Applications of predictive control

Alexandru Zigler

2025

1 General aspects

Predictive control is an advanced control technique, known and used for its advantages, especially in real-time systems. It significantly reduces the control complexity of systems with a high number of inputs and outputs (*MIMO*). The control technique aims to solve an optimization problem at each time step. Therefore, it is often preferred for implementation on computing systems over other methods. Another important aspect is its ability to satisfy a set of constraints, either *hardware* (physical system limitations) or *software* (imposed). Furthermore, as the name suggests, predictive control has the ability to foresee future events or behaviours and can apply decision algorithms accordingly. It makes use of the controlled plant's dynamic model to calculate the values of future states. The collection of future time steps included in the algorithm is called the prediction horizon. Among all predicted scenarios, the solution to the optimization problem that minimizes the cost function is selected. The corresponding control signal is applied at the current time step. The process repeats as the prediction horizon slides forward to the next time step.

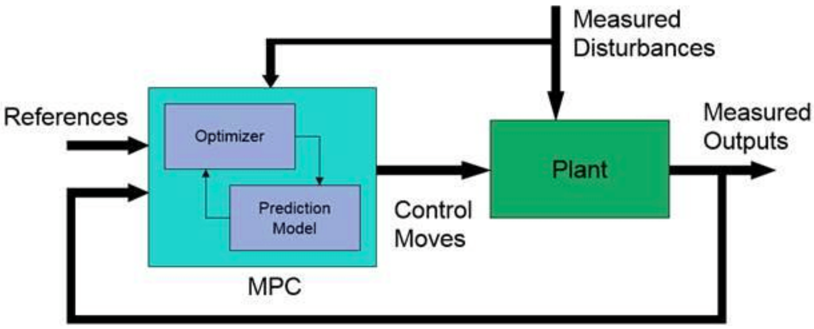


Figure 1: Control architecture

Design parameters

Sampling time

It is essential for system discretization, which depends on the system's time constants. This parameter can strongly influence the discrete system's dynamics and therefore impact control performance (trajectory tracking and disturbance rejection).

Prediction horizon

This parameter represents the number of future time steps considered by the controller at the current time step. The goal of the controller is to determine the system's future states based on the prediction model. If the horizon is too short, there is a risk that the control signal will not react in time and some constraints may not be satisfied. If the horizon is too long, the computational effort increases unnecessarily, especially when disturbances occur. However, it is recommended that the prediction horizon exceed the system's transient state.

Control horizon

The control horizon represents the time duration, in samples, over which the control signal is optimized and planned. Its value can be less than or equal to the prediction horizon. After the control horizon ends, the control signal is held constant until the prediction horizon is reached. This phenomenon reduces computational effort without compromising performance.

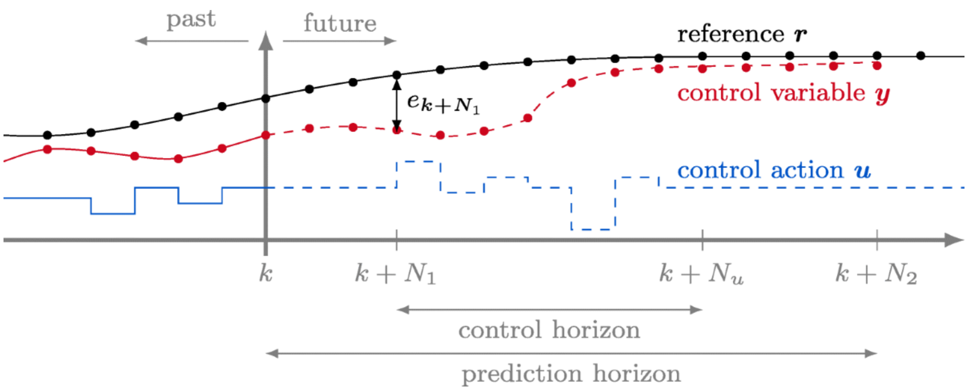


Figure 2: Signal evolution in Predictive Control

Cost function

The cost function defines the optimization problem based on the desired performance of the control system. Examples include minimizing the tracking error or reducing the control effort. The cost function can incorporate weighting factors that prioritize specific objectives. This approach is particularly useful when dealing with complex *MIMO* systems.

2 Case study

Consider a mechanical system model composed of two masses, two nonlinear springs, and two dampers. The mechanical parameters are listed in Table 1. The external forces applied to the masses serve as the inputs, while the outputs are the displacements of the masses.

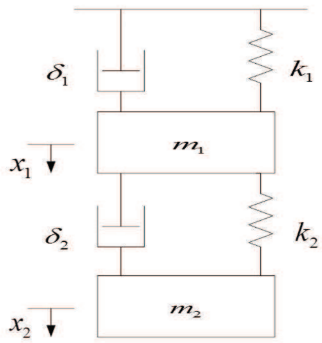


Figure 3: Mechanical system

Table 1: Mechanical parameters

Parameter	Description	Value
$k(N/m)$	elastic constants	$k_1 = \frac{2}{5}, k_2 = 1$
m(Kg)	masses	$m_1 = 1, m_2 = 2$
$\delta(Ns/m)$	damping factors	$\delta_1 = \frac{1}{10}, \delta_2 = \frac{1}{5}$
μ	nonlinear coefficients	$\mu_1 = \frac{1}{6}, \mu_2 = \frac{1}{10}$

The differential equations that describe the system are:

$$\begin{aligned} m_1\ddot{x}_1 &= -\delta_1\dot{x}_1 - k_1x_1 + \mu_1x_1^3 - k_2(x_1 - x_2) + \mu_2(x_1 - x_2)^3 + u_1 \\ m_2\ddot{x}_2 &= -\delta_2\dot{x}_2 - k_2(x_2 - x_1) + \mu_2(x_2 - x_1)^3 + u_2 \end{aligned} \tag{1}$$

Rewrite the model in state-space form:

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= -\frac{\delta_1}{m_1}x_3 - \frac{k_1}{m_1}x_1 + \frac{\mu_1}{m_1}x_1^3 - \frac{k_2}{m_1}(x_1 - x_2) + \frac{\mu_2}{m_1}(x_1 - x_2)^3 + \frac{u_1}{m_1} \\ \dot{x}_4 &= -\frac{\delta_2}{m_2}x_4 - \frac{k_2}{m_2}(x_2 - x_1) + \frac{\mu_2}{m_2}(x_2 - x_1)^3 + \frac{u_2}{m_2} \\ \mathbf{y} &= \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \end{aligned} \tag{2}$$

Let $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, then the Jacobian matrix is:

$$J(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1+k_2}{m_1} + \frac{3\mu_1}{m_1}x_1^2 + \frac{\mu_2}{m_1}(3x_1^2 - 6x_1x_2 + 3x_2^2) & \frac{k_2}{m_1} + \frac{\mu_2}{m_1}(-3x_1^2 + 6x_1x_2 - 3x_2^2) & -\frac{\delta_1}{m_1} & 0 \\ \frac{k_2}{m_2} + \frac{\mu_2}{m_2}(-3x_2^2 + 6x_1x_2 - 3x_1^2) & -\frac{k_2}{m_2} + \frac{\mu_2}{m_2}(3x_2^2 - 6x_1x_2 + 3x_1^2) & 0 & -\frac{\delta_2}{m_2} \end{bmatrix}$$

Consider the operating point $\boldsymbol{x}_0 = \mathbf{0}$, around which we want to linearize the system. Thus, we obtain the new steady state:

$$A = J(\boldsymbol{x})|_{\boldsymbol{x}=\mathbf{0}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1+k_2}{m_1} + & \frac{k_2}{m_1} & -\frac{\delta_1}{m_1} & 0 \\ \frac{k_2}{m_2} & -\frac{k_2}{m_2} + & 0 & -\frac{\delta_2}{m_2} \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The previous linear system will be discretized. Based on the discrete model, we will design predictive control laws and present their particularities.

$$A_d = I + T_sA, \qquad B_d = T_sB, \qquad C_d = C, \qquad D_d = D.$$

2.1 Basic predictive control

We start from the discrete linear model, generalized to the*MIMO* case.

$$
\begin{aligned}
&x(k+1) = A_d x(k) + B_d u(k) \\
&y(k) = C_d x(k), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m, y \in \mathbb{R}^p.
\end{aligned} \tag{3}
$$

We use the notations $\Delta x(k+1) := x(k+1) - x(k)$ and $\Delta y(k+1) := y(k+1) - y(k)$ for constructing the augmented model.

$$
\begin{aligned}
\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} A_d & 0_{n\times p} \\ C_d A_d & I_p \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix} \Delta u(k) \\
y(k) &= \begin{bmatrix} 0_{p\times n} & I_p \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}
\end{aligned} \tag{4}
$$

Rewrite system (4) compactly:

$$
\begin{aligned}
&x_a(k+1) = \Phi_a x_a(k) + \Gamma_a \Delta u(k) \\
&y(k) = C_a x_a(k), \quad x_a \in \mathbb{R}^{n+p}, \quad \Gamma_a \in \mathbb{R}^{(n+p)\times m}, \quad C_a \in \mathbb{R}^{p\times(n+p)}.
\end{aligned} \tag{5}
$$

Let N_p and N_c be the prediction and control horizons, respectively. We define the desired trajectory of the control signal as the sequence of solutions to the optimization problem at each time step:

$$
\Delta U = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_c-1) \end{bmatrix}^T .
$$

As we have the control signals for future time steps, we can write the next N_p states, predicted at the current time step k:

$$
\begin{aligned}
x_a(k+1|k) &= \Phi_a x_a(k) + \Gamma_a \Delta u(k) \\
x_a(k+2|k) &= \Phi_a^2 x_a(k) + \Phi_a \Gamma_a \Delta u(k) + \Gamma_a \Delta u(k+1) \\
\dots &= \dots \\
x_a(k+N_p|k) &= \Phi_a^{N_p} x_a(k) + \Phi_a^{N_p-1} \Gamma_a \Delta u(k) + \dots + \Phi_a^{N_p-N_c} \Gamma_a \Delta u(k+N_c-1).
\end{aligned} \tag{6}
$$

We perform a left multiplication of equation (6) with C_a in order to obtain the predicted output vector.

$$
\begin{bmatrix} C_a x_a(k+1|k)) \\ C_a x_a(k+2|k) \\ \vdots \\ C_a x_a(k+N_p|k) \end{bmatrix} = \begin{bmatrix} C_a \Phi_a \\ C_a \Phi_a^2 \\ \vdots \\ C_a \Phi_a^{Np} \end{bmatrix} x_a(k) + \begin{bmatrix} C_a \Gamma_a & 0 & \dots & 0 \\ C_a \Phi_a \Gamma_a & C_a \Gamma_a & \dots & 0 \\ \vdots & \vdots & \ddots & \\ C_a \Phi_a^{N_p-1} \Gamma_a & C_a \Phi_a^{N_p-2} \Gamma_a & \dots & C_a \Phi_a^{N_p-N_c} \Gamma_a \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}.
$$

Rewrite compactly:

$$
Y = \begin{bmatrix} y(k+1|k) & y(k+2|k) & \dots & y(k+Np|k) \end{bmatrix}^T = W x_a(k) + Z \Delta U. \tag{7}
$$

Consider the cost function:

$$J(\Delta U) = \frac{1}{2}(r - Y)^T Q (r - Y) + \frac{1}{2}\Delta U^T R \Delta U. \tag{8}$$

The symmetric positive definite matrices Q and R represent the weighting factors for tracking the reference signal r and for minimizing the control effort, respectively. As a note, $r - Y$ denotes the vector

$$\begin{bmatrix} r - y(k+1|k) & r - y(k+2|k) & \dots & r - y(k+Np|k) \end{bmatrix}^T.$$

Substitute (7) in (8).

$$J(\Delta U) = \frac{1}{2}(r - Wx_a(k) - Z\Delta U)^T Q (r - Wx_a(k) - Z\Delta U) + \frac{1}{2}\Delta U^T R \Delta U$$

$$J(\Delta U) = \frac{1}{2}\left[r - Wx_a(k)\right]^T Q \left[r - Wx_a(k)\right] - \left[r - Wx_a(k)\right]^T QZ\Delta U + \frac{1}{2}\Delta U^T \left(Z^T Q Z + R\right)\Delta U. \tag{9}$$

$$\frac{\partial J}{\partial \Delta U} = 0 \implies -Z^T Q (r - Wx_a(k)) + (Z^T Q Z + R)\Delta U = 0 \tag{10}$$

Thus, the solution of the optimization problem is $\Delta U^* = (R + Z^T Q Z)^{-1} Z^T Q (r - Wx_a)$. Further, we take $\Delta u(k)$ and calculate $u(k)$.

Results

For simulations, the following parameters have been chosen: $T_s = 0.1$, $N_p = 10$, $N_c = 10$, $R = 0.1 I_{20}$, $Q = I_{20}$, $r(t) = \begin{bmatrix} 1 & -2 \end{bmatrix}^T$.

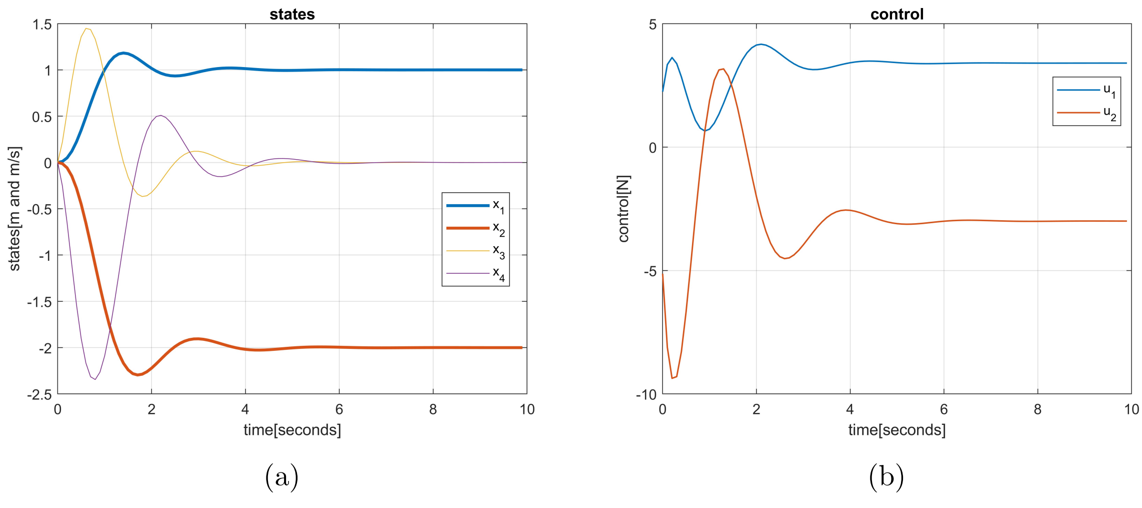
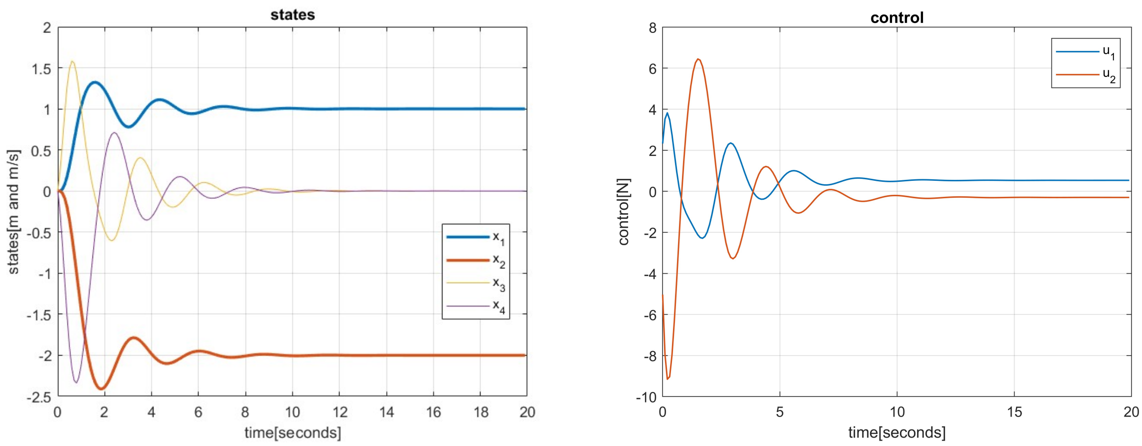


Figure 4: Time evolution of states and control signals for the linearized system

We can see that the linear predictive control technique yields good performance for the linearized system (Figure 4). The steady state error is zero, while the settling time is approximately 4 seconds for both outputs. The control signals have reasonable amplitudes.

Another observation is that decreasing the sampling time brings the system closer to instability. Also, increasing the prediction horizon improves performance, for example the overshoot is eliminated and the oscillations of the control signal are significantly reduced.



(a) (b)

Figure 5: Time evolution of states and control signals for the nonlinear system

Further, we take the control law that was designed based on the linearized system and validate it on the nonlinear system (Figure 5). Performance deteriorates compared to the previous case, but the trajectory tracking is still performed with zero steady state error. Oscillatory behaviours appear in the time evolution of states and control signals, but they can be diminished using a larger prediction horizon ($N_p = 50$). However, the selected reference signal $r(t) = \begin{bmatrix} 1 & -2 \end{bmatrix}^T$ is far from the operating point (i.e. $x_0 = 0$). When the reference signal is chosen far enough, the system becomes unstable.

2.2 Predictive control with constraints

Let us consider a set of constraints over the control signal:

$$u_{min} \leq U(k) \leq u_{max}, \text{ or, equivalently, } \begin{bmatrix} -U(k) \\ U(k) \end{bmatrix} \leq \begin{bmatrix} -u_{min} \\ u_{max} \end{bmatrix}.$$

The following equality is satisfied:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} u(k-1) + \begin{bmatrix} I_m & 0 & \dots & 0 \\ I_m & I_m & \dots & 0 \\ \vdots & \vdots & \ddots & \\ I_m & I_m & \dots & I_m \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix},$$

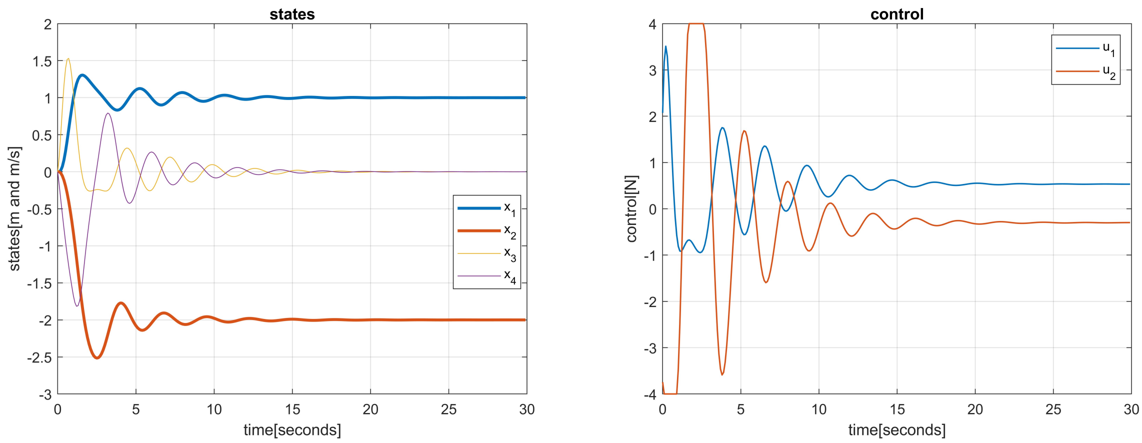
or, written compactly, $U(k) = Eu(k-1) + H\Delta U(k)$.

Designing a predictive control law with constraints is similar to the approach presented above. The only difference is the optimization problem, that will be subjected to the imposed constraints.

From the two equations written above, we can mathematically describe the defined constraints.

$$\begin{bmatrix} -H \\ H \end{bmatrix} \Delta U(k) \leq \begin{bmatrix} -u_{min} + Eu(k-1) \\ u_{max} - Eu(k-1) \end{bmatrix} \tag{11}$$

These constraints will be incorporated in the cost function (8). The resulting optimization problem is known as quadratic programming. Among the quadratic programming techniques, we can mention Karush-Kuhn-Tucker conditions, Active-set method, Hildreth's algorithm. Here, for solving the optimization problem, we used the *quadprog* function in MATLAB.



(a) (b)

Figure 6: Time evolution of states and control signals for the nonlinear system, with constraints over control signals

The following constraints were imposed: $-4 \leq u_1, u_2 \leq 4$ (Figure 6). They are globally satisfied in the control signal's time evolution. Satisfying performances from a trajectory tracking point of view are yet again met.

A similar process would be defining constraints for the outputs of the system.

3 Conclusions

To conclude, predictive control is highly useful among advanced control techniques. Its ability to foretell the states of the system in certain scenarios has countless applications and benefits. Another important aspect is the flexibility in designing the control law using constraints, which are essential for ensuring safe and reliable operation. Regarding the design complexity, the main inconvenience is solving the optimization problem, which obviously depends on the intricacies of the selected process model. In the linear case, the problem can be easily solved analytically. This is where different branches of predictive and optimal control come into play. Linearizing around a set of operating points and adapting the predictive control accordingly is an approach that may introduce numerical instability. On the other hand, the nonlinear predictive control uses a nonlinear prediction model, therefore the complexity increases. Some examples are nonlinear identification or modeling and optimizing the nonlinear cost function, whose convexity must be analyzed. From an implementation perspective, predictive control is computationally demanding. Specialized computing equipment is currently being developed, as well as automatic code generation applications. Another type of predictive control is explicit MPC, which solves the optimization problem offline.

Bibliography

- J. M. Maciejowski, "Predictive Control with Constraints," Prentice Hall, Upper Saddle River, 2002.
- Liuping Wang. 2009. Model Predictive Control System Design and Implementation Using MATLAB (1st. ed.). Springer Publishing Company, Incorporated.
- Bo Bernhardsson, Karl Johan Åström, Department of Automatic Control LTH, "Model Predictive Control" Lecture Lund University
- https://www.mathworks.com/videos/series/understanding-model-predictive-control.html