PROCESS AUTOMATION

A.Y. 2020/2021

MODEL PREDICTIVE CONTROL SUMMARY

Rossella Bonuomo

Veronica Vulcano

MODEL PREDICTIVE CONTROL

Model predictive control (MPC) is a class of process control methods whose aim is to control a process while satisfying a set of constraints; in fact, we can check the constraint violations while we are controlling the system.

It is based on an iterative, finite horizon optimization of a plant model; in fact, there is an explicitly use of a model that is used to compute the predicted output on n future time instant (finite prediction horizon). Basing on the error between the predicted output and a reference trajectory, it computes the optimal sequence of control actions with respect to an objective function to minimize. Once we have the optimal control sequence in the future, we are not going to apply all the sequence but only the first computed control action. So, if at time k we are computing the control action $u_k, u_{k+1}, \dots, u_{n-1}$, then we are going to apply only u_k to the actual process. At time $k+1$ we are going to do everything from scratch, so to compute another control sequence, because something might have been changed. This is the **receding strategy.**

The main advantages are that the tuning is easier with respect to PID controller, it can deal with many processes since it can handle time delays systems, non-minimum phase systems and also unstable one. Moreover, it can be used also for MIMO systems and it is able to compensate for disturbances. Finally, constraints are included in the process design. The drawbacks are that the derivation is complex with respect to the PID case and we need an accurate model of the process; the accuracy, in fact, depends on the knowledge we have of the model.

The MPC structure contains two main blocks: the **process model** and the **optimizer**. The process model is fed with the control action and produces the predicted output. There are many types of process model we can consider (transfer function, step and impulse response). The optimizer considers the error (difference between the reference trajectory and the predicted output) and computes an optimization problem of a cost function taking into account some constraints on the input, output and state variable. The idea is this one: we have a sequence of control actions in the future; then, since we have the model of the process, we can estimate the output over that horizon so we can compute the error between this estimated output and the set point.

Given the sequence:

$$\{u(t+k)\}_{k=0,\dots,N-1}$$

We can compute the sequence of predicted outputs:

$$\{\hat{y}(t+k)\}_{k=1,\dots,N-1}$$

If we know that we have to follow a set point of if we know the trajectory in the future, we have a sequence of errors:

$$\{e(t+k)\} = \begin{Bmatrix} w(t+1) - \hat{y}(t+1) \\ w(t+2) - \hat{y}(t+2) \\ \vdots \end{Bmatrix}$$

Then we use the error to evaluate the control actions. In fact, we have to consider the best control actions taking into account a cost function computed thanks to the error. The optimal control action is:

$$u^*(t+k|t), \qquad k = 0, \dots, N-1$$

But since we are using the receding strategy, we will feed the system only with $u^*(t|t)$. Then at time t+1 we will compute a new control action:

$$u^*(t+1+k|t+1), \qquad k = 0, \dots, N-1$$

And we will feed the system with the first one $u^*(t+1|t+1)$. Generally, $u^*(t+1|t+1) \neq u^*(t+1|t)$ because during this interval something might happen.

PREDICTION MODEL

The objective of the prediction model is the prediction of future samples of the output over the prediction horizon N:

$$\{\hat{y}(t+k|t)\}_{k=1,\dots,N}$$

To develop a model, we need to address a tradeoff between complexity and representativeness of the model; it should be accurate but should also be simple enough to do analysis on it.

Impulse response model

We feed the process with an impulse input and we measure the output.

$$u(t) = \delta(t) = \begin{cases} 1, & if\ t = 0 \\ 0, & otherwise \end{cases}$$

$$y(t) = \sum_{i=1,\dots,\infty} h_i u(t-i)\,, \qquad t = 0,1,2,\dots$$

If we assume that the process is stable without integrators, the output will annihilate with time (we assume after N samples). We approximate the output to the sum of N elements:

$$y(t) \cong \sum_{i=1,\dots,N} h_i u(t-i)\,, \qquad t = 0,1,2,\dots$$

If we use the z transformation, we can write it:

$$y(t) \cong \sum_{i=1,\dots,N} h_i u(t-i) \coloneqq H(z^{-1})u(t)$$

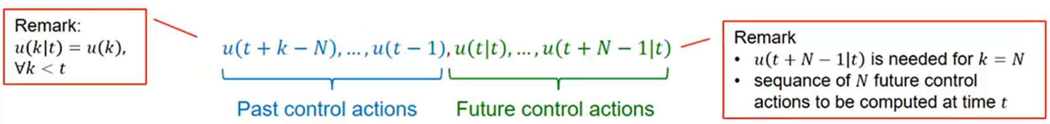
With $H(z^{-1}) = h_1 z^{-1} + h_2 z^{-2} + \cdots + h_n z^{-N}$ impulse response of the process.

$$y_\delta(k) = H(z^{-1})\delta(k) \xrightarrow{z} y_\delta(z) = H(z^{-1})\underbrace{\mathcal{Z}\{\delta(k)\}}_{1} = H(z^{-1})$$

The prediction of the output that the model will compute is:

$$\hat{y}(t+k|t) = \sum_{i=1,\dots,N} h_i u(t+k-i|t)$$

This is made of future and past samples.



For $i = 1, \dots, k$ $[(k-i) > 0]$ we have the unknown future inputs, so we have **future control actions** that we need to compute the predicted output. When $i = k+1, \dots, N$ $[(k-i) < 0]$, we have the **past control actions**.

With this model, time delay and nonminimum phase process are considered straightforwardly; if we have a delay, we will see that the first samples h_i are zeros while if we have non minimum phase terms, they will be negative (but we don't have to treat them in a special way).

Step response model

We feed the process with a step input and we measure the step response. With the impulse response, if the system is stable and it has no integrators, we see that the output annihilated. If we feed the process with a step, it will not annihilate but it will reach a steady state value. So the samples are significant up to the sample time where each sample is practically equal to the next one. Therefore, we consider the difference between two consecutive samples, Δu (that is going to annihilate after N sample)

$$y(k) \cong \sum_{i=1,\dots,N} g_i \Delta u(k-i) \coloneqq G(z^{-1})\Delta u(k), \qquad k = 0,1,2,\dots$$

With $G(z^{-1}) = g_1 z^{-1} + g_2 z^{-1} + \dots + g_N z^{-1}$ step response of the process.

$$y_{u_{-1}}(k) = G(z^{-1})\underbrace{\left(u_{-1}(k) - u_{-1}(k-1)\right)}_{\Delta u} \xrightarrow{z} y_{u_{-1}}(z) = G(z^{-1})(1-z^{-1})\mathcal{Z}\{u_{-1}(k)\} = G(z^{-1})$$

The prediction of the output that the model will compute is:

$$\hat{y}(t+k|t) = \sum_{i=1,\dots,N} g_i \Delta u(t+k-i|t) = G(z^{-1})\Delta u(t+k|t)$$

The step response model and the impulse are equivalent, we can switch from one to the other in this way:

$$\delta(t) = u_{-1}(t) - u_{-1}(t-1)$$

So also the output are computed in this way:

$$\begin{gathered} h_0 = g_0 - g_{-1} \\ h_1 = g_1 - g_0 \\ h_2 = g_2 - g_1 \\ h_i = g_i - g_{i-1} \end{gathered}$$

And

$$g_i = \sum_{j=0,\dots,i} h_j$$

Transfer function model

Suppose that the transfer function of a process is $P = \frac{B}{A}$, then we have $A(z^{-1})y(t) = B(z^{-1})u(t-1)$. So, the predicted output will be:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})}u(t+k-1|t)$$

In this case we can deal with unstable process and process with integrator; therefore, it is a more general process.

State space model

We might also use a state space model both with the **implicit representation** (input-state-output):

$$\begin{cases} x(t) = Ax(t-1) + Bu(t-1) \\ y(t) = Cx(t) \end{cases}$$

Or we may have **explicit representation**:

$$y(t+1) = Cx(t+1) = C\left(Ax(t) + Bu(t)\right)$$

The predicted output is:

$$\hat{y}(t+k|t) = C\hat{x}(t+k|t) = C\left(A^k x(t) + \sum_{i=1,\dots,k} A^{i-1}Bu(t+k-i|t)\right)$$

This is much more complex but can be extended to MIMO process, nonlinear process.

Disturbance model

Generally, the output depends on the input but also on the disturbance. Therefore, we will consider also a model of the disturbance. We will consider it constant over the prediction horizon. For example, we can measure the noise in the past and based on the last measure $n(t)$ we assume that $n(t)$ will remain constant over the prediction horizon.

The most common model for the disturbance in model predictive control is the **autoregressive model**:

$$n(t) = \frac{C(z^{-1})}{D(z^{-1})} e(t)$$

With C and D that are polynomial. C is used to model the colored noise. There are also different kinds of noise depending on the power density per unit of bandwidth of the power spectrum. The power density per unit of bandwidth is:

$$\frac{1}{f^\beta}$$

$\beta = 0 \rightarrow$ white noise;
$\beta = 1 \rightarrow$ pink noise;
$\beta = 2 \rightarrow$ brown noise;
$\beta = -1 \rightarrow$ blue noise.

FREE AND FORCED RESPONSES

The **free response** is the response of the system when no input applies, the **forced response** is the is the response that depends on the input. The control action is the sum of the free u_f and forced u_c response.

$$u = u_f + u_c$$

Let's see which are this value in the future. The free control action:

$$u_f(t+k) = \begin{cases} u(t+k) & \forall k < 0 \\ u(t-1) & \forall k \geq 0 \end{cases}$$

So in the future it is equal to the last applied control action.
The forced response is:

$$u_c(t+k) = \begin{cases} 0 \quad \forall k < 0 \\ u(t+k|k) - u(t-1), \forall k \geq 0 \end{cases}$$

So in the future, it is $u_c(t+k) = u(t+j) - u_f(t+j)$.

Also the output can be computed as the sum of the free and forced response:

$$\hat{y}(t+k|t) = \hat{y}_f(t+k|t) + \hat{y}_c(t+k|t)$$

- The **free response** is the output prediction for $u = u_f$; it is the evolution of the process from the present state (it does not depend on the different sequence of control actions that we might apply).
- The **forced response** $\hat{y}_c$ is the output prediction for $u = u_c$; it is the evolution of the process due to the future control actions (prediction due to future control action variations).

OBJECTIVE FUNCTION

To find the best control action, we solve an optimization problem in which we have to minimize a cost function. This function is usually expressed as:

$$J(N_1, N_2, N_i) = \sum_{j=N_1}^{N_2} \delta(j) \underbrace{\left(\hat{y}(t+k|t) - w(t+j|t)\right)}_{predicted\ future\ errors}^2 + \sum_{j=1}^{N_u} \lambda(j)\left(\Delta u(t+j-1)\right)^2$$

δ and λ are weighing terms that depend on the sample; for instance, future predicted errors are more important than the one in the beginning. Moreover, we may want to limit the control action, so we need δ.

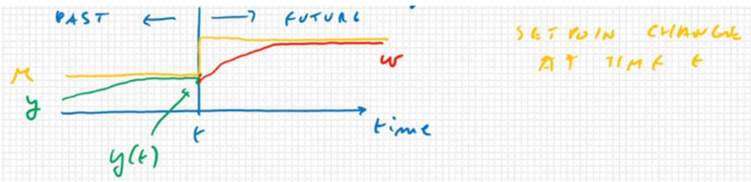
N_2 is the **predicted horizon**. Generally, we start to evaluate the error just after a while, that is why the summation starts from N_1. N_u is the **control horizon.** Usually we set $N_u \leq N_2$. We want this for two reasons: one because there are too many unknowns, so we can cut the number of unknowns; the other reason is that after some samples in the signal Δu we might see undesirable high frequency control changes.

When we have a delay, we have that the response of the model starts a bit late; for this reason, we tune the parameter N_1 in a way that we avoid the first samples that are zero.

When we have a non-minimum phase term, the output might have an inversion in the beginning. Since we are going towards the set point, we are not interested in this inversion so also in this case we set N_1 in a way to avoid the first negative samples.

In the objective function, the error is the difference between the predicted output and the reference trajectory; this may be known or not in the future.

Sometimes the reference trajectory does not coincide with the reference set point.



In this case, there is an abrupt change in the reference set point (r) so we may want to approach it in a smoother way. In this case, we define the reference trajectory w that will converge to r but since we are following a smoother trajectory, we probably will have a smoother output response. In this case, we define the reference trajectory in the future like this:

$$w(t+k|t) = \begin{cases} y(t) & if\ k = 0 \\ \alpha w(t+k|t) + (1-\alpha) r(t+k) & if\ k > 0 \end{cases}$$

$\alpha \in (0,1)$, so we have a convex combination of w and r. w starts from y and it asymptotically reaches r; α tells us how fast this asymptotic behavior is. For example, if α is close to 1, the rise time is lower.

OPTIMIZATION PROBLEM

We have defined J, since it is a cost function, we want to minimize it over all the possible sequence of control actions:

$$\min_{\underline{u}} J\big(u(t|t), u(t+1|t), \dots, u(t+N_2-1|t)\big)$$

This is an **unconstrained optimization problem**. Where $\underline{u} = [u(t|t), u(t+1|t), \dots, u(t+N_2-1|t)]^T$. This is the vector collecting the unknowns of our problem. We have N_2 unknowns for the problem.

We can solve this problem in close form if we have a linear system and a quadratic cost function.

Sometimes, we also have constraints; in this case we have a **constrained optimization problem:**

$$\min_{\underline{u}} J(\underline{u})$$
$$s.t. u_{min} \leq \; u(t+j|t) \leq u_{max}, \qquad j = 0, \dots, N_2 - 1$$

Or we might have constraints for the variation of u, so we have $N_2 - 1$ constraints of the kind:

$$\Delta_{min} \leq u(t+j+1|t) - u(t+j|t) \leq \Delta_{max}, \qquad j = 0, \dots, N_2 - 2$$

We might have constraints on the output, so we might have additional constraints of the form:

$$y_{min} \leq y(t+j|t) \leq y_{max}, \qquad j = 0, \dots, N_2 - 2$$

One way to reduce the number of unknowns is to set $N_u < N_2$. We just put $\Delta u = 0$ after N_u samples and this is the control horizon concept.

$$u(t+j-1|t) = u(t+N_u-1|t), \qquad N_u + 1 \leq j \leq N_2$$

This means that:

$$\Delta u(t+j-1|t) = 0, \qquad N_u + 1 \leq j \leq N_2$$

We can reduce even more the number of unknowns of the problem by reducing the kinds of input that we can give to the process. The easiest way to do it is to represent the control signal $u(t+k|t)$ just as a linear combination of functions. For example, we might have a linear combination of n functions $B_i(k)$.

$$u(t+k|t) = \sum_{i=1}^{n} \mu_i(t) B_i(k)\,, \qquad k = 0,1, \dots, N_u - 1$$

$B_i(k)$ is the **base function**. We just need to compute a linear combination of those, so we have some parameters $\mu_i(t)$. Of course, it is not true that we can represent every signal with this kind of structure; we would need infinite base functions to represent all the possible signals, but maybe we can be able to select a set of a base functions which are enough for representing the control signals that we need.

Now the unknowns of the problem are the coefficient μ_i; therefore, we just have to find n value for this coefficient while before we had to consider N_u unknown (but $n < N_u$).

COMMERCIAL MPC SCHEMA

We will see different algorithm to solve the MPC.

Dynamic matrix control

This is the step response model, so we need to have stable process without integrator. Let's see how we can obtain the predicted output. First, we consider the disturbance constant in the prediction:

$$\hat{n}(t+k|t) = \hat{n}(t|t), \qquad k = 1,..p \quad (1)$$

$$\hat{n}(t|t) = y_m(t) - y(t)$$

The step response model is:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t-i) \quad (2)$$

The model is infinite dimensional but since we have assumed that there is no integrator, we can stop the summation after n samples. The predicted output is:

$$\hat{y}(t+k|t) = y(t+k) + \hat{n}(t+k|t)$$

$$\hat{y}(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i|t) + \hat{n}\,(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) \; + \hat{n}(t|t)$$

Now we use the free and forced response concept. In fact, in the summation, there are past and future control actions. Up to $i = k$, we have future control input. The other terms from $i = k+1$ to infinity, are past control actions and they are known.

$$\hat{y}(t+k|t) = \underbrace{\sum_{i=1}^{k} g_i \Delta u(t+k-i|t)}_{unknown} + \underbrace{\sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i)}_{known} + \hat{n}(t|t)$$

Since $\hat{n}(t|t) = y_m(t) - y(t)$

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i|t) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - y(t)$$

We replace the expression of $y(t)$ from (2) (known term)

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i|t) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i)$$

And we focus on the second term:

$$\sum_{i=k+1}^{\infty} g_i \Delta \mathrm{u}(t+k-i) = \sum_{i=1}^{\infty} g_{i+k} \Delta u(t-i)$$

So, we can match it with the fourth term:

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i|t) + y_m(t) + \sum_{i=1}^{\infty} (g_{i+k} - \mathrm{g_i}) \Delta u(t-i)$$

The first term is the **forced response**, the second and the third terms are the **free response.** The free response only depends on known and past samples, the force response depends on future actions.

We use the assumptions that the process is asymptotically stable and there are no integrators in the process (so after a while the step response stabilizes to a given value) which means that after N samples, the coefficient is always the same:

$$g_{k+1} \cong g_i \quad i > N$$

So, we can approximate the free response using a finite number of elements.

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i|t) + f(t+k)$$

With

$$f(t+k) = y_m(t) + \sum_{i=1}^{N} (g_{i+k} - \mathrm{g_i}) \Delta u(t-i)$$

We can also handle the case of control horizon which is smaller than the prediction horizon; the control horizon is m and the predicted horizon is p. In this case, in the summation representing the forced response, some terms are null. The obtained model would be:

$$\hat{y}(t+k|t) \cong \sum_{i=k-m+1}^{k} g_i \Delta u(t+k-i|t) + f(t+k)$$

Let’s focus on the way we can solve this problem. We define the matrix of the coefficient as

$$\boldsymbol{G} := \begin{pmatrix} g_1 & 0 & 0 & \dots & 0 \\ g_2 & g_1 & 0 & \dots & 0 \\ g_3 & g_2 & g_1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ g_m & g_{m-1} & g_{m-2} & \dots & g_1 \\ g_{m+1} & g_m & g_{m-1} & \dots & g_2 \\ \dots & \dots & \dots & \dots & \dots \\ g_p & g_{p-1} & g_{p-2} & \dots & g_{p-m+1} \end{pmatrix} \in \mathbb{R}^{p\times m}$$

In the $m-th$ line, we have that all Δu are multiplied by a non-zero element of g, then, due to the fact that $m < p$, the $p-m$ rows are equal to the m rows.

We also introduce the state vector:

$$\boldsymbol{x}(t) := \underbrace{(y_m(t) \quad \Delta u(t-1) \quad \Delta u(t-2) \quad \dots \quad \Delta u(t-N+1))^T}_{\text{known terms}} \in \mathbb{R}^{(N+1)\times 1}$$

That is made by known terms that are the measured output and past control actions.

Therefore, we can write everything in compact form:

$$\hat{y} = Gu + Fx$$

With

$$\boldsymbol{F} := \begin{pmatrix} 1 & g_2 - g_1 & g_3 - g_2 & \dots & g_{1+N} - g_N \\ 1 & g_3 - g_1 & g_4 - g_2 & \dots & g_{2+N} - g_N \\ 1 & g_4 - g_1 & g_5 - g_2 & \dots & g_{3+N} - g_N \\ \dots & \dots & \dots & \dots & \dots \\ 1 & g_{p+1} - g_1 & g_{p+2} - g_2 & \dots & g_{p+N} - g_N \end{pmatrix} \in \mathbb{R}^{p\times(N+1)}$$

If we are able to measure the disturbance, and we assume that it can be modelled though the step response model:

$$D(z^{-1}) = d_1 z^{-1} + d_2 z^{-2} + .. + d_{N_d} z^{-N_d}$$

We obtain a new model:

$$\hat{y} = Gu + f + Dd + f_d$$

With

$$\boldsymbol{D} := \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ d_2 & d_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ d_p & d_{p-1} & d_{p-2} & \dots & d_{p-m+1} \end{pmatrix} \in \mathbb{R}^{p \times m}$$

f_d is the part of the output that depends on past disturbances:

$$f_d := \left(f_d(t+k)\right)_{k=1\dots p} \in R^{mx1}$$

And d is the vector of disturbance increments along the control horizon that is known for assumption:

$$d := \left(\Delta d(t+k|t)\right)_{k=1\dots m} \in R^{px1}$$

Therefore, both D_d, f_d are known thus they are part of the free response. We define a new free response function as $f' = f + Dd + f_d$.

Let's see how we can obtain the closed form solution in this case.

we want to solve:

$$\min_u J(u)$$

With $J(u) = e^T e,\ \ e = \hat{y} - w,\ \ \hat{y} = Gu + f$. These are all vector quantities.

the optimal solution of the optimization problem is:

$$u^* = (G^T G)^{-1} G^T (w - f)$$

We can also compute the optimal cost:

$$J^* = J(u^*)$$

This problem has a limitation; to obtain a non-singular matrix for G, we have to chose the prediction horizon $p \geq m + d$, where m is the control horizon and d is the disturbance.

What if we have a complete cost function?

$$\min_u \hat{e}^T \hat{e} + \lambda u^T u$$

In this case we use a non-zero value for λ weighting the square control action.

$$J(u) = (\widehat{w} - \hat{y})^T (\widehat{w} - \hat{y}) + \lambda u^T u$$

When we took the gradient, there are some terms related to λ, but we can still find the optimal control sequence in closed form:

$$u^* = (G^T G + \lambda I)^{-1} G^T (\widehat{w} - f)$$

In case we have constraints, we cannot solve the problem in closed form. Let's look at the shape of the constraints. We can design the constraint by just giving some value to the coefficient c_i, c_u, c_j.

$$\sum_{i=1}^{N}\left(c_y^{i,j}\hat{y}(t+i|t)+c_u^{i,j}u(t+i-1|t)+c_j\right)\le 0$$

Note: in the exercise $u^* = \Delta u(t|t)$, and $u(t) = u(t-1) + \Delta u(t|t)$.

Model algorithm control

it is similar to the DMC, but it uses the impulse response model and can be used only if $m = p$ so it cannot handle the delays. Also this algorithm, can be applied only to stable process without integrators.

We assume that the disturbance is constant:

$$\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t), \quad k = 1 \dots m$$

And the model is the impulse response one:

$$\hat{y}(t|t) = \sum_{i=1}^{N} h_i u(t-i)$$

So the predicted output is:

$$\hat{y}(t+k|t) = \sum_{i=1}^{N} h_i u(t+k-i|t) + \hat{n}(t+k|t)$$

The predicted output is the sum of the free and forces response:

$$\hat{y}(t+k|t) = f_c(t+k) + f_f(t+k) + \hat{n}(t|t), k = 1, \dots, m$$

- $f_c(t+k) = \sum_{i=1}^{k} h_i u(t+k-i|t)$ ← forced response
- $f_f(t+k) = \sum_{i=k+1}^{N} h_i u(t+k-i)$ ← free response

And also in this case future and past control actions appear. For instance, at $k = m = p$, we have:

- $f_c(t+m) = \underbrace{h_m u(t|t) + \cdots + h_2 u(t+m-2|t) + h_1 u(t+m-1|t)}_{m \text{ terms (all future control actions)}}$

- $f_f(t+m) = \underbrace{h_N u(t-(N-m)) + \cdots + h_{m+2} u(t-2) + h_{m+1} u(t-1)}_{N-m \text{ terms (past control actions)}}$

So we can describe a matrix formulation; we define a vector of the past control actions and a vector of present and future control actions:

We define:

$u_- \in R^{(N-1)x1} \rightarrow$ vector of past control actions.
$u_+ \in R^{mx1} \rightarrow$ vector of present and future control actions
$\hat{y} \in R^{mx} \rightarrow$ vector of the output predictions along the prediction horizon.
We define two matrices:

$$\boldsymbol{H}_1 := \begin{pmatrix} h_1 & 0 & \dots & 0 \\ h_2 & h_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ h_m & h_{m-1} & \dots & h_{+1} \end{pmatrix} \in \mathbb{R}^{m\times m}$$

$$\boldsymbol{H}_2 := \begin{pmatrix} h_N & h_{N-1} & \dots & h_i & \dots & h_2 \\ 0 & h_N & \dots & h_{i+1} & \dots & h_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & h_N & \dots & h_{m+1} \end{pmatrix} \in \mathbb{R}^{m\times(N-1)}$$

And we define the predicted output as:

$$\hat{y} = H_1 u_+ + H_2 u_- + n$$

$$\hat{y} = H_1 u_{1_+} + f$$

the reference trajectory w is:

$$w(t+k) = \begin{cases} w(t) = y_m(t) \\ aw(t+k-1) + (1-a)r(t+k), \quad k = 1,..m \end{cases}$$

Once we have all the needed quantity, we define the optimization problem. We have a least square minimization:

$$J = e^T e + \lambda u^T u$$

$$u_+^* = (H_1^T H_1 + \lambda I)^{-1} H_1^T (w - f)$$

We know that $m = p$. If $d = 0, \lambda = 0$ then we obtain:

$$u^* = H_1^{-1}(w - f)$$

Predictive functional control (PFC)

The optimization problem is performed faster; in fact, PFC aims at reducing the number of unknown of the optimization problem. It is divided into 2 methodologies:

1. The control signal is a linear combination of **base functions;**
2. We evaluate the error not along the whole prediction horizon but only at some coincidence points.

In this case, we use a state-space process model; this means that some of the limitations that we had for previous methods are not present; for instance, we can deal with unstable process and with process with integrators. We can also deal with delay, but they have to be implicit in the model.

The discrete-time state-space model LTI is:

$$\begin{cases} x(t) = Mx(t-1) + Nu(t-1) \\ y(t) = Qx(t) \end{cases}$$

We also assume to have a constant disturbance along the horizon:

$$\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t), \qquad k = 1, \dots, p$$

Let's face the case 1., in which we have a **structured control law**; the future control actions are linear combinations of some base functions:

$$u(t+k|t) \coloneqq \sum_{i=1,\dots,n_B} \mu_i(t) B_i(k)$$

The unknown are the coefficients μ_i; they are n_B (number of base functions that we use to represent the control law). $n_B < N_u$ so we have drastically reduced the number of unknown to determine. A very common and used base function set is the <u>polynomial</u> one:

$$\mathcal{B} = \{B_i = k^{i-1}, i = 1, \dots, n_B\}$$

We have to compute the predicted output. We exploit the fact that we are using known inputs, so we are using the base functions to compute the inputs. First of all, we compute the system response to the base functions. We are always dealing with linear systems; so, in the end, we will express the system response as a linear combination of these free responses to the base functions. The prediction is:

$$\hat{y}(t+k|t) = QM^k x(t) + \sum_{i=1,\dots,n_B} y_{B_i}(k)\mu_i(t) + \hat{n}(t+k|t)$$

With:

$$\sum_{i=1,\dots,n_B} y_{B_i}(k) = (QM^{k-1}NB_i(0) + QM^{k-2}NB_i(1) + \cdots + QNB_i(k-1))$$

So the output is a linear combination of known functions.

We can define the predicted error as the difference between the predicted output and the future trajectory:

$$\hat{e}(t+k|t) = \hat{y}(t+k|t) - \hat{w}(t+k|t)$$

Let's now face case 2. In this case we will evaluate the cost function only over some coincidence points:

$$J(\underline{u}) = \sum_{j=1}^{n_H} \left|\hat{e}(t+h_j|t)\right|^2 + \lambda \sum_{j=1}^{n_H} \left(\Delta u(t+h_j-1|t)\right)^2 = \underline{\hat{e}}^T \underline{\hat{e}} + \lambda \underline{u}^T \underline{u}$$

Now we can solve the optimization problem.

$$\min_{u(t+h_i-1),i=1,2,\dots,n_H} J(u)$$

The solution is

$$\underline{\mu}^* = \left(\underline{Y}_B^T \underline{Y}_B\right)^{-1} \underline{Y}_B^T \underline{d}$$

Where $\underline{d} = \left(\underline{w} - \underline{f}\right)$.

$$\boldsymbol{Y_B} := \begin{pmatrix} \boldsymbol{y}_B(h_1) \\ \dots \\ \boldsymbol{y}_B(h_{n_H}) \end{pmatrix} \in \mathbb{R}^{n_H \times n_B}$$

$$\boldsymbol{d}(t) := \begin{pmatrix} d(t+h_1) \\ \dots \\ d(t+h_{n_H}) \end{pmatrix} \in \mathbb{R}^{n_H \times 1}$$

$\boldsymbol{y}_B(h_j) := \begin{pmatrix} y_{B_1}(h_j) & \dots & y_{B_{n_B}}(h_j) \end{pmatrix} \in \mathbb{R}^{1 \times n_B}$ ← this vector collects the base function responses at time h_j

$\boldsymbol{\mu}(t) := \begin{pmatrix} \mu_1(t) \\ \dots \\ \mu_{n_B}(t) \end{pmatrix} \in \mathbb{R}^{n_B \times 1}$ ← vector of the n_B unknows

$d(t+h_j) := \hat{w}(t+h_j|t) - QM^{h_j} x(t) - \hat{n}(t+k|t)$ ← reference trajectory – free response

We are almost done; we have these n_B coefficients and we can use them to compute the control action $u^*(t|t)$:

$$u^*(t|t) := \sum_{i=1,\dots,n_B} \mu_i^*(t) B_i(0)$$

MPC WITH NO MODEL UNCERTAINTY

We have a discrete-time system that is not subject to any uncertainty (either in the form of unknown additive disturbance or imprecise knowledge of the system parameters):

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k+1) = Cx(k+1) \end{cases}$$

Where $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$.

Our preliminary assumption is that we can control this system, so the pair (A, B) is stabilizable and the pair (A, C) is observable. We also assume that we can measure the state.

Since we are in an optimization framework, we should add a cost function and constraints (in particular, **linear constraints** because we are talking about the dynamic of the system). We express generic constraints in terms of:

$$Fx(k) + Gu(k) \leq b$$

Where $F \in \mathbb{R}^{n_c \times n_x}$, $G \in \mathbb{R}^{n_c \times n_u}$, $b \in \mathbb{R}^{n_c}$ (we have n_c constraints). If $F = 0$ we have input constraints, while if $G = 0$ we have state constraints. Moreover, we can write nonstationary constraints just by writing that both F and G depends on the time k. In process control, we always have to deal with constraints since quantities are always bounded.

A pair $\big(x(k), u(k)\big)$ is *feasible* if it verifies the constraints:

$$\big(x(k), u(k)\big) \Leftrightarrow Fx(k) + Gu(k) \leq b$$

A sequence $\big\{\big(x(0), u(0)\big), \big(x(1), u(1)\big), \dots\big\}$ is *feasible* if all the pairs are feasible:

$$\big\{\big(x(0), u(0)\big), \big(x(1), u(1)\big), \dots\big\} \Leftrightarrow \big(x(k), u(k)\big)\ is\ feasible, k = 0,1, \dots$$

Now, we look at the **optimal regulation problem**. In the classical regulation problem, we want to design a controller that drives the system state to some desired reference point; in our case, this is the origin of the state-space, so we want $x(k) \to 0$ with time. In the regulation problem within the optimal framework, we have to design a cost.

We define a quadratic cost which we have to write in the infinite horizon and unconstrained case:

$$J(x(0), \{u(0), u(1), \dots\}) = \sum_{k=0}^{\infty} \big(\|x(k)\|_Q^2 + \|u(k)\|_R^2\big)$$

Where $Q \in \mathbb{R}^{n_x \times n_x}$, $R \in \mathbb{R}^{n_u \times n_u}$. J depends on the initial state $x(0)$ and on the sequence of infinite control actions.

We assume that R is symmetric, positive definite (i.e., the eigenvalues are real and strictly positive) and Q is symmetric, positive semidefinite (i.e., the eigenvalues are real and nonnegative).

Now, we want to find the sequence of control actions which minimizes the cost, so we define an infinite horizon quadratic problem.

J^* is the minimum cost over all the possible sequences of control actions:

$$J^*\big(x(k)\big) = \min_{\{u(k), u(k+1), \dots\}} J(x(k), \{u(k), u(k+1), \dots\})$$

The constrained infinite horizon problem has no closed form solution. Therefore, let's give a look to the unconstrained quadratic problem first. The problem of minimizing the quadratic cost in the unconstrained case is addressed by **Linear Quadratic optimal control**.

Unconstrained infinite horizon quadratic problem

We have a **linear state feedback control**. Our control law is:

$$u(k) = Kx(k)$$

Where $K \in \mathbb{R}^{n_u \times n_x}$. With this kind of control is that we can write the closed loop dynamics very easily:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ u(k) = Kx(k) \end{cases} \Rightarrow x(k+1) = (A + BK)x(k)$$

This means that we can write both the state and the control action at time k in terms of the initial state only:

$$\begin{cases} x(k) = (A + BK)^k x(0) \\ u(k) = K(A + BK)^k x(0) \end{cases}$$

The consequence is that J can be written just as:

$$J(x(0), \{u(0), u(1), \dots\}) = J\big(x(0)\big)$$

This is a great advantage because we don't have an infinite number of control actions to optimize anymore, but we just have to choose the optimal gain. From optimal control, we know that $J\big(x(0)\big)$ can be written as:

$$J\big(x(0)\big) = x(0)^T W x(0) = \|x(0)\|_W^2$$

Where $W \in \mathbb{R}^{n_x \times n_x}$ and is equal to:

$$W = \sum_{k=0}^{\infty} [((A + BK)^k)^T (Q + K^T R K)(A + BK)^k]$$

If $(A + BK)$ is strictly stable (i.e., the eigenvalues are within the unit circle), then W has a finite number of elements. Moreover, if R is positive definite and Q is positive semidefinite, then also $J(x)$ is a positive definite function which implies that W is positive definite.

LEMMA 1 (LYAPUNOV MATRIX EQUATION)

The matrix W is the unique positive definite solution of the Lyapunov matrix equation:

$$W = (A + BK)^T W (A + BK) + Q + K^T R K$$

If and only if $(A + BK)$ is strictly stable.

Thanks to the knowledge of W we can express the cost in closed form; this means that we can also solve our problem because we can find the optimal gain K^* and therefore the optimal control law by solving a discrete time algebraic Riccati equation.

THEOREM 1 (DISCRETE TIME ALGEBRAIC RICCATI EQUATION)

The optimal gain matrix K^* minimizing $J\big(x(0)\big) = \|x(0)\|_W^2$ for any initial state $x(0) \in \mathbb{R}^{n_x}$ is the unique solution of the Riccati equation:

$$K^* = (B^T W B + R)^{-1} B^T W A$$

Notice that K^* is the same for any initial point.

So, we have the optimal control law for the discrete-time LTI system under quadratic cost function without constraints:

$$u(k) = K^* x(k)$$

MPC dual-mode prediction paradigm

MPC has 2 main differences with respect to LQ optimal control:

1. We deal with a finite horizon
2. We deal with constraints

The **infinite horizon constrained problem** can be written as:

$$\min_{\{u(0|k),u(1|k),\dots\}} J(x(k), \{u(0|k), u(1|k), \dots\})$$
$$s.t.\ \ Fx(i|k) + Gu(i|k) \leq b,\ \ i = 0,1, \dots$$

Where:

$$J(x(k), \{u(0|k), u(1|k), \dots\}) = \sum_{i=0}^{\infty} \left(\|x(i|k)\|_Q^2 + \|u(i|k)\|_R^2\right)$$

And $x(i|k)$ and $u(i|k)$ denote the predicted values of the model state and input, respectively, at time $k+i$ based on the information available at time k. Notice that we have an infinite number of constraints; thus, the constrained minimization of this cost would be an infinite-dimensional optimization problem, which is in principle intractable.

However, we will divide this problem into two subproblems:

- In *mode 1*, we solve a finite horizon constrained problem:

  $$\min_{\{u(0|k),u(1|k),\dots,u(N-1|k)\}} J_{RH}\,(x(k), \{u(0|k), u(1|k), \dots, u(N-1|k)\})$$

  $$s.t. Fx(i|k) + Gu(i|k) \leq b, \qquad i = 0,1, \dots, N-1$$

  We have a number of N control actions to determine under N constraints.

- In *mode 2*, we solve the infinite horizon unconstrained problem:

  $$\min_{\{u(N|k),u(N+1|k),\dots\}} J\big(x(N|k)\big)$$

We have to define how the function J_{RH} looks like:

$$\underline{u}(k) = \{u(0|k), \dots, u(N-1|h)\}$$

$$J_{RH}\left(x(k),\underline{u}(k)\right)=\underbrace{\sum_{i=0}^{N-1}\left(\|x(i|k)\|_Q^2+\|u(i|k)\|_R^2\right)}_{J\ over\ the\ finite\ horizon}+\underbrace{\|x(N|h)\|_W^2}_{final\ cost\ (penalty\ term)}$$

The final cost is the cost that we reach after N samples (so after the mode 1 is finished) under the control $u = k * x$.

If we use this controller (regulation), then after some samples, $x \to 0$ but this implies that also $u \to 0$ so we are in the pair $\left(x(k), u(k)\right) = (0,0)$. This satisfies the constraints all the time. Therefore, after N samples it is like the constraints are not more active since they are always satisfied. This is exactly the infinite horizon unconstrained problem that we solved before. Now, we can limit ourselves to the study of the mode 1 (finite horizon constrained problem) by means of optimization:

$$\min_{\underline{u}(k)} J_{RH}\left(x(k),\underline{u}(k)\right)$$

$$s.t. Fx(i|k) + Gu(i|k) \leq b, \qquad i = 0,1,\dots,N-1$$

We can solve it numerically.

So, we have two solutions; one is the optimal solution for the first N steps of the constrained problem:

$$u(i|k) = g_{RH}\big(x(k)\big), \quad i = 0,\dots,N-1$$

The other is the optimal solution for the remaining steps in the unconstrained case:

$$\begin{cases} u(i|k) = k^* x(N|k), & i = N, N+1, \dots \\ J^*\big(x(N|k)\big) = \|x(N|k)\|_W^2 \end{cases}$$

Combining these solutions, we have the optimal solution for the constrained infinite horizon problem which we denote as:

$$\{u^\infty(0|k), u^\infty(1|k), \dots\}$$

We have concluded that:

$$\begin{cases} u^\infty(i|k) = k^* x(i|k), & i = \overline{N}, \overline{N}+1, \dots \\ J^*\big(x(\overline{N}|k)\big) = \|x(\overline{N}|k)\|_W^2 \end{cases}$$

If the prediction horizon of mode 1 is chosen to be sufficiently long, namely if $N \geq \overline{N}$, then:

solution of the finite horizon constraints quadratic programming problem with terminal cost $\|x^*(\overline{N}|k)\|^2$

Optimal solution if the LQR problem starting at time $k + N$

$$\{u^*(0|k), u^*(1|k), \dots, u^*(N-1|h), k^* x^*(N|k), k^* x^*(N+1|k), \dots\} =$$

$$= \{u^\infty(0|k), u^\infty(1|k), \dots, u^\infty(N-1|h), u^\infty(N|k), u^\infty(N+1|k), \dots\}$$

This is the optimal solution for the infinite horizon constrained quadratic problem.

THEOREM

If there exists a finite horizon $\overline{N}$ which depends on the starting point $x(k)$ such that if $N \geq \overline{N}$, then:

1. $\underline{u}^*(k)$ coincides with the optimal control action: $\{u^\infty(0|k), u^\infty(1|k), \dots, u^\infty(N-1|k)\}$.
2. $J_{RH}\left(x(k), \underline{u}^*(k)\right) = J^*(k)$, with $J^*(k)$ optimal cost of the infinite constrained problem.