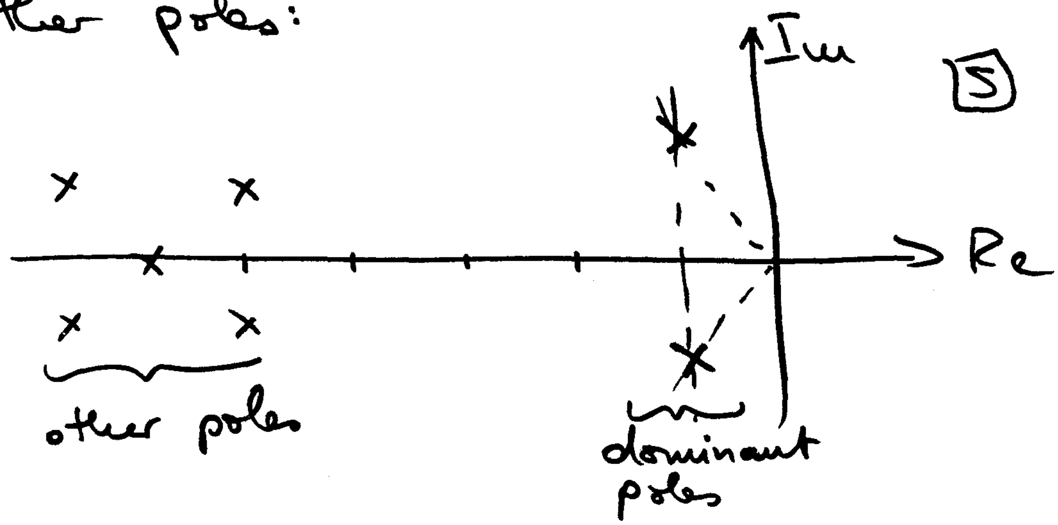
State Feedback through Optimal Control

Problem: Sometimes, the pole locations may be inconvenient to specify, because all our assumptions base on the analysis of second order systems. Can these assumptions be generally adapted to higher order systems? Studies indicate that this is only true with good accuracy if the dominant poles are five times closer to the imaginary axis than any other poles:

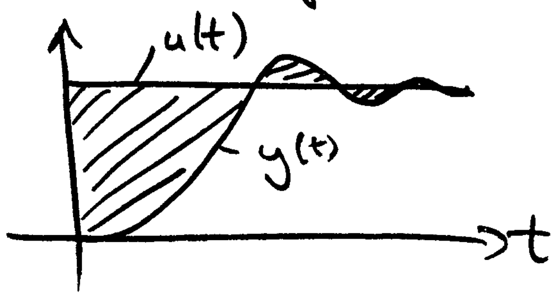


This, we can hardly afford as the feedback gains will become too large (in general). However, without this, we may be in for some surprises, e.g., when simulating the step response (overshoot >> 5%).

Deadbeat control can be an answer, but is optimized to <u>one</u> sort of input signal only.

<u>Question</u>: Can we come up with a design that does <u>not</u> base on second order approximations?

<u>Answer</u>: We need to formulate this as an optimization problem:



$$e = u - y$$

$\Rightarrow$ e.g. $$PI = \sum_{i=\emptyset}^{N} e_i^2 \overset{!}{=} \underset{\forall u}{\mathrm{Min}}$$

(ISE criterion),

or: $$PI = \sum_{i=\emptyset}^{N} |e_i| \overset{!}{=} \underset{\forall u}{\mathrm{Min}}$$

(IAE criterion),

or: $$PI = \sum_{i=\emptyset}^{N} i \cdot |e_i| \overset{!}{=} \underset{\forall u}{\mathrm{Min}}$$

(ITAE criterion).

We shall show that optimization leads to a state feedback if:

- system is linear
- performance index is quadratic
- $N \longrightarrow \infty$

Let us look first at the more general problem:

$$\left| \begin{array}{l} \underline{x}(k+1) = \underline{f}(\underline{x}(k), \underline{u}(k), k) \\ PI = \sum_{i=0}^{N-1} \varphi(\underline{x}(i), \underline{u}(i), i) \overset{!}{=} \underset{\forall \underline{u}(i)}{Min} \end{array} \right|$$

$\underline{u}(i)$ is a set of discrete parameters such that:

$$\frac{\partial PI}{\partial \underline{u}(i)} = 0$$

We can introduce a set of Lagrangian multipliers $\lambda(i)$:

$$\left| \begin{array}{l} \widetilde{PI} = \sum_{i=0}^{N-1} \varphi(\underline{x}(i), \underline{u}(i), i) + \underline{\lambda}'(i+1)\left[\underline{x}(i+1) - \underline{f}(\underline{x}(i), \underline{u}(i), i)\right] \\ \qquad \overset{!}{=} \underset{\forall \underline{u}(i), \underline{\lambda}(i)}{Min} \end{array} \right.$$

With $\underline{\lambda}(i)$ being additional parameters, this unconstraint optimization problem has obviously the same answer as the previous constraint optimization problem, since:

$$\frac{\partial \widetilde{PI}}{\partial \underline{\lambda}(i)} = \phi \Rightarrow \underline{x}(i+1) = \underline{f}(\underline{x}(i), \underline{u}(i), i)$$

is the former constraint.

Let us call the solution $\underline{x}^\circ(i)$ the <u>optimal</u> solution.

Similarly: $\underline{u}^\circ(i)$, $\underline{\lambda}^\circ(i)$ are the optimal parameters.

We apply variations around the optimum:

small

$$\left|\begin{array}{l} \underline{x}(i) = \underline{x}^\circ(i) + \varepsilon \cdot \underline{\eta}(i) \\ \underline{u}(i) = \underline{u}^\circ(i) + \delta \cdot \underline{\mu}(i) \\ \underline{\lambda}(i) = \underline{\lambda}^\circ(i) + \gamma \cdot \underline{\omega}(i) \end{array}\right|$$

$$\Rightarrow \widetilde{PI} = \sum_{i=\phi}^{N-1} \varphi(\underline{x}^\circ(i) + \varepsilon \underline{\eta}(i), \underline{u}^\circ(i) + \delta \underline{\mu}(i), i)$$

$$+ (\underline{\lambda}^\circ(i+1) + \gamma \cdot \underline{\omega}(i+1))' \cdot (\underline{x}^\circ(i+1) + \varepsilon \cdot \underline{\eta}(i+1) -$$

$$\underline{f}(\underline{x}^\circ(i) + \varepsilon \cdot \underline{\eta}(i), \underline{u}^\circ(i) + \delta \underline{\mu}(i), i)$$

We expand into a Taylor series:

$$\varphi(\underline{x}^0(i) + \varepsilon \underline{y}(i), \underline{u}^0(i) + \delta \underline{\mu}(i), i)$$
$$= \varphi(\underline{x}^0(i), \underline{u}^0(i), i) + \varepsilon \underline{y}'(i) \cdot \frac{\partial \varphi}{\partial \underline{x}}$$
$$+ \delta \underline{\mu}'(i) \cdot \frac{\partial \varphi}{\partial \underline{u}}$$

etc.

Since $\widetilde{PI}$ is minimum:

$$\Rightarrow \quad \frac{\partial \widetilde{PI}}{\partial \varepsilon} = \frac{\partial \widetilde{PI}}{\partial \delta} = \frac{\partial \widetilde{PI}}{\partial \gamma} = \phi$$

if $\varepsilon = \delta = \gamma = \phi$

This leads to the discrete version of the Euler equation:

$$\boxed{\frac{\partial \varphi(k)}{\partial \underline{x}(k)} + \frac{\partial \varphi(k-1)}{\partial \underline{x}(k)} = \phi}$$

Let us call:

$$\overrightarrow{PI} = \sum_{i=\phi}^{N-1} \phi(\underline{x}(i), \underline{x}(i+1), \underline{u}(i), \underline{\lambda}(i), i) \overset{!}{=} \underset{\substack{\forall \underline{u}(i) \\ \forall \underline{\lambda}(i) \\ \forall \underline{x}(i)}}{Min}$$

where:

$$\phi(\underline{x}(i), \underline{x}(i+1), \underline{u}(i), \underline{\lambda}(i), i) = \varphi(\underline{x}(i), \underline{u}(i), i) + \underline{\lambda}'(i+1) \cdot \left[\underline{x}(i+1) - \underline{f}(\underline{x}(i), \underline{u}(i), i) \right]$$

We can apply the Euler equations three times for the three sets of parameters:

$$(1) \quad \underbrace{\frac{\partial \phi(k)}{\partial \underline{x}(k)}}_{\frac{\partial \varphi}{\partial \underline{x}} - \left[\underline{\lambda}'(k+1) \cdot \frac{\partial \underline{f}}{\partial \underline{x}}\right]'} + \underbrace{\frac{\partial \phi(k-1)}{\partial \underline{x}(k)}}_{\underline{\lambda}(k)} = \phi$$

$$\frac{\partial \varphi}{\partial \underline{x}} - \left[\underline{\lambda}'(k+1) \cdot \frac{\partial \underline{f}}{\partial \underline{x}}\right]' + \underline{\lambda}(k) = \phi$$

$$\Rightarrow \left(\frac{\partial \underline{f}}{\partial \underline{x}}\right)' \cdot \underline{\lambda}(k+1) = \underline{\lambda}(k) + \frac{\partial \varphi}{\partial \underline{x}}$$

$$\Rightarrow \quad \underline{\lambda}(k+1) = \left(\frac{\partial \underline{f}}{\partial \underline{x}}\right)'^{-1} \cdot \left[\underline{\lambda}(k) + \frac{\partial \varphi}{\partial \underline{x}}\right]$$

$\Rightarrow$ Adjugate System.

(2) $$\underbrace{\frac{\partial \Phi(k)}{\partial \underline{u}(k)}}_{\frac{\partial \varphi}{\partial \underline{u}} - \left(\frac{\partial f}{\partial \underline{u}}\right)' \cdot \underline{\lambda}(k+1)} + \underbrace{\frac{\partial \Phi(k-1)}{\partial \underline{u}(k)}}_{\phi} = \phi$$

$$\frac{\partial \varphi}{\partial \underline{u}} - \left(\frac{\partial f}{\partial \underline{u}}\right)' \cdot \underline{\lambda}(k+1) = \phi$$

$$\Rightarrow \boxed{\frac{\partial \varphi}{\partial \underline{u}} = \left(\frac{\partial f}{\partial \underline{u}}\right)' \cdot \underline{\lambda}(k+1)}$$

$\Rightarrow$ Optimality condition

(3) $$\underbrace{\frac{\partial \Phi(k)}{\partial \underline{\lambda}(k)}}_{\phi} + \underbrace{\frac{\partial \Phi(k-1)}{\partial \underline{\lambda}(k)}}_{\underline{x}(k) - \underline{f}(\underline{x}(k-1),\, \underline{u}(k-1),\, k-1)} = \phi$$

$$\Rightarrow \boxed{\underline{x}(k+1) = \underline{f}(\underline{x}(k), \underline{u}(k), k)}$$

$\Rightarrow$ state equations

This formulation was modified by Hamilton:

$$\mathcal{H}(\underline{x}(i), \underline{u}(i), \underline{\psi}(i+1), i) = \varphi(\underline{x}(i), \underline{u}(i), i) + \underline{\psi}'(i+1)\underline{f}(\underline{x}(i), \underline{u}(i), i)$$

Using $\mathcal{H}$ instead of Φ, the three conditions can be reformulated as:

$$\left|\begin{array}{lll} \dfrac{\partial \mathcal{H}(k)}{\partial \underline{x}(k)} = \underline{\psi}(k) & : & \text{adjugate system} \\ \dfrac{\partial \mathcal{H}(k)}{\partial \underline{u}(k)} = \phi & : & \text{optimality cond.} \\ \dfrac{\partial \mathcal{H}(k)}{\partial \underline{\psi}(k+1)} = \underline{x}(k+1) & : & \text{state equations} \end{array}\right.$$

Application to Linear Systems:

$$\left|\begin{array}{l} \underline{x}(k+1) = F\,\underline{x}(k) + G\,\underline{u}(k) \\ \underline{y}(k) = H\,\underline{x}(k) + I\,\underline{u}(k) \end{array}\right|$$

$$\left| PI = \underline{x}'(N)\,S\,\underline{x}(N) + \sum_{i=0}^{N-1} \left\{ \underline{x}'(i)\,Q\,\underline{x}(i) + \underline{u}'(i)\,R\,\underline{u}(i) \right\} \right|$$

where: $S \geq \phi$; $Q \geq \phi$; $R > \phi$

$$\Rightarrow \mathcal{H}(\underline{x}(i), \underline{u}(i), \underline{\psi}(i+1), i) =$$
$$\underline{x}'(i) Q \underline{x}(i) + \underline{u}'(i) R \underline{u}(i)$$
$$+ \underline{\psi}'(i+1) \cdot \left[F \underline{x}(i) + G \underline{u}(i) \right]$$

$$\Rightarrow \frac{\partial \mathcal{H}(i)}{\partial \underline{x}(i)} = 2Q\underline{x}(i) + F'\underline{\psi}(i+1) = \psi(i)$$

$$\Rightarrow \boxed{\underline{\psi}(i+1) = (F')^{-1} \cdot \left[\psi(i) - 2Q\underline{x}(i) \right]}$$

$$\frac{\partial \mathcal{H}(i)}{\partial \underline{u}(i)} = 2R\underline{u}(i) + G'\underline{\psi}(i+1) = \emptyset$$

$$\Rightarrow \quad 2R\underline{u}(i) + G'(F')^{-1}\underline{\psi}(i) - 2G(F')^{-1}Q\underline{x}(i) = \emptyset$$

$$\Rightarrow \boxed{\underline{u}(i) = \tfrac{1}{2} R^{-1} G' (F')^{-1} \left[2Q\underline{x}(i) - \underline{\psi}(i) \right]}$$

$$\frac{\partial \mathcal{H}(i)}{\partial \underline{\psi}(i+1)} = \boxed{\underline{x}(i+1) = F\underline{x}(i) + G\underline{u}(i)}$$

$$\Rightarrow \boxed{\begin{aligned} \underline{x}(i+1) = {} & \left(F + GR^{-1}G'(F')^{-1}Q\right)\underline{x}(i) \\ & - \tfrac{1}{2} G R^{-1} G' (F')^{-1} \cdot \underline{\psi}(i) \end{aligned}}$$

Let us call :

$$\underline{\xi} = \begin{bmatrix} \underline{x} \\ \underline{\psi} \end{bmatrix}$$

$$\Rightarrow \underline{\xi}(i+1) = \underbrace{\begin{bmatrix} (F+GR^{-1}G'(F')^{-1}Q) & (-\frac{1}{2}GR^{-1}G'(F')^{-1}) \\ -2(F')^{-1}Q & (F')^{-1} \end{bmatrix}}_{\text{Hamiltonian matrix}} \underline{\xi}$$

$\Rightarrow$ This is a discrete $2n \times 2n$ system. By solving it with :

$$\left| \begin{array}{rcl} \underline{x}(\phi) & = & \underline{x}_0 \\ \lim_{i \to \infty} \psi(i) & = & \phi \end{array} \right|$$

we have found yet another way to represent the same problem. We have converted the unconstraint optimization problem into a boundary value problem.

We can simplify this a little bit by letting:

$$\underline{\tilde{\psi}}(k) = \tfrac{1}{2}\underline{\psi}(k)$$

$$\Leftrightarrow \underline{\psi}(k) = 2\underline{\tilde{\psi}}(k)$$

write: $\underline{\tilde{\xi}} = \begin{bmatrix} \underline{x} \\ \underline{\tilde{\psi}} \end{bmatrix}$

$$\Rightarrow \underline{\tilde{\xi}}(k+1) = \begin{bmatrix} (F+GR^{-1}G'(F')^{-1}Q) & -GR^{-1}G'(F')^{-1} \\ -(F')^{-1}Q & (F')^{-1} \end{bmatrix} \underline{\tilde{\xi}}(k)$$

is our modified boundary value problem.

Solution: We write:

$$\underline{\tilde{\psi}}(t) = P(t)\cdot \underline{x}(t)$$

So far, we did not make any assumption since $P(t)$ can be anything.

$$\underline{x}(k+1) = \left[F + GR^{-1}G'(F')^{-1}Q\right]\underline{x}(k)$$
$$- GR^{-1}G'(F')^{-1}P\underline{x}(k)$$

$$\Rightarrow \underline{x}(k+1) = \left[F + GR^{-1}G'(F')^{-1}(Q-P)\right]\underline{x}(k)$$

$$\underline{\tilde{\psi}}(k+1) = -(F')^{-1}Q\,\underline{x}(k) + (F')^{-1}\tilde{\psi}(k)$$
$$= -(F')^{-1}Q\,\underline{x}(k) + (F')^{-1}P\,\underline{x}(k)$$
$$= -(F')^{-1}(Q-P)\,\underline{x}(k)$$
$$\equiv P(k+1)\,\underline{x}(k+1)$$

$$\Rightarrow P(k+1)\left[F + GR^{-1}G'(F')^{-1}(Q-P)\right]\underline{x}(k)$$
$$= -(F')^{-1}(Q-P)\,\underline{x}(k)$$

must be true for <u>all</u> $\underline{x}(k)$

$$\Rightarrow P(k+1)\left[F + GR^{-1}G'(F')^{-1}(Q-P)\right]$$
$$= -(F')^{-1}(Q-P)$$

In our case, P will be a <u>constant</u> matrix $\Longrightarrow$ look for steady-state:

$$P(k+1) \equiv P(k)$$

$$\Rightarrow \quad PF + PGR^{-1}G'(F')^{-1}(Q-P) = -(F')^{-1}(Q-P)$$

$$\Rightarrow \quad F'PF + F'PGR^{-1}G'(F')^{-1}(Q-P) = -(Q-P) = P-Q$$

$$\Rightarrow \quad P - F'PF - F'PGR^{-1}G'(F')^{-1}(Q-P) - Q = \emptyset$$

This can be reformulated into:

$$\boxed{P - F'PF + F'PG(R+G'PG)^{-1}GP'F - Q = \emptyset}$$

$\Rightarrow$ discrete algebraic matrix-Riccati equation. Can be solved for P.

$$\Rightarrow \quad \underline{u}(i) = \underbrace{R^{-1}G'(F')^{-1}(Q-P)}_{-K}\,\underline{x}(i)$$

$$K = -R^{-1}G'(F')^{-1}(Q-P)$$

can be rewritten as:

$$\boxed{K = (R+G'PG)^{-1}G'PF}$$

$\Rightarrow$ state feedback