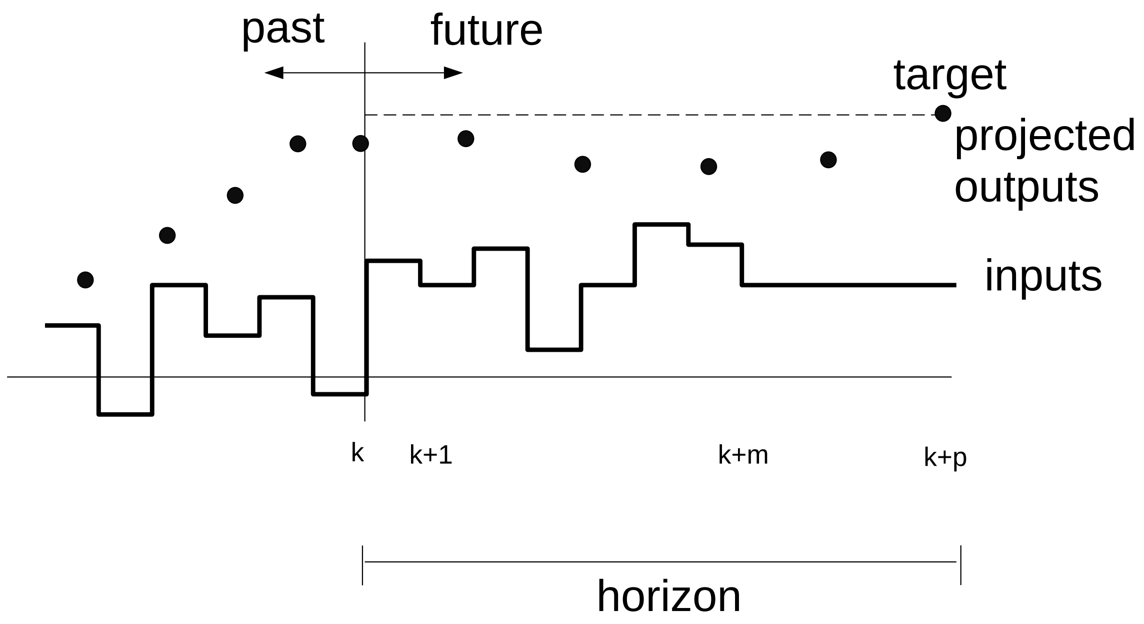


- *Memory:* stores the effect of *past* inputs on future outputs.
- *Predictor:* combines information stored in the memory with model information to predict the effect of hypothesized future input adjustments on future outputs.

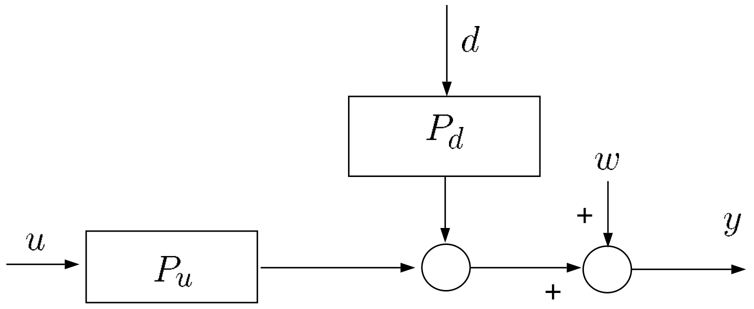
  $$(y(k{+}1|k), y(k{+}2|k), \ldots, y(k{+}p|k)) = f\left(\mathcal{I}(k), (\Delta u(k), \ldots, \Delta u(k+m-1))\right)$$

  where $\mathcal{I}(k)$ denotes the all the available information at k (stored in the memory).
- *Objective function and constraints*
- *Optimization program*

User-chosen parameters are the prediction horizon, control horizon, and parameters in the objective function and constraints.

2.3.2 BASIC PROBLEM SETUP

The basic system description we assume is



u : manpulated variable
d : measured / modelled disturbance
w_y : unmeasured disturbance + model / bias error effect

2.3.3 DEFINITION AND UPDATE OF MEMORY

Define the memory (state vector) as the effect of *past* deviation + current bias of *known* inputs (u and d) on the future output behavior:

$$\tilde{Y}(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{bmatrix} \quad \text{with} \quad \begin{array}{r} \Delta u(k) = \Delta u(k+1) = \cdots\cdots = 0 \\ \Delta d(k) = \Delta d(k+1) = \cdots\cdots = 0 \\ w_y(k) = w_y(k+1) = \cdots\cdots = 0 \end{array}$$

The memory update simply consists of:

$$M\tilde{Y}(k-1) + \begin{bmatrix} S^u & S^d \end{bmatrix} \overbrace{\begin{bmatrix} \Delta u(k-1) \\ \Delta d(k-1) \end{bmatrix}}^{\Delta v(k-1)} \quad \rightarrow \quad \tilde{Y}(k)$$

2.3.4 PREDICTION EQUATION

We can develop the prediction based on $\tilde{Y}(k)$ in the following manner ($y(k+\ell|k)$ denotes $y(k+\ell)$ predicted at $t=k$):

$$
\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ \vdots \\ y(k+p|k) \end{bmatrix}
= \underbrace{\begin{bmatrix} \tilde{y}(k+1/k) \\ \tilde{y}(k+2/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+p/k) \end{bmatrix}}_{y(k+1),\cdots,y(k+p) \text{ with } \begin{array}{c} \Delta u(k)=\Delta u(k+1)=\cdots\cdots=0 \\ \Delta d(k)=\Delta d(k+1)=\cdots\cdots=0 \\ w_y(k)=w_y(k+1)=\cdots\cdots=0 \end{array}}
$$

$$
+\begin{bmatrix} S_1^u \\ S_2^u \\ \vdots \\ \vdots \\ S_p^u \end{bmatrix} \Delta u(k|k) + \begin{bmatrix} 0 \\ S_1^u \\ S_2^u \\ \vdots \\ S_{p-1}^u \end{bmatrix} \Delta u(k+1|k) + \cdots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ S_1^u \end{bmatrix} \Delta u(k+p-1|k)
$$

$$
+\begin{bmatrix} S_1^d \\ S_2^d \\ \vdots \\ \vdots \\ S_p^d \end{bmatrix} \Delta d(k) + \begin{bmatrix} 0 \\ S_1^d \\ S_2^d \\ \vdots \\ S_{p-1}^d \end{bmatrix} \Delta d(k+1|k) + \cdots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ S_1^d \end{bmatrix} \Delta d(k+p-1|k)
$$

$$
+\begin{bmatrix} w(k+1|k) \\ w(k+2|k) \\ \vdots \\ \vdots \\ w(k+p|k) \end{bmatrix}
$$

There are more than a few terms (marked with $(\cdot|k)$) on the right-hand side that are unavailabe at time k.

- Assume *piece-wise constant disturbances*, i.e.,

$$\Delta d(k+1|k) = \Delta d(k+2|k) = \cdots\cdots = 0$$

- Assume the effect of unmeasured disturbance, model error, etc. are described as a *piece-wise constant* signal.

$$w_y(k+1|k) = w_y(k+2|k) = \cdots\cdots = w_y(k|k) \approx \underbrace{y(k)}_{\text{real meas.}} - \underbrace{\tilde{y}(k/k)}_{\text{model prediction}}$$

- For flexibility in adjusting the computational load, consider only m ($\leq p$) input moves $(\Delta u(k|k), \Delta u(k+1|k), \cdots, \Delta u(k+m-1|k))$. This means, in your prediction, assume

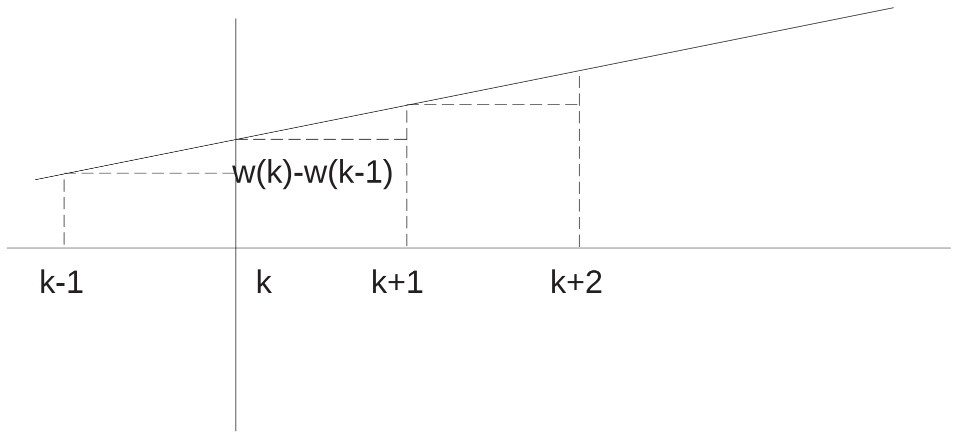
$$\Delta u(k+m|k) = \Delta u(k+m-1|k) = \cdots = \cdots = \Delta u(k+p-1|k) = 0$$

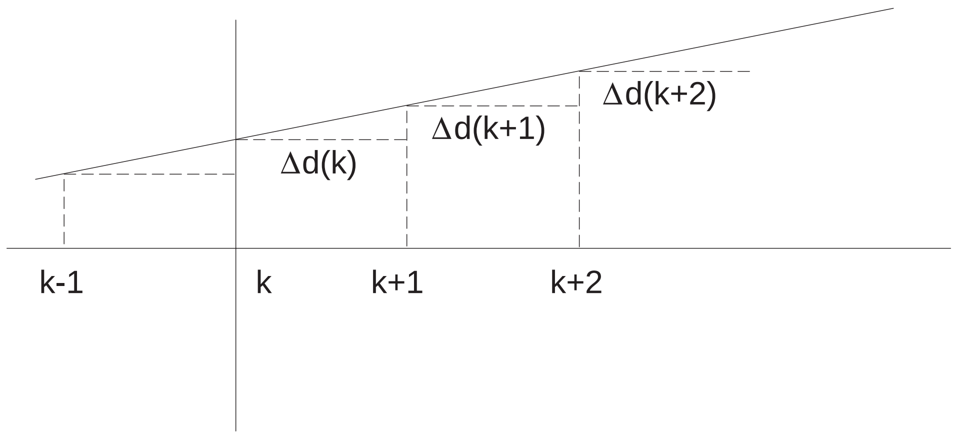
In summary,

$$
\begin{aligned}
\mathcal{Y}_{k+1|k} \; = \; & \underbrace{\begin{bmatrix} \tilde{y}(k+1/k) \\ \tilde{y}(k+2/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+p/k) \end{bmatrix}}_{\substack{M_p\tilde{Y}(k) \\ \text{from} \\ \text{memory}}} + \underbrace{\begin{bmatrix} S_1^d \\ S_2^d \\ \vdots \\ \vdots \\ S_n^d \end{bmatrix} \Delta d(k)}_{\substack{S^d\Delta d(k) \\ \text{feedforward} \\ \text{term}}} \\
+ & \underbrace{\begin{bmatrix} y(k)-\tilde{y}(k/k) \\ y(k)-\tilde{y}(k/k) \\ \vdots \\ \vdots \\ y(k)-\tilde{y}(k/k) \end{bmatrix}}_{\substack{\mathcal{I}_p(y(k)-\tilde{y}(k/k)) \\ \text{feedback} \\ \text{term}}} + \underbrace{\begin{bmatrix} S_1^u & 0 & \cdots & \cdots & 0 \\ S_2^u & S_1^u & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ S_m^u & S_{m-1}^u & \cdots & \cdots & S_1^u \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ S_p^u & S_{p-1}^u & \cdots & \cdots & S_{p-m+1}^u \end{bmatrix}}_{\substack{\mathcal{S}^u \\ \text{dynamic} \\ \text{matrix}}} \underbrace{\begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix}}_{\substack{\Delta\mathcal{U}(k) \\ \text{future} \\ \text{input} \\ \text{moves}}}
\end{aligned}
$$

NOTE: More complex (dynamic) extrapolation of the feedback errors is possible. For instance, for ramp disturbances, use

$$
\begin{aligned}
&\mathbf{\Delta d(k)} = \Delta d(k+1|k) = \cdots\cdots = \Delta d(k+p-1|k) \\
&w_y(k+\ell|k) = w_y(k|k) + \ell(w_y(k|k) - w_y(k-1|k-1))
\end{aligned}
$$





2.3.5 QUADRATIC CRITERION

$$\begin{aligned}\min_{\Delta u(j|k)} \Big[V(k) \ &\overset{\Delta}{=}\ \textstyle\sum_{j=1}^{p} (r(k+i|k) - y(k+i|k))^T Q (r(k+i|k) - y(k+i|k)) \\ &\quad + \textstyle\sum_{\ell=0}^{m-1} \Delta u^T(k+\ell|k) R \Delta u(k+\ell|k) \Big]\end{aligned}$$

Q and R are weighting matrices; they are typically chosen as diagonal matrices.

Note that the objective function can be rewritten as

$$V(k) = \begin{bmatrix} r(k+1|k) - y(k+1|k) \\ r(k+2|k) - y(k+2|k) \\ \vdots \\ r(k+p|k) - y(k+p|k) \end{bmatrix}^T \begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q \end{bmatrix} \begin{bmatrix} r(k+1|k) - y(k+1|k) \\ r(k+2|k) - y(k+2|k) \\ \vdots \\ r(k+p|k) - y(k+p|k) \end{bmatrix}$$
$$+ \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix}^T \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix}$$

$$\Downarrow$$

$$V(k) = (\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k))^T \bar{Q} (\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k)) + \Delta\mathcal{U}^T(k) \bar{R} \Delta\mathcal{U}(k)$$

where

$$\mathcal{R}(k+1|k) = \begin{bmatrix} r(k+1|k) \\ r(k+2|k) \\ \vdots \\ \vdots \\ r(k+p|k) \end{bmatrix}; \quad \mathcal{Y}(k+1|k) = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ \vdots \\ y(k+p|k) \end{bmatrix}$$

and

$$\bar{Q} = \text{blockdiag}[Q,\ Q,\ \ldots, \ldots,\ Q]; \quad \bar{R} = \text{blockdiag}[R,\ R,\ \ldots, \ldots,\ R]$$

Note that

$$\mathcal{Y}(k+1|k) = \underbrace{M_p \tilde{Y}(k) + S^d \Delta d(k) + \mathcal{I}_p \left(y(k) - \tilde{y}(k/k) \right)}_{\text{known term}} + \mathcal{S}^{\mathcal{U}} \Delta\mathcal{U}(k)$$

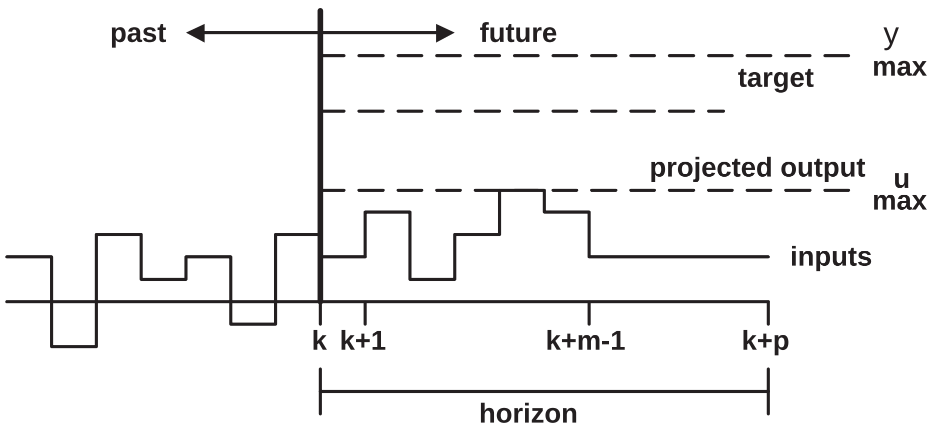
Hence, $V(k)$ is a quadratic function of $\Delta\mathcal{U}(k)$.

2.3.6 CONSTRAINTS

Constraints include

- Input magnitude constraints
- Input rate constraints
- Output magnitude constraints

At $t = k$, one has



$$
\begin{aligned}
& u_{\min} \le u(k+\ell|k) \le u_{\max} \\
& |\Delta u(k+\ell|k)| \le \Delta u_{\max}, \qquad \ell = 0, \cdots, m-1 \\
& y_{\min} \le y(k+j|k) \le y_{\max}, \quad j = 1, \cdots, p
\end{aligned}
$$

We want to express the above constraints as a linear inequalty in the form of

$$\mathcal{C}^{\mathcal{U}} \Delta \mathcal{U}(k) \le \mathcal{C}(k)$$

Manipulated Input Magnitude Constraints

$$u_{\min} \leq u(k+\ell|k) \leq u_{\max}, \qquad \ell = 0, \cdots, m-1$$

$$\Downarrow$$

$$\begin{array}{r}
\overbrace{u(k-1) + \sum_{i=0}^{\ell} \Delta u(k+i|k)}^{u(k+\ell|k)} \geq u_{\min} \\
\underbrace{-u(k-1) - \sum_{i=0}^{\ell} \Delta u(k+i|k)}_{-u(k+\ell|k)} \geq -u_{\max}
\end{array}$$

$$\Downarrow$$

$$\begin{bmatrix} \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ I & I & \cdots & I \end{bmatrix} \\ -\begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ I & I & \cdots & I \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix} \geq \begin{bmatrix} \begin{bmatrix} u_{\min} - u(k-1) \\ u_{\min} - u(k-1) \\ \vdots \\ u_{\min} - u(k-1) \end{bmatrix} \\ -\begin{bmatrix} u_{\max} - u(k-1) \\ u_{\max} - u(k-1) \\ \vdots \\ u_{\max} - u(k-1) \end{bmatrix} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} I_L \\ -I_L \end{bmatrix} \Delta \mathcal{U}(k) \geq \begin{bmatrix} \begin{bmatrix} u_{\min} - u(k-1) \\ \vdots \\ u_{\min} - u(k-1) \end{bmatrix} \\ -\begin{bmatrix} u_{\max} - u(k-1) \\ \vdots \\ u_{\max} - u(k-1) \end{bmatrix} \end{bmatrix}$$

Manipulated Variable Rate Cosntraints

$$|\Delta u(k+\ell|k)| \leq \Delta u_{\max}, \qquad \ell = 0, \cdots, m-1$$

$$\Downarrow$$

$$-\Delta u_{\max} \leq \Delta u(k+\ell|k) \leq \Delta u_{\max}$$

$$\Downarrow$$

$$\begin{gathered} \Delta u(k+\ell|k) \geq -\Delta u_{\max} \\ -\Delta u(k+\ell|k) \geq -\Delta u_{\max} \end{gathered}$$

$$\Downarrow$$

$$\begin{bmatrix} \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & 0 & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & I \end{bmatrix} \\ -\begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & 0 & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & I \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix} \geq \begin{bmatrix} -\begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ -\begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta \mathcal{U}(k) \geq \begin{bmatrix} -\begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ -\begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \end{bmatrix}$$

Output Magnitude Constraints

$$y_{\min} \le y(k+j|k) \le y_{\max}, \qquad j = 1, \cdots, p$$

$$\Downarrow$$

$$y(k+j|k) \ge y_{\min}$$
$$-y(k+j|k) \ge -y_{\max}$$
$$\Downarrow$$

$$\begin{bmatrix} M_p\tilde{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) + \mathcal{S}^{\mathcal{U}} \Delta \mathcal{U}(k) \\ -M_p\tilde{Y}(k) - \mathcal{S}^d \Delta d(k) - \mathcal{I}_p(y(k) - \tilde{y}(k/k)) - \mathcal{S}^{\mathcal{U}} \Delta \mathcal{U}(k) \end{bmatrix} \ge \begin{bmatrix} \mathcal{Y}_{\min} \\ -\mathcal{Y}_{\max} \end{bmatrix}$$

where

$$\mathcal{Y}_{\min} = \begin{bmatrix} y_{\min} \\ y_{\min} \\ \vdots \\ y_{\min} \end{bmatrix} \qquad \mathcal{Y}_{\max} = \begin{bmatrix} y_{\max} \\ y_{\max} \\ \vdots \\ y_{\max} \end{bmatrix}$$

since

$$\mathcal{Y}(k+1|k) \triangleq \begin{bmatrix} y(k+1|k) \\ \vdots \\ y(k+p|k) \end{bmatrix} = M_p\tilde{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) + \mathcal{S}^{\mathcal{U}} \Delta \mathcal{U}(k)$$

$$\Downarrow$$

$$\begin{bmatrix} \mathcal{S}^{\mathcal{U}} \\ -\mathcal{S}^{\mathcal{U}} \end{bmatrix} \Delta \mathcal{U}(k) \ge \begin{bmatrix} \mathcal{Y}_{\min} - M_p\tilde{Y}(k) - \mathcal{S}^d \Delta d(k) - \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \\ -\mathcal{Y}_{\max} + M_p\tilde{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \end{bmatrix}$$

In summary, we have

$$\begin{bmatrix} I_L \\ -I_L \\ I \\ -I \\ \mathcal{S}^{\mathcal{U}} \\ -\mathcal{S}^{\mathcal{U}} \end{bmatrix} \Delta\mathcal{U}(k) \geq \begin{bmatrix} \begin{bmatrix} u_{\min} - u(k-1) \\ \vdots \\ u_{\min} - u(k-1) \end{bmatrix} \\ -\begin{bmatrix} u_{\max} - u(k-1) \\ \vdots \\ u_{\max} - u(k-1) \end{bmatrix} \\ -\begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ -\begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ \begin{bmatrix} \mathcal{Y}_{\min} - M_p\tilde{Y}(k) - \mathcal{S}^d\Delta d(k) - \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \\ -\mathcal{Y}_{\max} + M_p\tilde{Y}(k) + \mathcal{S}^d\Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \end{bmatrix} \end{bmatrix}$$

The above is in the form of linear equality,

$$\mathcal{C}^{\mathcal{U}}\Delta\mathcal{U}(k) \geq \mathcal{C}(k)$$

Note that $\mathcal{C}^{\mathcal{U}}$ is a constant matrix while $\mathcal{C}(k)$ must be updated at each time step.

Although not treated here, time-varying constraints can easily be incorporated into the formulation.

2.3.7 QUADRATIC PROGRAMMING

Problem:

At each sample time, we have a minimization with an objective function

$$V(k) = (\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k))^T \bar{Q} (\mathcal{R}(k+1|k) - \mathcal{Y}(k+1|k)) + \Delta\mathcal{U}^T(k) \bar{R} \Delta\mathcal{U}(k)$$

with the prediction equation constraint

$$\mathcal{Y}(k+1|k) = M_p \tilde{Y}(k) + S^d \Delta d(k) + \mathcal{I}_p \left(y(k) - \tilde{y}(k/k) \right) + \mathcal{S}^{\mathcal{U}} \Delta\mathcal{U}(k)$$

and the inequality constraint

$$\mathcal{C}^{\mathcal{U}} \Delta\mathcal{U}(k) \geq \mathcal{C}(k)$$

Putting Into the Standard QP Form:
Substituting the prediction equation constraint into the objective gives

$$\begin{aligned} V(k) &= \mathcal{E}^T(k)\bar{Q}\mathcal{E}(k) - \underbrace{2\mathcal{E}^T(k)\bar{Q}\mathcal{S}^{\mathcal{U}}}_{\mathcal{G}^T(k)} \Delta\mathcal{U}(k) + \Delta\mathcal{U}^T(k) \underbrace{\left(\mathcal{S}^{\mathcal{U}T}\bar{Q}\mathcal{S}^{\mathcal{U}} + \bar{R}\right)}_{\mathcal{H}} \Delta\mathcal{U}(k) \\ \mathcal{E}(k) &= \mathcal{R}(k+1|k) - M_p\tilde{Y}(k) - S^d\Delta d(k) - \mathcal{I}_p \left(y(k) - \tilde{y}(k/k)\right) \end{aligned}$$

Note that $\mathcal{E}(k)$ can be computed with information given to us at time k. Hence, $V(k)$ is a quadratic function of $\Delta\mathcal{U}(k)$ with hessian matrix $\mathcal{H}$ and gradient vector $\mathcal{G}(k)$.

Since we have a minimization of a quadratic objective with a linear inequality constraint, we have a quadratic programming (QP). The standard form of quadratic programming is

$$\begin{gathered} \min_x x^T H x^T - g^T x \\ Cx \geq c \end{gathered}$$

The parameters that should be supplied to the QP solver are H, g, C and c.

In our case, at $t = k$,

$$
\begin{aligned}
&x : \Delta\mathcal{U}(k) \\
&H : \mathcal{H} \triangleq (\mathcal{S}^{\mathcal{U}})^T \bar{Q} \mathcal{S}^{\mathcal{U}} + \bar{R} \\
&g : \mathcal{G}(k) \triangleq 2(\mathcal{S}^{\mathcal{U}})^T \bar{Q}(M_p\tilde{Y}(k) + \mathcal{S}^d\Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) - \mathcal{R}_{k+1|k}) \\
&C : \mathcal{C}^u \triangleq \begin{bmatrix} I_L \\ -I_L \\ I \\ -I \\ \mathcal{S}^{\mathcal{U}} \\ -\mathcal{S}^{\mathcal{U}} \end{bmatrix} \\
&c : \mathcal{C}(k) \triangleq \begin{bmatrix} \begin{bmatrix} u_{\min} - u(k-1) \\ \vdots \\ u_{\min} - u(k-1) \end{bmatrix} \\ -\begin{bmatrix} u_{\max} - u(k-1) \\ \vdots \\ u_{\max} - u(k-1) \end{bmatrix} \\ \begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ -\begin{bmatrix} \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ \begin{bmatrix} \mathcal{Y}_{\min} - M_p\tilde{Y}(k) - \mathcal{S}^d\Delta d(k) - \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \\ -\mathcal{Y}_{\max} + M_p\tilde{Y}(k) + \mathcal{S}^d\Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) \end{bmatrix} \end{bmatrix}
\end{aligned}
$$

Following are some comments on quadratic programming.

- QP is convex and therefore fundamentally tractable.
- The solution doesn't necessarily lie at the vertices of feasible region(unlike LPs). One may have any number of active constraints

(up to the QP dimension).

- The size of QP is $m \times n_u$ where m is the control input horizon and n_u is the number of input. Computational time of QP depends on many things (e.g., Hessian size, its structure, number of constraints, the proximity of the solution to the active constraints) and is difficult to predict.
- Off-the-shelf QP solver is available, but is often not the best choice in terms of computational efficiency. Because the hessian and gradient for QP tends to be highly structured (sparse), an algorithm tailored to take advantage of this is recommended.
- QP solver requires inversion of the hessian. Since the hessian is a constant matrix (given fixed input / output weights and model parameters), it only needs to be inverted once off-line. This eliminates the time-consuming step of inverting the hessian at each QP run. Only when the weighting matrices are model parameters are changed, hessian needs to be recomputed and inverted in the background.
- Since most QPs are feasible-path algorithms, the number of inequality constraints also affect the computational time. One should use the constraints sparingly.
- The most well-known solution strategy is the active set strategy. In this method, first a feasible solution is found. Then, the least squares problem is solved with the active constraints as equality constraints. The optimality of the solution is checked through Kuhn-Tucker conditions. If they are not satisfied, the active constraint set is updated and the procedure is repeated.
- Another emerging method is the interior point (IP) method. In the IP method, a barrier function is used to trap the solution *within* the feasible region. Newton interation is used to converge to the optimum. This method has many attractive features like quick convergence (most

problems converge with 5-50 iterations regardless of the problem size) and ability exploit the problem structure.

2.3.8 SUMMARY OF REAL-TIME IMPLMENTATION

1. *Initialization:* Initialize the memory vector $\tilde{Y}(0)$ and the reference vector $\mathcal{R}(1|0)$. Set $k = 1$.
2. *Memory Update:*

$$\tilde{Y}(k-1) \quad \rightarrow \quad M\tilde{Y}(k-1)+S^u\Delta u(k-1)+S^d\Delta d(k-1) \quad \rightarrow \quad \tilde{Y}(k)$$

$$\begin{array}{c}\left[\begin{array}{c} \tilde{y}(k-1/k-1) \\ \tilde{y}(k/k-1) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-3/k-1) \\ \tilde{y}(k+n-2/k-1) \end{array}\right] \\ \tilde{y}(k+n-1/k-1) \\ \vdots \end{array} \qquad \left[\begin{array}{c} \tilde{y}(k/k-1) \\ \tilde{y}(k+1/k-1) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-2/k-1) \\ \tilde{y}(k+n-1/k-1) \end{array}\right] + \left[\begin{array}{c} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{n-1} \\ S_n \end{array}\right] \Delta v(k-1) = \left[\begin{array}{c} \tilde{y}(k/k) \\ \tilde{y}(k+1/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-2/k) \\ \tilde{y}(k+n-1/k) \end{array}\right]$$

3. *Reference Vector Update:* Update $\mathcal{R}(k+1|k)$ by shifting $\mathcal{R}(k|k-1)$ and entering a new reference value.

$$\left[\begin{array}{c} r(k|k-1) \\ r(k+1|k-1) \\ \vdots \\ \vdots \\ r(k+p-2|k-1) \\ r(k+p-1|k-1) \end{array}\right] \qquad\qquad \left[\begin{array}{c} r(k+1|k) \\ r(k+2|k) \\ \vdots \\ \vdots \\ r(k+p-1|k) \\ r(k+p|k) \end{array}\right]$$

4. *Measurement Intake:* Take in new measurement $y(k)$ and $\Delta d(k)$.

5. *Calculation of the Gradient Vector and Constraint Vector:*

   $$\tilde{G}(k) = 2(\mathcal{S}^{\mathcal{U}})^T \bar{Q}(M_p \tilde{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p(y(k) - \tilde{y}(k/k)) - \mathcal{R}(k+1|k))$$

   Update the constraint vector $\mathcal{C}(k)$ similarly.

6. *Solve QP:* Call the QP subroutine with pre-inverted $\mathcal{H}$,$\mathcal{C}^{\mathcal{U}}$ and computed $\mathcal{G}(k), \mathcal{C}(k)$.

7. *Implementation of input*: Implement $\Delta u(k|k)$:

   $$u(k) = u(k-1) + \Delta u(k|k)$$

8. Go back to Step 2 after setting $k = k + 1$.

2.4 ADDITIONAL ISSUES

2.4.1 FEASIBILITY ISSUE AND CONSTRAINT RELAXATION

- Output constraints can become infeasible (impossible to satisfy). For example, if we require $-\epsilon \leq y(k+\ell|k) \leq \epsilon$ for all ℓ, as $\epsilon \to 0$, the constraint becomes infeasible.
- When the QP is declared infeasible, one must relax the output constraints. Various ways to relax the constraints exist:
  - Relax the constraint starting from the initial time one by one until it is feasible.