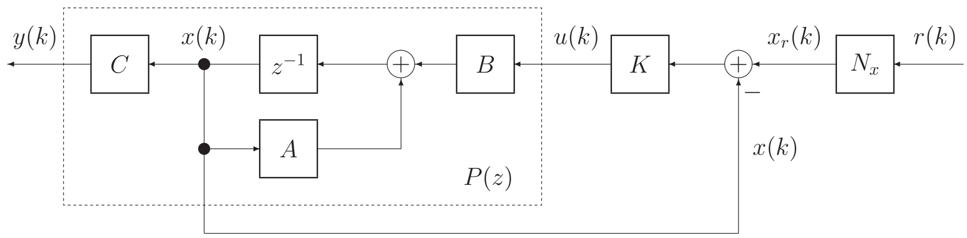
Reference Tracking

The idea is to set the problem up as driving the state to a desired reference value.

Approach: (assume state feedback for the moment)



Design the state feedback gain, K, for good closed-loop pole positions.

Implement it as: $u(k) \;=\; K\,(x_r(k) - x(k))$.

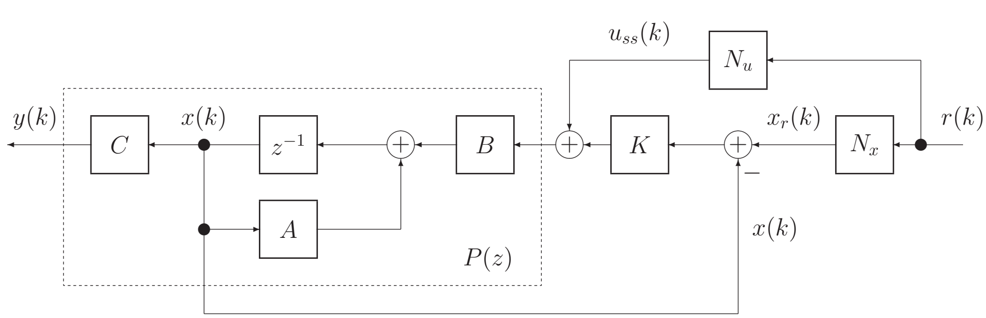
$x_r(k)$ can be thought of as a "reference state." This will make $x(k) \longrightarrow x_r(k)$ as $k \longrightarrow \infty$, with the specified closed-loop dynamics.

Reference inputs

The reference state, x_r, is determined by, $x_r(k) \;=\; N_x\, r(k)$.

For type 0 systems (no poles at $z = 1$), this will give a steady state error. In such systems $u \neq 0$ at a non-zero equilibrium, and so $x \neq x_r$.

Feedforward correction: $u_{ss}(k)$ provides a steady-state input.



The control input is: $u(k) \;=\; K\,(x_r(k) - x(k)) \;+\; u_{ss}(k)$.

Reference input gain matrices:

We must design N_x to generate the reference state, and N_u to generate the control input required to hold the system at the reference state.

Define $x_{ss} \;=\; \lim_{k \longrightarrow \infty} x(k)$ (the steady state value of $x(k)$, if it exists).

We want,

$$x_r \;=\; N_x\, r \;=\; x_{ss} \qquad \text{and} \qquad C\, x_{ss} = y_r \;=\; r.$$

This means that $C\, N_x\, r \;=\; r$, and if we want this to hold for all r, then we need,

$$C\, N_x \;=\; I.$$

At steady state,

$$x(k+1) \;=\; A\, x(k) \;+\; B\, u(k) \qquad \Longrightarrow \qquad x_{ss} \;=\; A\, x_{ss} \;+\; B\, u_{ss},$$

and rearranging gives,

$$(A - I)\, x_{ss} \;+\; B\, u_{ss} \;=\; 0 \qquad \Longrightarrow \qquad (A - I)\, N_x\, r \;+\; B\, N_u\, r \;=\; 0.$$

Again, we want this to hold for all r and so we require,

$$(A - I)\, N_x \;+\; B\, N_u \;=\; 0.$$

Reference input matrices:

Combining the previous equations gives,

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} \;=\; \begin{bmatrix} 0 \\ I \end{bmatrix},$$

or, if the inverse exists,

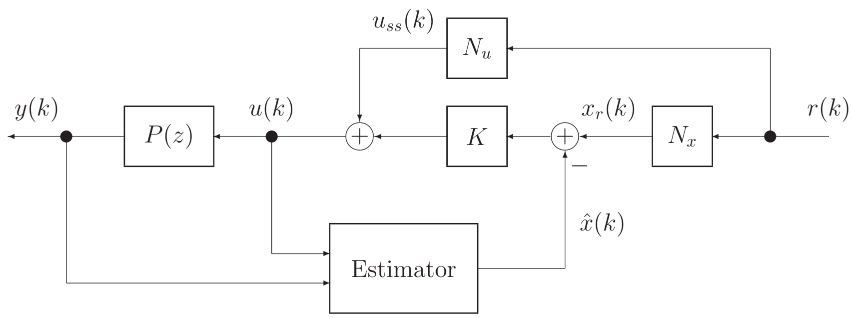
$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} \;=\; \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

The matrix, N_u is effectively the inverse of the plant steady state gain.

This design will give a zero steady state error to a step, but only if N_u provides the exact input required for the desired steady-state output.

This will not happen exactly in practice. What is a better method for getting zero steady state error?

Output feedback case: We simply augment this with an estimator.



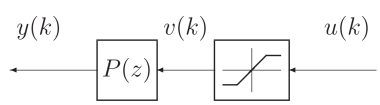
The control is applied using the estimated state, $\hat{x}(k)$, in place of the true state, $x(k)$.

$$u(k) \;=\; K\;(x_r(k) - \hat{x}(k)) \;+\; u_{ss}(k),$$

or, in terms of $r(k)$,

$$u(k) \;=\; K\;(N_x\, r(k) - \hat{x}(k)) \;+\; N_u\, r(k).$$

Input saturation (and other nonlinearities)



The actual input to the plant, $v(k)$, is limited:

$$v(k) = \begin{cases} u_{max} & \text{if} \;\; u(k) > u_{max} \\ u(k) & \text{if} \;\; u_{min} \leq u(k) \leq u_{max} \\ u_{min} & \text{if} \;\; u(k) < u_{min} \end{cases}$$

Common cases: $u_{min} = -u_{max}$ (symmetric), $u_{min} = 0$ (asymmetric)

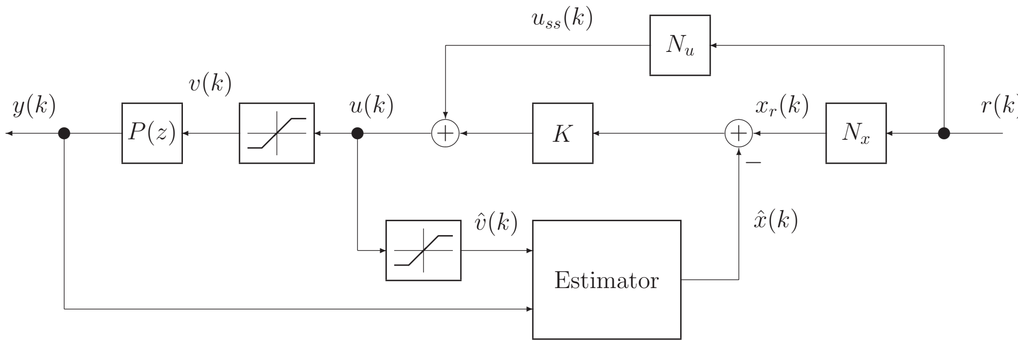
This happens in **every** physical situation.

Saturation can be a difficult problem if the control signal $u(k)$ frequently exceeds the limits, or exceeds them by a large margin.

The control design becomes nonlinear (and more difficult).

We can at least prevent the saturation from corrupting the estimator.

Estimator with input saturation



Model the actual plant input as $\hat{v}(k)$ (often accurate for saturation).

Make sure that the estimator "sees" the same input as the plant.

In some cases we can actually measure $v(k)$ and use that in the estimator.