# Dynamic Matrix Controller (DMC)

## Introduction

The DMC controller is used for linear systems. We use a nonlinear system and linearize it at an operaintg point to obtain a linear system.

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. In Section 2, we will provide a summary of the design process of the DMC 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.

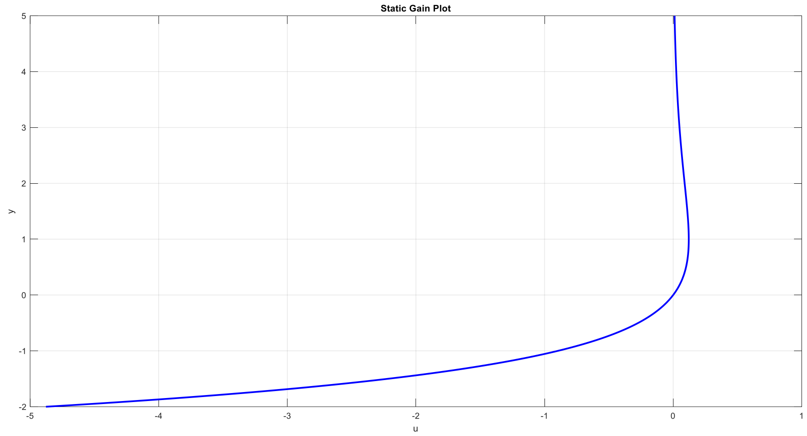


Figure 1: static gain plot for input $u$ and output $x_1$

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.

If we want to control the nonlinear system in the whole operational space, we can use, for example, the EPFC controller which is provided in this same repo. However, here we confine ourselves to the region where the input-output relation is one-to-one and therefore, we consider the point $(u = -0.897, x_1 = -1)$ as our operating point. The chosen operating point can be seen in fig. 2.

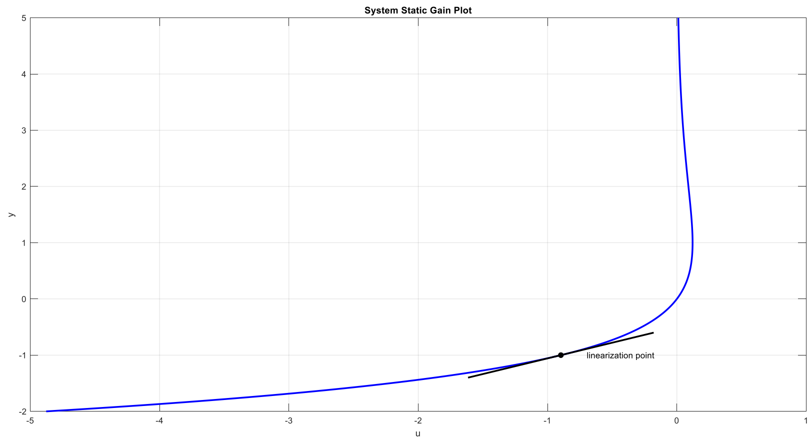


Figure 1: Static gain plot and the selected operating point

The linearized system is as follows:

$$A = \begin{bmatrix} 0 & 1 \\ -1.7941 & -1.1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad C = [1 \quad 0], \qquad D = 0 \tag{4}$$

The linear system has two poles with real values of $-0.55$, which means it has a time constant of about $1.81s$. By sampling $5$ to $10$ times in a time constant, the samping time has to be between $0.18s$ and $0.36s$. We select $0.2s$ as our sampling time. Note that the more correct approach would be to select the sample time for the closed-loop system, not the open-loop system, but this works just fine in this case.

The discrete-time linear system is:

$$A_d = \begin{bmatrix} 0.9668 & 0.1774 \\ -0.03183 & 0.7717 \end{bmatrix}, B_d = \begin{bmatrix} 0.0185 \\ 0.1774 \end{bmatrix}, C_d = [1 \quad 0], D_d = 0 \tag{5}$$

Its step response is plotted in fig. 3.

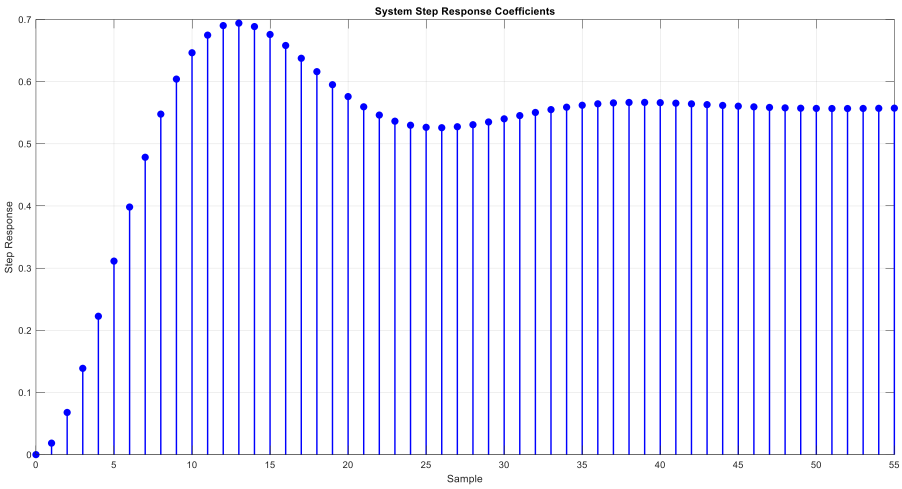


Figure 3: Step response of the Discrete Linearized System

Note that after sample 55, the step response coefficients do not change that much, hence we select $N = 55$ as our model horizon in the next sections.

## Section 2: DMC Controller Design

A model-predictive controller has three features: a predictive model, an optimization problem and a receding horizon. Any controller with these three features is considered a model-predictive controller.

One of the most famous MPC controllers is the Dynamic Matrix Controller or DMC. This controller needs only the step response coefficients of the system. Assume $g_0, g_1, g_2, \ldots$ are the step response coefficients of the system. These coefficients can be obtained by applying a step input to the system and sampling the output (with a reasonable sampling time). The output of the system can be written as

$$y(t+1) = g_0\Delta u(t+1) + g_1\Delta u(t) + g_2\Delta u(t-1) + \cdots \tag{6}$$

For a stable system we can assume that after a sample, for example the $N$th sample, the step response stabilizes, so

$$g_{N+1} = g_{N+2} = \cdots \tag{7}$$

With some simple math, the relation between the future outputs of the system $Y(t+1)$ and its future inputs $\Delta U$ and its past inputs $\overline{\Delta \mathrm{U}}$ can be written as

$$Y(t+1) = G\Delta U + \bar{G}\,\overline{\Delta \mathrm{U}} + g_{N+1}\overline{U}, \tag{8}$$

in which $G$ is a Toeplitz matrix and $\bar{G}$ is a Hankel matrix. Eq. (8) is the predictive model we'll be using.

The cost function is defined as

$$J = \left(Y_d(t+1) - Y_p(t+1)\right)^T Q \left(Y_d(t+1) - Y_p(t+1)\right) + \Delta U^T(t) R \Delta U(t) \tag{9}$$

This equation settles a trade-off between the magnitude of the input's changes and reference tracking. $Y_d(t+1)$ denotes the desired values for the output in the future samples. It is obtained from filtering the reference input $y_{sp}$. The filter is a first-order filter with its pole at $-\alpha$.

$$\begin{cases} y_d(t) = y_{sp}(t) \\ y_d(t+i) = \alpha y_d(t+i-1) + y_{sp}(t+i), \qquad i = 1,2,\ldots,P \end{cases} \tag{10}$$

$P$ is the prediction horizon which indicates how many future samples are considered in the optimization of the output. It is usually between 11 and 22.

$Y_p(t+1)$ is the predicted output. When MPC is used in a closed-loop manner, $Y_p(t+1)$ is calculated using eq. (11) in which $Y_m(t+1)$ is the output that the model predicts for the future samples.

$$Y_p(t+1) = Y_m(t+1) + D(t+1) \tag{11}$$

$D(t+1)$ is the disturbance that we predict for the future samples. Given the fact that noise and disturbance are usually not predictable, $D(t+1)$ is calculated from eq. (14).

$$D(t+1) = d(t)\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}, \qquad d(t) = y_p(t) - y_m(t) \tag{12}$$

In this equation, $y_p(t)$ is the process output at time $t$ which is available and $y_m(t)$ is the output of the model at time $t$ which is also available. Basically, it is assumed that the disturbance will not change and its magnitude is estimated using the prediction error of the predictive model at the current sample.

The control input can be calculated by taking the derivative of eq. (9).

$$\frac{\partial J}{\partial \Delta U} = 0 \Rightarrow \Delta U = (G^T Q G + R)^{-1} G^T Q \bar{E}(t+1) \tag{13}$$

$\bar{E}(t+1)$ in this equation is calculated using

$$\bar{E}(t+1) = Y_d(t+1) - Y_{past}(t+1) - D(t+1) \tag{14}$$

$Y_{past}(t+1)$ is the output of the predictive model without any input; basically the *free response.* So basically the control input can be written as

$$\Delta U = K_{DMC}\bar{E}(t+1), \tag{15}$$

where we have

$$K_{DMC} = (G^T Q G + R)^{-1} G^T Q \tag{16}$$

DMC is a type 1 controller, so it won't have steady-state error to a step input.

Therefore, as we saw, the control input is calculated for the next $M$ samples, but only the first one is applied to the system. This might seem counterintuitive or unproductive to even calculate the extra inputs at the first place if we're going to just ignore them. However, the mere degrees-of-freedom this adds does have an impact on the control input applied to the system.

In the next section, we see the simulation results. Note that the DMC controller is designed (and can only be designed) for the linearized system, but its generated control input is then applied to both the linearized and the original nonlinear system. The further the system is from the operation point, the less accurate the linearized system describes it; therefore, we do expect to see performance degradation in those cases.

## Section 3: Applying the Controller to the System

Now we apply the controller to the system. The tuning parameters are considered to be

$$P = 11, M = 8, Q = I, R = 0.3107I, \alpha = 0.5$$

Noise and disturbance are considered like this, wherever they are used:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -0.33e^{-(x_1+d)}(x_1 + d) - 1.1x_2 + u \\ y = x_1 + n \end{cases} \tag{17}$$

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

The output of the two systems, the control input and the systems' states are shown in figures 4, 5 and 6 respectively.

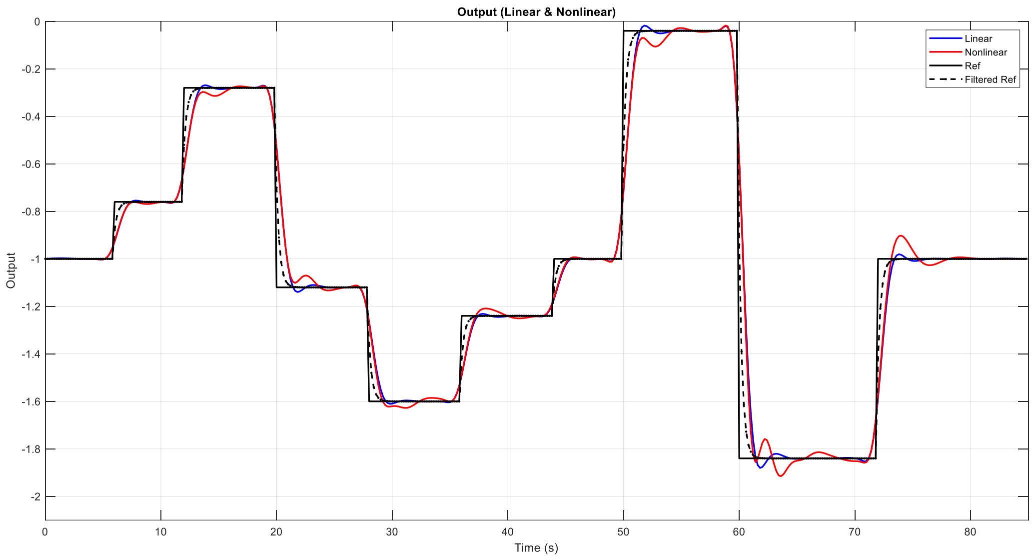


Figure 4: System output

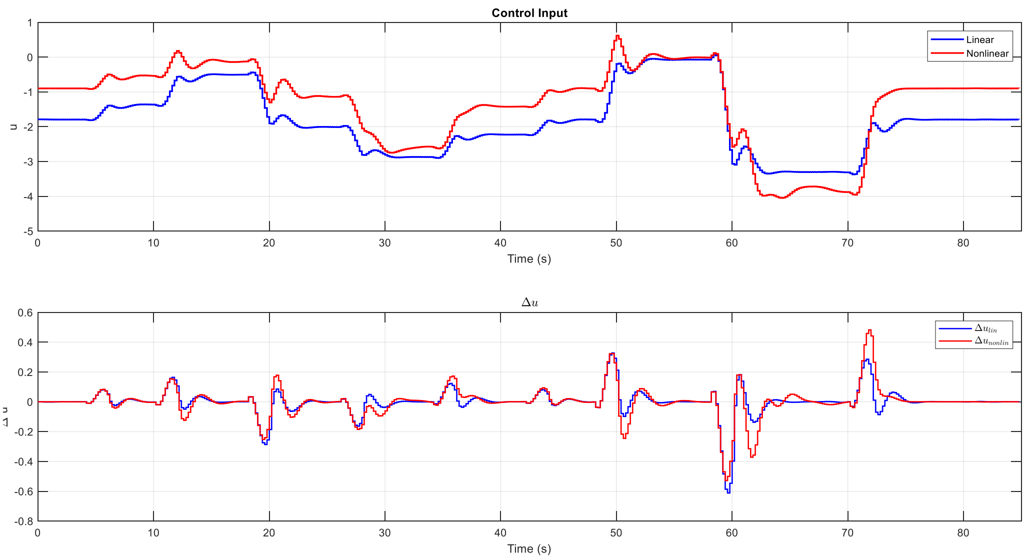


Figure 5: Control Input

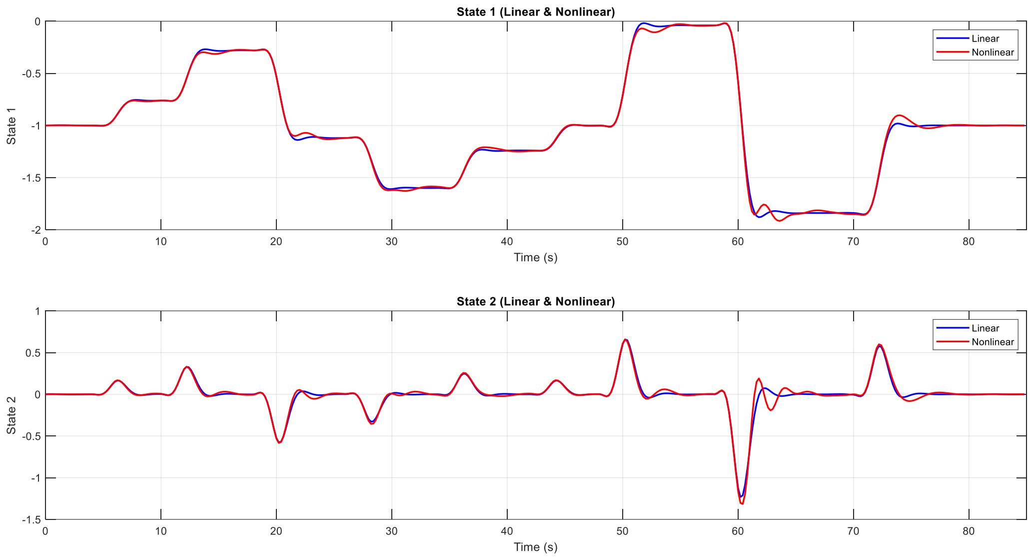


Figure 6: System States