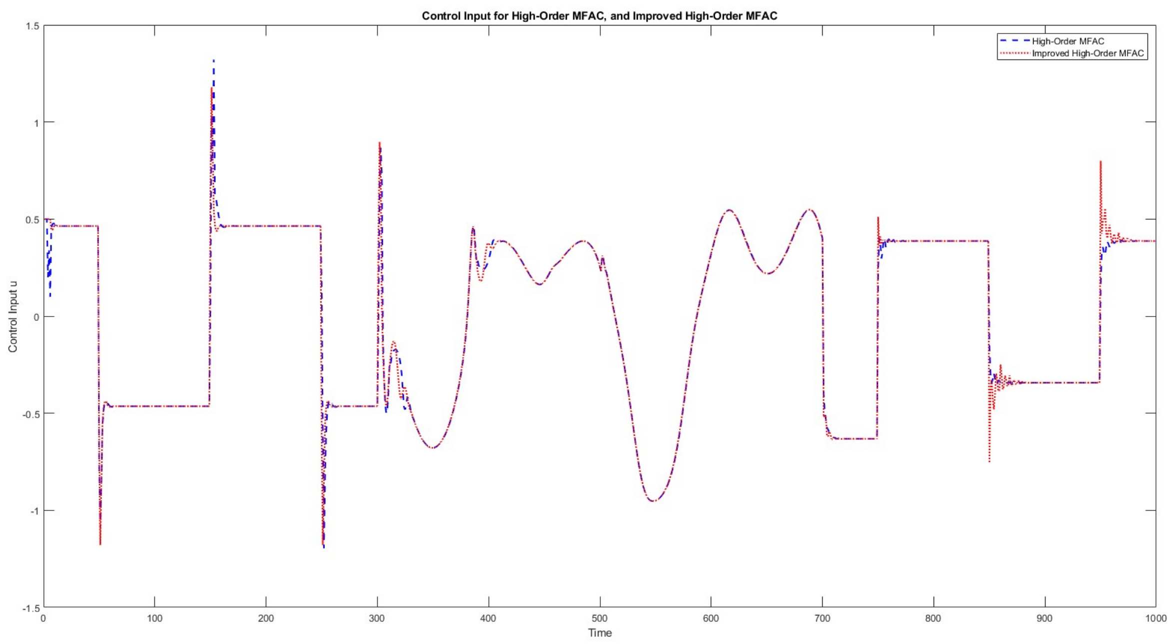

Figure 5: Control input

Overall, the IHOMFAC algorithm demonstrates improved control performance compared to the standard MFAC algorithm. By incorporating more control knowledge into the algorithm and considering the previous time data in both the control input and parameter estimation algorithms, the IHOMFAC algorithm provides better flexibility and optimization. The simulation results validate the effectiveness of the improved high-order model-free adaptive control in optimizing control performance.