Extended Predictive Functional Controller (EPFC)

Introduction

The following exemplar system is used in simulations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.33e^{-x_1}x_1 - 1.1x_2 + u \\ y = x_1 \end{cases} \tag{1}$$

In Section 1, we first examine the nonlinear system and introduce a variable transformation. By using this transformation, we will be able to apply inputs over a wider range, thereby covering all points of the static gain plot.

In Section 2, we will provide a summary of the design process of the EPFC model predictive controller and review its underlying concept.

In Section 3, we will observe the results of applying the controller to the system.

Section 1: Examining the Nonlinear System

The exemplar system used is as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.33e^{-x_1}x_1 - 1.1x_2 + u \\ y = x_1 \end{cases} \tag{2}$$

To calculate the static gain, we just have to set $\dot{x}_1 = 0$ and $\dot{x}_2 = 0$.

$$\dot{x}_1 = x_2 = 0, \qquad \dot{x}_2 = -0.33e^{-x_1}x_1 - 1.1x_2 + u = 0 \Rightarrow u = 0.33e^{-x_1}x_1 \tag{3}$$

State x_1 is the system output, provided in fig. 1.

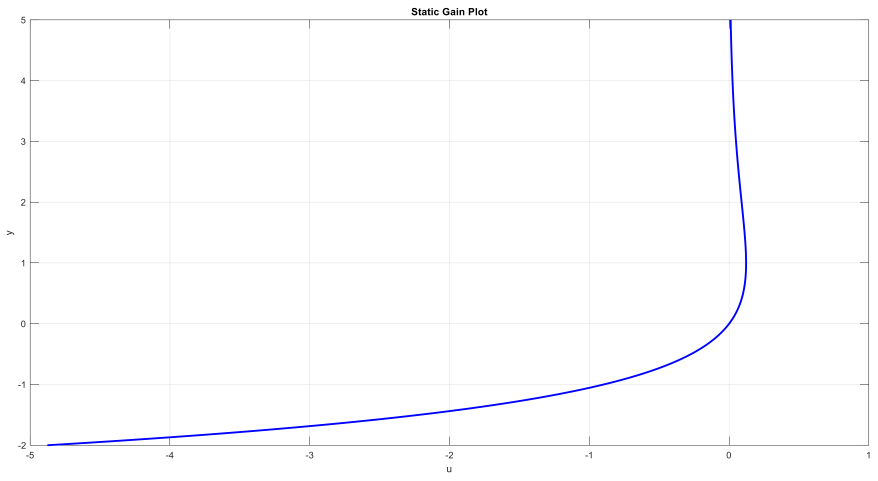

Figure1 : static gain plot for input u and output x_1

As we can see, in the region where $u > 0$, for a given value of u, two different outputs may be possible. The resulting output depends on the previous state of the system and other factors, which determine which of the two outcomes will occur.

In order to operate effectively in the $u > 0$ region as well, we need a variable transformation that renders the system's static gain one-to-one. To achieve this, we consider the following variable transformation:

$$u = -x_1 + v \Rightarrow v = u + x_1 \tag{4}$$

Meaning we consider the new system of equations in (5).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.33e^{-x_1}x_1 - 1.1x_2 - x_1 + v \\ y = x_1 \end{cases} \tag{5}$$

We consider the variable v as the control variable and employ model predictive control to determine its value. It is important to note, however, that the actual signal applied to the system remains u, and v is used solely as an intermediate variable-in effect, we are implementing an internal loop around the closed-loop system.

The static gain plot of the new system, from input v to output x_1, is presented in fig. 2.

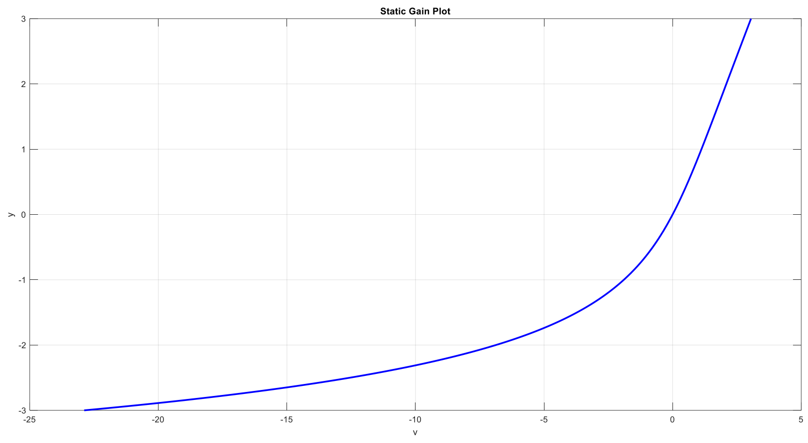

Figure 2: static gain from input v to output x_1

We observe that with the new input v, the system's static gain becomes one-to-one. We choose the operating point as $x_1 = 0.5$, so that significant portions of both the negative and positive ranges are covered, and transitions between all regions of operation can be observed (the output will approximately vary between -2 and 3).

Referring to Fig. 1, we can see that in the region starting approximately from $y = 1$, the DC gain of the system obtained through linearization becomes negative. If we were to use the input u directly, this could lead to complications. However, by using the variable v, this issue is avoided.

These results will be clearly observed in the simulation section.

Section 2: EPFC Controller Design

2.1 PFC Controller Design

In a PFC controller, the control input or its derivatives are considered to be an expansion of known basis functions:

$$\dot{u}(t) = \alpha_0\varphi_0(t) + \alpha_1\varphi_1(t) + \cdots + \alpha_n\varphi_n(t) \tag{6}$$

We do the same thing. We expand the derivative of the control signal as to make the system type equal to 1. We use pulse functions as the basis functions, with their width in time equal to our control sampling time. Then we can write:

$$\Delta u(t_i) = u(t_i + T_s) - u(t_i) = \int_{t_i}^{t_i+T_s} \dot{u}(\tau)d\tau = \alpha_i T_s \Rightarrow \alpha_i = \frac{\Delta u(t_i)}{T_S} \tag{7}$$

In a PFC controller, instead of manipulating the next M samples of the input to control the next P samples of the output, we consider coincidence points. For example, if we have three input coincidence points and three output coincidence points, we try to find $u(t)$, $u(t+m_1)$ and $u(t+m_2)$ such that $y(t+\mu_1)$, $y(t+\mu_2)$ and $y(t+\mu_3)$ reach their desired values at $y_d(t+\mu_1)$, $y_d(t+\mu_2)$ and $y_d(t+\mu_3)$. In between the input coincidence points, the control input is considered to be constant.

The reference signal is considered to be unprogrammed, meaning we do not know its future values beforehand. We use the following relation (filter) to determine $y_d(t+\mu_1)$:

$$y_d(t+\mu_i) = \psi^{\mu_i}y(t) + (1-\psi^{\mu_i})r(t), \qquad i = 1,2,3 \tag{8}$$

Assuming three input coincidence points, we have to use the following relations to calculate Δu:

$$\begin{cases} \psi^{\mu_1}y(t) + (1-\psi^{\mu_1})r(t) = g_{\mu_1}\Delta u(t) + g_{\mu_1-m_1}\Delta u(t+m_1) + g_{\mu_1-m_2}\Delta u(t+m_2) + Y_{past}(t+\mu_1) + d(t) \\ \psi^{\mu_2}y(t) + (1-\psi^{\mu_2})r(t) = g_{\mu_2}\Delta u(t) + g_{\mu_2-m_1}\Delta u(t+m_1) + g_{\mu_2-m_2}\Delta u(t+m_2) + Y_{past}(t+\mu_2) + d(t) \\ \psi^{\mu_3}y(t) + (1-\psi^{\mu_3})r(t) = g_{\mu_3}\Delta u(t) + g_{\mu_3-m_1}\Delta u(t+m_1) + g_{\mu_3-m_2}\Delta u(t+m_2) + Y_{past}(t+\mu_3) + d(t) \end{cases} \tag{9}$$

Solving the three equations above, $\Delta u(t)$, $\Delta u(t+m_1)$ and $\Delta u(t+m_2)$ will be determined.

If the number of matching points for the input and output is not equal, having either fewer or more degrees of freedom than required may result in the set of equations (9) having no solution or a non-unique solution.

Therefore, an alternative approach is to define a cost function instead of enforcing the output to match a set of desired points exactly. This cost function can then be minimized.

The objective function is defined as shown in equation (10) (for two coincidence points).

$$J = \left(y_d(t+\mu_1) - y_p(t+\mu_1)\right)^2 q_1 + \left(y_d(t+\mu_2) - y_p(t+\mu_2)\right)^2 q_2 + \\ \Delta u^2(t)r_1 + \Delta u^2(t+m_1)r_2 \tag{10}$$

By minimizing this objective function, the values of Δu can be determined. We will formulate and solve these equations for an example with two input matching points and three output matching points. These results will later be used in subsequent sections.

$$y_{\mathrm{p}}(t+\mu_i) = g_{\mu_i}\Delta u(t) + g_{\mu_i - m_i}\Delta u(t+m_1) + Y_{past}(t+\mu_i) + d(t), i = 1,2,3 \quad (11)$$

$$y_d(t+\mu_i) = \psi^{\mu_i}y(t) + (1-\psi^{\mu_i})r(t), \qquad i = 1,2,3 \quad (12)$$

We define e_i as:

$$e_i = y_d(t+\mu_i) - Y_{past}(t+\mu_i) - d(t), \qquad i = 1,2,3 \quad (13)$$

Therefore, we have

$$y_d(t+\mu_i) - y_p(t+\mu_i) = e_i - g_{\mu_i}\Delta u(t) - g_{\mu_i - m_i}\Delta u(t+m_i), \qquad i = 1,2,3 \quad (14)$$

The cost function will be:

$$J = q_1\left(e_1 - g_{\mu_1}\Delta u(t) - g_{\mu_1 - m_1}\Delta u(t+m_1)\right)^2 +$$
$$q_2\left(e_2 - g_{\mu_2}\Delta u(t) - g_{\mu_2 - m_1}\Delta u(t+m_1)\right)^2 +$$
$$q_3\left(e_3 - g_{\mu_3}\Delta u(t) - g_{\mu_3 - m_1}\Delta u(t+m_1)\right)^2 +$$
$$r_1\Delta u^2(t) + r_2\Delta u^2(t+m_1) \quad (15)$$

We use Quadrativ Programming to solve this set of equations. With this method, we can later easily add constraints on control inputs, etc.

Note that we calculate Δu in the discrete domain and apply it to the system. If we were to actually plot $\dot{u}$, we would see that it was an expansion of the basis functions with coefficients $\frac{\Delta u}{T_S}$, but we do not need to do that (because u is calculated in the discrete domain).

2.2 Extended PFC controller Design

To use extended PFC we just have to linearize the system at each sample and use a PFC controller, but a few things should be noted.

Y_{past} or free response which is used in the controller will use the nonlinear model. We can just assume $\Delta u = 0$ (u constant) and move the nonlinear dynamics forward.

$$u(t) = u(t+1) = \cdots = u(t-1) \tag{16}$$

There are several methods available for linearization. The conventional approach involves using the Jacobian matrix. Another method is based on applying perturbations.

For this purpose, we apply the following set of inputs to the model once:

$$u(t) = u(t+1) = \cdots = u(t-1) + \delta \tag{17}$$

We call the output Y_{per}.

By subtracting the outputs Y_{per} and Y_{past} element-wise, the effect of the applied perturbation amplitude δ becomes evident.

The step responses at time t can then be obtained using equation (18).

$$g_i = \frac{y_{per}(t+i) - y_{past}(t+i)}{\delta} \tag{18}$$

To obtain more accurate values for g_i, the perturbation amplitude δ must be chosen small. A value of δ around 10% of $u(t-1)$ is generally appropriate.

In the following simulations, we will use the perturbation method to obtain the step responses.

Another important consideration is the term D. In standard PFC (Predictive Functional Control), D -which represents the difference between the model output and the system output- was primarily due to disturbances. In the EPFC case, however, a portion of D also arises from linearization error.

$$\mathrm{D(t+1)} = \mathrm{D}^{\mathrm{ext}}(t+1) + D^{nl}(t+1) \tag{19}$$

D^{ext} which is the unpredictable part of D is calculated as:

$$D^{ext}(t+1) = d^{ext}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} \tag{20}$$

where d^{ext} is obtained from (21).

$$d^{ext} = y(t) - y^{nl}(t) \tag{21}$$

$D^{nl}(t+1)$ is the part that considers the linearization error. We can see its relation in (22).

$$D^{nl}(\Delta u) = Y^{nl} - Y^{li} \tag{22}$$

Here, Y^{nl} and Y^{li} are obtained by applying Δu to the nonlinear and linear systems, respectively.

Note that since Y^{nl} is a nonlinear function of Δu, the term D^{nl} will also be a nonlinear function of Δu. Therefore, the equation that ultimately needs to be solved to find Δu is nonlinear and does not have a closed-form solution. It must be solved using an iterative method.

The problem to be solved iteratively is expressed in equation (23).

$$h(D^{nl}) = 0, \qquad h(D^{nl}) = Y^{nl} - Y^{eli} \tag{23}$$

Y^{eli} is the approximation of the nonlinear system that is obtained from linearization.

$$Y^{eli} = Y^{li} + D^{nl} \tag{24}$$

Therefore, we begin with an initial guess for D^{nl}. For example, the initial guess can be $D^{nl} = 0$.

Next, by considering D as $D = D^{ext} + D^{nl}$ and using the solution method introduced previously, we determine Δu. This Δu is then applied to both the linear and nonlinear models.

From the difference between their outputs, we obtain the updated value D^{nl}_{l+1}. Then, equation (25) is used to update D^{nl}_{l+1}.

$$D^{nl}_{l+1} = D^{nl}_{l+1} + \beta h\left(D^{nl}_{l+1}\right) \tag{۲۵}$$

$h(D^{nl}_{l+1})$ is defined in (23).

We will continue this iterative method until one of the following two stopping criteria is met:

1. The number of iterations exceeds a predefined limit, denoted as N_i.
2. The difference between D^{nl}_{l} and D^{nl}_{l+1} becomes smaller than a specified tolerance, denoted as TOL.

In the subsequent simulations, this fixed-point iteration method will be employed.

Section 3: Applying the Controller to the System

Now we apply the controller to the system. Discretization time is considered to be $0.2s$ and the following design parameters are used:

$$q = 1, \qquad r = 0.1, \qquad \psi = 0.25, \qquad x0 = [0.5; 0], \qquad u0 = -0.1001$$

Note that q, r and ψ aren't necessarily scalars, but are written like that for abbreviation.

The paramters of the fixed-point iteration method are considered to be

$$\beta = 0.1, \qquad N_i = 5, \qquad TOL = 0.001$$

Perturbation is used as the linearization method and the reference signal is considered to be unprogrammed.

We first consider one input coincidence point and one output coincidence point:

$$m = [0], \qquad \mu = [7]$$

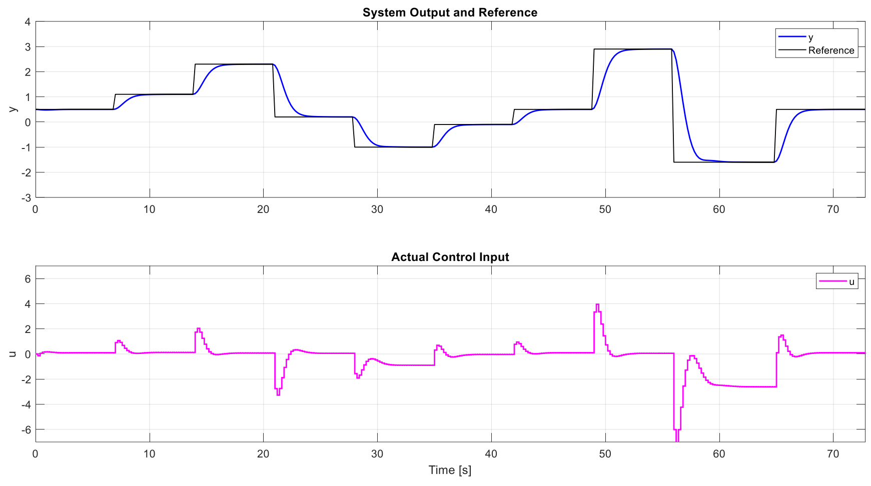

Figure 3: System Output and Control Input

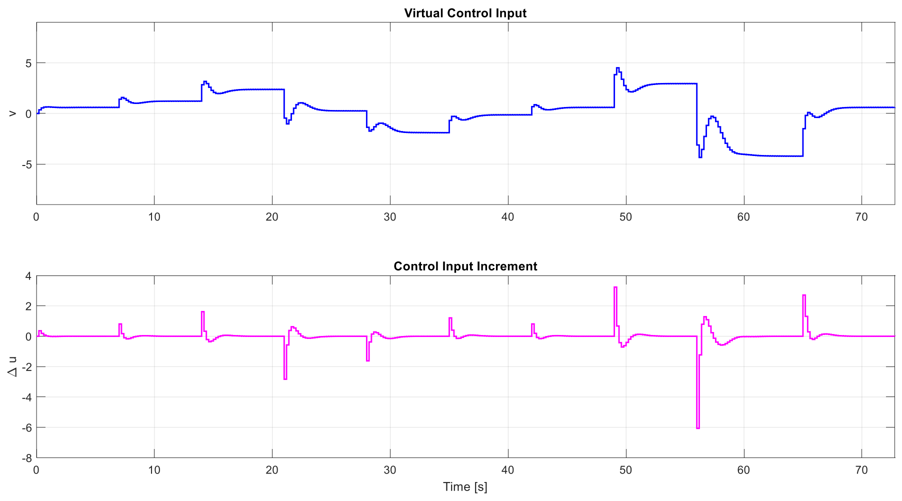

Figure 4: Intermediate Variable v and Δu

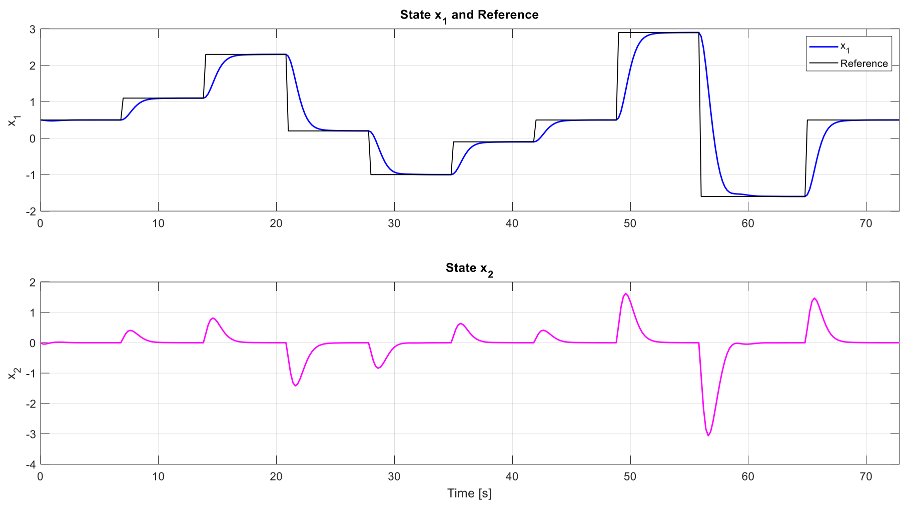

Figure 5: System States

Further Analysis can be done using different coincidence points, adding noise, disturbance and model uncertainty and changing the design parameters. Some figures are included here, but more can be generated using the provided codes.

Noise and disturbance aree considered like this:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.33e^{-x_1}x_1 - 1.1x_2 + u + d \\ y = x_1 + n \end{cases} \tag{۲۶}$$

d is a pulse disturbance and n is the noise added only to the sensor (not the process), but it does indirectly affect the process.

Considering a $5N$ disturbance and a white Gaussian noise with mean of 0 and power of $-30dB$, the results are included.

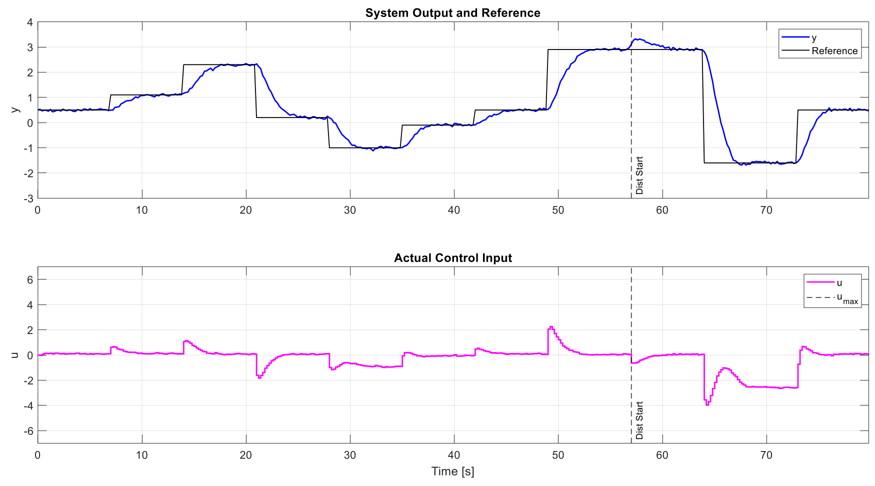

Figure 6: System Output and Control Input in presence of noise and disturbance

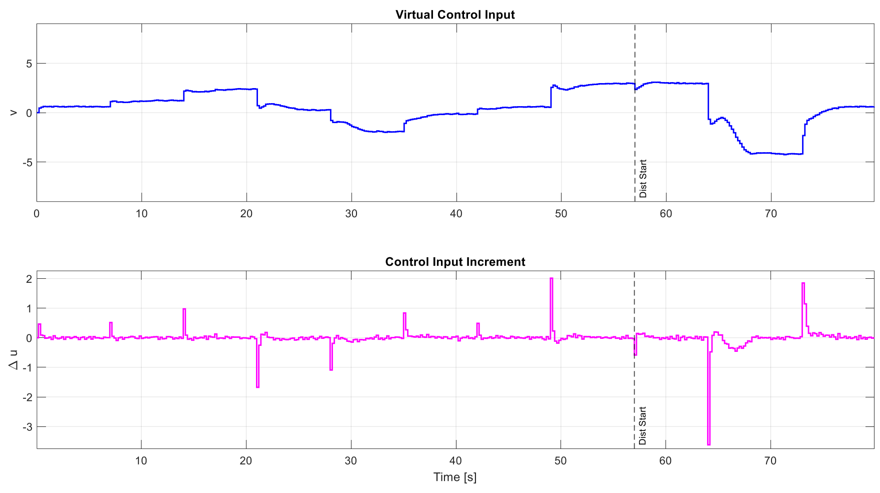

Figure 7: Intermediate Variable v and Δu in presence of noise and disturbance

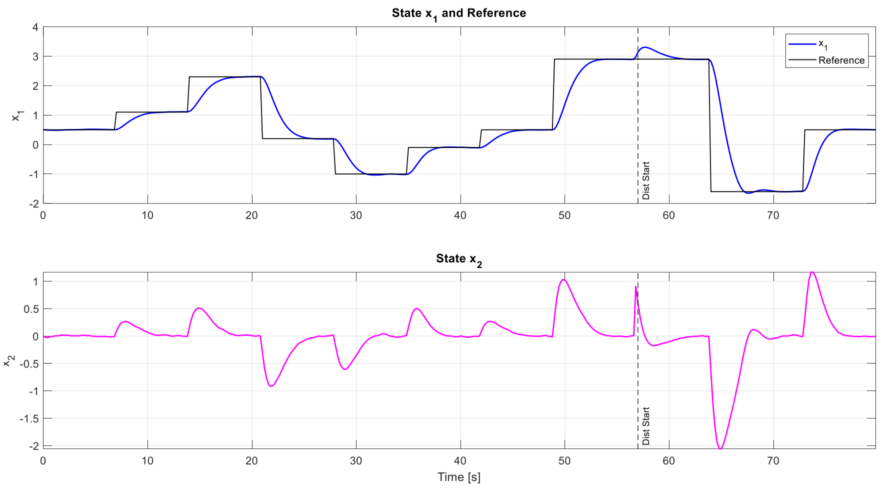

Figure 8 : System States in presence of noise and disturbance

We had considered $\mu = [4,7,10]$, $m = [0]$, $q = [1,1,1]$, $r = 0.5$ and $\psi = [0.3,0.3,0.3]$.