Lecture 10: Practical Issues

I. Reference Tracking

- **Initial Remarks: Steady State Tracking Problem Setup**

Consider the linear system model with selected output which might be a subspace (or say, linear combination) of the state:

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

Goal: Track given reference r such that $y_k \to r$ as $k \to \infty$.

Note:

This is a simplified scenario, and it usually is used in chemical and bioengineering. Reaching the target accurately is very important for them while the transient process is actually not that vital. For robotics, this kind of steady state is usually not sufficient.

It is obvious that in this simple setting the reference r could not continue varying. Therefore, we need to find a steady state (fixed) r and determine the steady state target (x_s, u_s), i.e.:

$$\begin{aligned} x_s &= Ax_s + Bu_s \\ Cx_s &= r \end{aligned} \iff \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

In other words, given a target r and the system (A, B, C), we calculate the steady state and inputs through the above equation, and we hope it has the solution.

Note: The existence of the solution to the above linear equation can be discussed, here for simplicity, we only consider the dimension issue. Firstly, the system matrix A is apparently square, then:

1. If we have the same number of the control and output (# of columns of B $=$ # of rows of C), the resulting equation matrix is square, and the equation probably has a unique solution (with the determinant of the equation matrix not to be 0).
2. If the control input is more than the target output (# of columns of B $>$ # of rows of C), the resulting equation matrix is a "fat" matrix, in this scenario, and the equation probably has a solution, even multiple solutions.
3. If we have less control than the target output (# of columns of B $<$ # of rows of C), the resulting equation matrix is a "thin" matrix, in this scenario, and the equation probably has no solution, it is in some sense a kind of "underactuated" scenario.

- **Steady State Target**

In real applications, the system has constraints. Therefore, we need to consider the constrained problem. In case of multiple feasible u_s (i.e. the "fat" matrix case discussed in the notes above), we can choose to compute the 'cheapest' steady state (x_s, u_s) and result in the optimization problem:

$$\min \ u_s^{\mathrm{T}} R_s u_s$$
$$\text{subj. to. } \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$
$$x_s \in \mathcal{X}, \ u_s \in \mathcal{U}$$

In general, we assume that the target problem is feasible. But if no solution exists (e.g. the "thin" matrix case discussed in the notes above), we can compute the reachable set point that is "closest" to r:

$$\min \ (Cx_s - r)^{\mathrm{T}} Q_s (Cx_s - r)$$
$$\text{subj. to. } x_s = Ax_s + Bu_s$$
$$x_s \in \mathcal{X}, \ u_s \in \mathcal{U}$$

Note: these are not the only way to handle the problem, but they are the most generally used method.

- **RHC Reference Tracking**

Now, we assume that we have already solved the problem of determine the steady state target (x_s, u_s) using the method mentioned above. We now use control (MPC) to bring the system to a desired steady-state condition (x_s, u_s), yielding the desired output $y_k \to r$.

The MPC is designed as follows:

$$\min_{u_0,\dots,u_{N-1}} \ \|y_N - Cx_s\|_P^2 + \sum_{k=0}^{N-1} \|y_k - Cx_s\|_Q^2 + \|u_k - u_s\|_R^2$$
$$\text{subj. to. [model constraints]}$$
$$x_0 = x(k)$$

Drawback: the controller will show offset **in case of unknown model errors or disturbances.**

- **RHC Reference Tracking without Offset**

Consider the discrete-time, time-invariant system (possibly nonlinear, uncertain)

$$x_m(k+1) = g(x_m(k), u(k))$$
$$y_m = h(x_m(k))$$

Objective:

Design an MPC in order to make $y(k)$ track the reference signal $r(k)$, i.e. $(y(k) - r(k)) \to 0$ for $t \to \infty$. We would study step references and focus on zero steady state error: $y(k) \to r_\infty$ as $k \to \infty$.

Consider the augmented model:

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k)$$
$$d(k+1) = d(k)$$
$$y(k) = Cx(k) + C_d d(k)$$

with constant disturbance $d(k) \in \mathbb{R}^{n_d}$

Note: This is to say, we introduce a hypothetical input (disturbance) which we actually don't know the value but know it is a constant. In the end, we want to observe the system and then estimate this $d(k)$ such that our system can track the reference without offset. We assume the disturbance to be constant since we only care about the steady state and this idea is a bit like PI control: we use this $d(k)$ and its accumulating effect along the time to try to capture the complex structure (possibly nonlinear, uncertain etc.) of the offset.

State observer for augmented model is then (prediction + correction):

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(-y_m(k) + C\hat{x}(k) + C_d \hat{d}(k) \right)$$

Lemma (Steady State of the Observer):

Suppose the observer is stable and the number of outputs p equals the dimension of the constant disturbance n_d. The observer steady state satisfies:

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ y_{m,\infty} - C_d \hat{d}_\infty \end{bmatrix}$$

Where $y_{m,\infty}$ and u_∞ are the steady state measured outputs and inputs.

That is to say, the observer output $C\hat{x}_\infty + C_d \hat{d}_\infty$ tracks the measurement $y_{m,\infty}$ without offset.

$\Rightarrow$ If we set $y_{m,\infty} = r_\infty$, we get our goal for offset-free tracking. This property gives us inspiration to design the MPC tracking without reference.

Note: Now we assume that the state observer is stable and there are requirements on the dimensions of the outputs and disturbance. The condition of the observer to be stable, in other words, how to properly construct an observable augmented system (choosing the B_d, C_d), and a stable observer (choosing the L_x, L_d), would be discussed in the next section.

MPC Reference Tracking without Offset: Final Formulation

$$\begin{aligned} \min_{u_0,\dots,u_{N-1}} \quad & \|x_N - \bar{x}_k\|_P^2 + \sum_{k=0}^{N-1} \|x_k - \bar{x}_k\|_Q^2 + \|u_k - \bar{u}_k\|_R^2 \\ \text{subj. to.} \quad & x_{k+1} = Ax_k + Bu_k + B_d d_k \quad k = 0,1,\dots,N \\ & x_k \in \mathcal{X},\ u_k \in \mathcal{U}, \qquad\qquad k = 0,1,\dots,N \\ & x_N \in \mathcal{X}_f \\ & d_{k+1} = d_k \qquad\qquad\qquad\quad k = 0,1,\dots,N \\ & x_0 = \hat{x}(k) \\ & d_0 = \hat{d}(k) \end{aligned}$$

With estimates $\hat{x}(k), \hat{d}(k)$ from the observer

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(-y_m(k) + C\hat{x}(k) + C_d \hat{d}(k) \right)$$

and the target $\bar{x}_k, \bar{u}_k$ given by:

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}(k) \\ r(k) - C_d \hat{d}(k) \end{bmatrix}$$

To sum up, the MPC tracking problem now is two-phase:

Phase 1 (Observer):

1. Use the augmented system with hypothetical disturbance (B_d, C_d) and construct the state observer with proper gains (L_x, L_d)
2. Get the real measurement $y_m(k)$ to get the estimate $\hat{x}(k), \hat{d}(k)$
3. Use the estimate $\hat{d}(k)$ to calculate the target $\bar{x}_k, \bar{u}_k$

Phase 2 (MPC controller):

1. Set the initial condition of the MPC to be the estimate $\hat{x}(k), \hat{d}(k)$
2. Set the target $\bar{x}_k, \bar{u}_k$ calculated above in the MPC cost and do MPC.

And this method is also guaranteed by the theorem:

Theorem (MPC tracking without offset):

Denote by $\kappa\left(\hat{x}(k), \hat{d}(k), r(k)\right) = u_0^*$ the control law when the estimated state and disturbance are $\hat{x}(k)$ and $\hat{d}(k)$. Consider the case where the number of constant disturbances equals the number of (tracked) outputs $n_d = p = r$. Assume the RHC is recursively feasible and unconstrained for $k \geq j$ with $j \in \mathbb{N}^+$ and the closed-loop system.

$$x(k+1) = f\left(x(k), \kappa\left(\hat{x}(k), \hat{d}(k), r(k)\right)\right)$$

$$\hat{x}(k+1) = (A + L_x C)\hat{x}(k) + (B_d + L_x C_d)\hat{d}(k) + B\kappa\left(\hat{x}(k), \hat{d}(k), r(k)\right) - L_d y_m(k)$$

$$\hat{d}(k+1) = L_d C\hat{x}(k) + (I + L_d C_d)\hat{d}(k) - L_d y_m(k)$$

Converges to $\hat{x}(k) \to \hat{x}_\infty, \hat{d}(k) \to \hat{d}_\infty, y_m(k) \to y_{m,\infty}$ as $t \to \infty$. Then $y_m(k) \to r_\infty$ as $t \to \infty$

Last question: How do we choose the matrices B_d and C_d in the augmented model and how do we choose the gains L_x, L_d in the observer?

Lemma (Observability of the Augmented System):

The augmented system, with the number of outputs p equal to the dimension of the constant disturbance n_d and $C_d = I$ is observable if and only if (C, A) is observable and;

$$\det\begin{bmatrix} A - I & B_d \\ C & I \end{bmatrix} = \det(A - I - B_d C) \neq 0$$

Note: If the plant has no integrators, then $\det(A - I) \neq 0$ and we can choose $B_d = 0$. If the plant has integrators then B_d has to be chosen specifically to make $\det(A - I - B_d C)$.

As for the stability of the state estimator, refer to the notes for lecture 3 about observer pole placement and Kalman filter.

II. Soft Constraints

- **Initial Remarks: Motivation of Soft Constraints**

Input constraints are dictated by physical constraints on the actuators and are usually “hard”, state or output constraints arise from practical restrictions on the allowed operating range and are **rarely hard**.

Hard state/output constraints always lead to **complications in the controller implementation** because:

1. Feasible operating regime is constrained even for stable systems
2. Controller patches must be implemented to generate reasonable control action when measured/ estimated states move outside feasible range because of disturbances or noise.

Therefore, In industrial implementations, **typically, state constraints are softened.**

- **Mathematical Formulation**

Original Problem:

$$\min_{z} \; f(z)$$

$$\text{subj. to. } g(z) \leq 0$$

"Softened" problem:

$$\min_{z.\epsilon} \; f(z) + l(\epsilon)$$

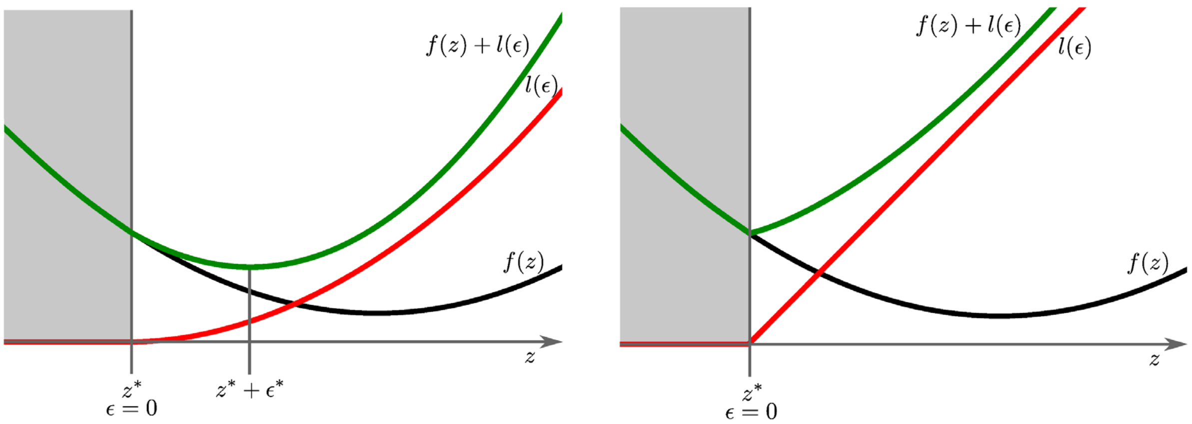
$$\text{subj. to. } g(z) \leq \epsilon$$

$$\epsilon \geq 0$$

Our requirement on $l(\epsilon)$:

If the original problem has a feasible solution z^*, then then the softened problem should have the same solution z^* and $\epsilon = 0$.

Quadratic Penalty and Linear Penalty



Constraint function $g(z) \triangleq z - z^* \leq 0$ induces feasible region (grey)

$\Rightarrow$ minimizer of the original problem is z^*

Introduce **Quadratic Penalty** $l(\epsilon) = v\epsilon^2$ for $\epsilon \geq 0$

$\Rightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^* + \epsilon^*, \epsilon^*)$ instead of $(z^*, 0)$ because at $\epsilon = 0$, the slope of quadratic function is still 0.

Introduce **Linear Penalty** $l(\epsilon) = u\epsilon$ for $\epsilon \geq 0$

$\Rightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^*, 0)$, because at $\epsilon = 0$, we can (and should) choose u large enough so that $u + \lim_{z \to z^*} f'(z) > 0$.

- **Main Result**

From the discussion above, we can see that the quadratic penalty help but does not guarantee our requirement, the linear penalty can exactly satisfy our requirement but the hessian of the linear penalty is not ensured to be positive (and hence not strongly convex). Therefore, we can use the blended version of the penalty that can handled every situation:

Theorem (Exact Penalty Function):

$l(\epsilon) = u\epsilon$ satisfies the requirement for any $u > u^* \geq 0$, where u^* is the optimal Lagrange multiplier for the original problem. To make the objective strongly convex, we typically use:

$$l(\epsilon) = u\epsilon + v\epsilon^2 \text{ with } u > u^* \text{ and } v > 0$$

And the extension to multiple constraints $g_j(z) \leq 0,\ j = 1, \ldots, r$ is:

$$l(\epsilon) = \sum_{j=1}^{r} u_j \epsilon_j + v_j \epsilon_j^2 \text{ with } u_j > u_j^* \text{ and } v_j > 0$$

III. Generalizing the MPC Horizon

Modify the problem as:

$$\begin{aligned}
\min_U \ & \|x_{N_y}\|_P^2 + \sum_{k=0}^{N_y - 1} \|x_k\|_Q^2 + \|u_k\|_R^2 \\
\text{subj. to. } & y_{\min}(k) \leq y_k \leq y_{\max}(k), && k = 1, \ldots, N_c \\
& u_{\min} \leq u_k \leq u_{\max}, && k = 0, 1, \ldots, N_u \\
& x_0 = x(k) \\
& x_{k+1} = Ax_k + Bu_k, && k \geq 0 \\
& y_k = Cx_k, && k \geq 0 \\
& u_k = Kx_k && N_u \leq k \leq N_y
\end{aligned}$$

With $N_u \leq N_y$ and $N_c \leq N_y - 1$

Many applications require time-varying constraints, e.g. $y_{\min}(k),\ y_{\max}(k)$

Complexity can be reduced by introducing separate horizons N_u, N_c, N_y

But: all theoretical feasibility and stability guarantees are lost! **So this is generally not recommended!**

One effective and practical way to reduce the computational effor is: Move-blocking,

Strategy: select several optimization variables (control) to be fixed over time intervals in the future $\Rightarrow$ degrees of freedom in optimization problem are reduced, and by choosing the blocking strategies carefully RHC stability results remain applicable.

Note: with computing resource increasing, it actually doesn't matter that much now.