Plug & Play Control: Adding Hardware to Online Control Systems

Jan Bendtsen, Jakob Stoustrup, Klaus Trangbaek and Torben Knudsen

Department of Electronic Systems
Aalborg University

Denmark

Outline

A Challenge for Model-based Control

Model-based (or high level) control ought to provide improved performance for virtually every production system in the world. Why is it not used everywhere?

- A model-based control system typically has *higher development costs* than a classical control system
- Operators may have process knowledge, but not necessarily the resources to perform complicated modeling and control design work
- An industrial process is a “living” system - the conditions for which the model based control system were designed for, might change within a fairly short time frame
- Therefore, the model based control system has to be *continuously maintained* by highly skilled engineers, or . . .
- . . . the model-based control system risks being *turned off* shortly after the first major process change

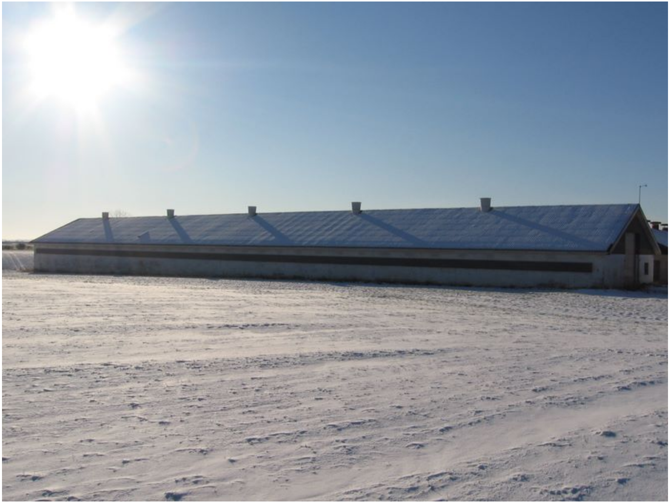
Plug-and-Play Control

Plug-and-Play Control aims to alleviate some of the aforementioned stumbling blocks for model-based control by

- *Automatically detecting* when a sensor, actuator or subsystem is added, replaced or removed
- *Automatically identifying* its relation to the existing model
- *Automatically reconfiguring* itself to utilize the new hardware

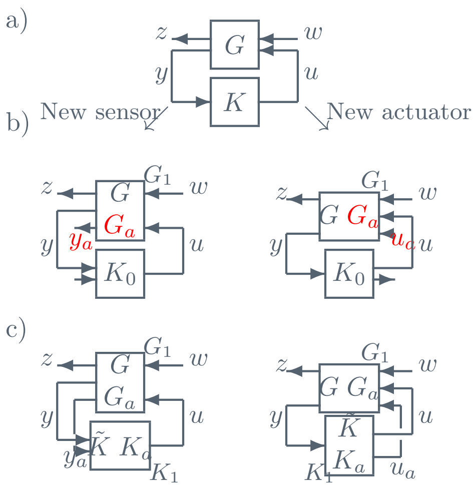
... and, if necessary, allow operators to roll back the updates to return to the previous version.

Plug-and-Play Control



Plug-and-Play Control

The Plug-and-Play Problem



Coprime factorizations

Two polynomials $m(s)$ and $n(s)$ are said to be *coprime* if their greatest common divisor is 1.
This is equivalent to the existence of two other polynomials $x(s)$ and $y(s)$ satisfying the *Bezout identity*

$$x(s)m(s) + y(s)n(s) = 1$$

This notion holds for matrices as well, but since matrix multiplication is not commutative, there is a 'left' and a 'right' version:

$$\begin{aligned} \begin{bmatrix} X_r & Y_r \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} &= I \\ \begin{bmatrix} \widetilde{M} & \widetilde{N} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \end{bmatrix} &= I \end{aligned}$$

Note that this is equivalent to requiring $\begin{bmatrix} M \\ N \end{bmatrix}$ and $\begin{bmatrix} \widetilde{M} & \widetilde{N} \end{bmatrix}$ to be invertible (in whatever space they live in).

Coprime factorizations

Given a system $G(s)$, we say that G has a *right coprime factorization* if there exist stable transfer matrices $M(s)$ and $N(s)$ such that

$$G = NM^{-1}$$

Likewise, we say that G has a *left coprime factorization* if there exist stable transfer matrices $\widetilde{M}(s)$ and $\widetilde{N}(s)$ such that

$$G = \widetilde{M}^{-1}\widetilde{N}$$

If we are lucky enough, a particular choice of $M(s), N(s), \widetilde{M}(s)$ and $\widetilde{N}(s)$ might even satisfy the so-called *Double Bezout identity*:

$$\begin{bmatrix} X_r & Y_r \\ -\widetilde{N} & \widetilde{M} \end{bmatrix} \begin{bmatrix} M & -Y_l \\ N & X_l \end{bmatrix} = I$$

Double coprime factorization, state space

Given a system $G(s)$ with stabilizable and detectable state space realization

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

Let F and L be constant matrices of appropriate dimensions such that $A+BF$ and $A+LC$ are Hurwitz, and define

$$\begin{bmatrix} M & -Y_l \\ N & X_l \end{bmatrix} = \left[\begin{array}{c|cc} A+BF & B & -L \\ \hline F & I & 0 \\ C+DF & D & I \end{array}\right]$$

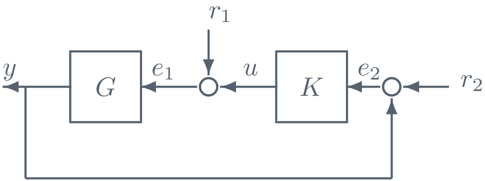
$$\begin{bmatrix} X_r & Y_r \\ -\widetilde{N} & \widetilde{M} \end{bmatrix} = \left[\begin{array}{c|cc} A+LC & -B-LD & L \\ \hline F & I & 0 \\ C & -D & I \end{array}\right]$$

Then

$$G = NM^{-1} = \widetilde{M}^{-1}\widetilde{N}$$

is a double coprime factorization.

Internal stability



This closed loop system above is *internally stable* iff all four of the following transfer functions are stable:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} (I - KG)^{-1} & (I - KG)^{-1}K \\ (I - GK)^{-1}G & (I - GK)^{-1} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

(Note that both G and K must have the same number of in- and outputs.)

Lemma 5.10 in Zhou et. al.

Given a system/controller interconnection with coprime factorizations $G = NM^{-1} = \widetilde{M}^{-1}\widetilde{N}$ and $K = UV^{-1} = \widetilde{V}^{-1}\widetilde{U}$. Then the following statements are equivalent:

1. The feedback system is internally stable.
2. $\begin{bmatrix} M & U \\ N & V \end{bmatrix}$ is invertible in $\mathcal{RH}_\infty$.
3. $\begin{bmatrix} \widetilde{V} & -\widetilde{U} \\ -\widetilde{N} & \widetilde{M} \end{bmatrix}$ is invertible in $\mathcal{RH}_\infty$.
4. $\widetilde{M}V - \widetilde{N}U$ is invertible in $\mathcal{RH}_\infty$.
5. $\widetilde{V}M - \widetilde{U}N$ is invertible in $\mathcal{RH}_\infty$.

Plant/controller factorization

Given a system

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

and a stabilizing observer-based controller

$$K = \left[\begin{array}{c|c} A + BF + LC + LDF & -L \\ \hline F & 0 \end{array}\right]$$

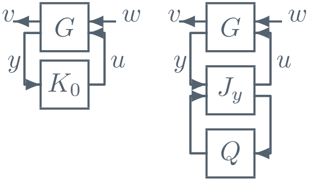
Then

$$\begin{bmatrix} M & U \\ N & V \end{bmatrix} = \left[\begin{array}{c|cc} A + BF & B & -L \\ \hline F & I & 0 \\ C + DF & D & I \end{array}\right]$$

$$\begin{bmatrix} \widetilde{V} & -\widetilde{U} \\ -\widetilde{N} & \widetilde{M} \end{bmatrix} = \left[\begin{array}{c|cc} A + LC & -B - LD & L \\ \hline F & I & 0 \\ C & -D & I \end{array}\right]$$

is a double coprime factorization.

The Youla-Kucera parameterization of all stabilizing controllers



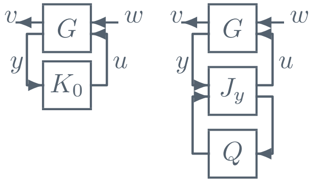
Consider the control loop in the left part of the figure and assume that the controller K_0 stabilizes the system G. Factorize the lower right part of G and K_0 as

$$G_{yu} = NM^{-1} = \tilde{M}^{-1}\tilde{N} \qquad and \qquad K_0 = UV^{-1} = \tilde{V}^{-1}\tilde{U}$$

with $N, M, \tilde{M}, \tilde{N} \in \mathcal{RH}_\infty$, and $U, V, \tilde{U}, \tilde{V} \in \mathcal{RH}_\infty$, with the factors satisfying

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = \begin{bmatrix} M & U \\ N & V \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

The Youla-Kucera parameterization of all stabilizing controllers



All stabilizing controllers for G can now be parameterized according to the Youla-Kucera parameterization

$$K(Q) \;=\; \mathcal{F}_l\,(J_y, Q) \;=\; K_0 \,+\, \tilde{V}^{-1}Q(I \,+\, V^{-1}NQ)^{-1}V^{-1},$$

with $Q \in \mathcal{RH}_\infty$, i.e.,

- $\mathcal{F}_l(G, K(Q)$ is stable for any stable Q
- for *any* stabilizing controller K_i, a stable Q exists so that $K(Q) = K_i$.

The Youla-Kucera parameterization of all stabilizing controllers

More explicitly,

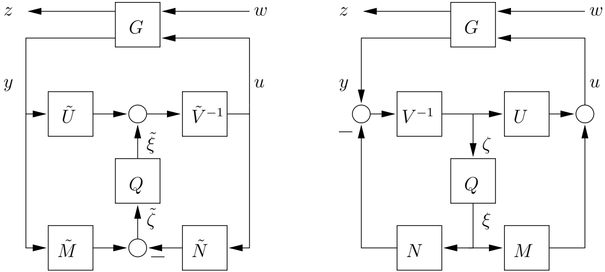
$$
\begin{aligned}
K(Q) &= (U + MQ)(V + NQ)^{-1} = (\widetilde{V} + Q\widetilde{N})^{-1}(\widetilde{U} + Q\widetilde{M}) \\
&= \mathcal{F}_l(J_y, Q)
\end{aligned}
$$

where

$$
J_y = \begin{bmatrix} UV^{-1} & \widetilde{V}^{-1} \\ V^{-1} & -V^{-1}N \end{bmatrix}
$$

The Youla-Kucera parameterization of all stabilizing controllers

The Youla-Kucera parameterization can be implemented in this fashion:



$$K(Q) = U(Q)V(Q)^{-1} = \widetilde{V}(Q)^{-1}\widetilde{U}(Q)$$

where

$$\begin{aligned} U(Q) &= U + MQ\,, \quad V(Q) = V + NQ, \\ \widetilde{U}(Q) &= \widetilde{U} + Q\widetilde{M}\,, \quad \widetilde{V}(Q) = \widetilde{V} + Q\widetilde{N}\,, \quad Q \in \mathcal{RH}_\infty \end{aligned}$$

Phasing in new controls

Consider a plant with some stabilizing, but conservative controller K_0. After some time of operation, the plant parameters are identified better than the original guess

Once the new parameters are identified, a new controller may be found using appropriate methods, e.g., by solving an LQG design problem or similar.

It will then be required to switch from K_0 to K_1, in a *bumpless* manner and *without losing stability.*

Recall that all stabilizing controllers for G can be constructed as

$$K(Q) = \mathcal{F}_l(\mathcal{K}, Q) = K_0 + \tilde{V}^{-1}Q(I + V^{-1}NQ)^{-1}V^{-1},$$

with $Q \in \mathcal{RH}_\infty$, i.e., the closed loop is stable for any stable Q.

Note further that if $Q \in \mathcal{RH}_\infty$ then so is γQ for all $\gamma \in [0; 1]$.

Phasing in new controls

Factorize as usual:

$$\begin{aligned} G_{yu} &= NM^{-1} = \tilde{M}^{-1}\tilde{N} \\ K_0 &= U_0 V_0^{-1} = \tilde{V}_0^{-1}\tilde{U}_0 \\ K_1 &= U_1 V_1^{-1} = \tilde{V}_1^{-1}\tilde{U}_1 \end{aligned}$$

and compute

$$\begin{bmatrix} \tilde{V}_i & -\tilde{U}_i \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U_i \\ N & V_i \end{bmatrix} = \begin{bmatrix} M & U_i \\ N & V_i \end{bmatrix} \begin{bmatrix} \tilde{V}_i & -\tilde{U}_i \\ -\tilde{N} & \tilde{M} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

for $i = 0, 1$. *Note that the plant factors $N, M, \tilde{N}$ and $\tilde{M}$ must be the same in both identities!*

Phasing in new controls

The Q that transforms K_0 into K_1 is given by

$$Q_1 = \widetilde{U}_1 V_0 - \widetilde{V}_1 U_0 = \widetilde{V}_1 (K_1 - K_0) V_0$$

The transition from K_0 to K_1 is achieved by way of

$$K(\gamma Q) = (U_0 + M\gamma Q)(V_0 + N\gamma Q)^{-1}$$

by gradually varying γ from 0 to 1.
Notice that it is always possible to roll back to the original controller by dialing γ back to 0.

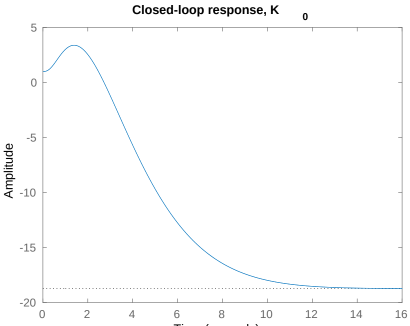
Example

Consider the open-loop unstable system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \qquad y = x_1$$

A (rather poor) initial stabilizing observer-based regulator is given by

$$L = \begin{bmatrix} -10^{-13} \\ -160 \end{bmatrix}, \qquad F = \begin{bmatrix} -4.29 & 6 \end{bmatrix}$$



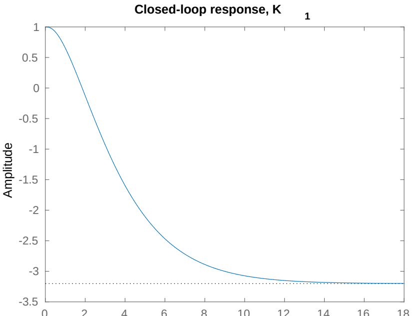
Example

An LQG regulator design with performance measure

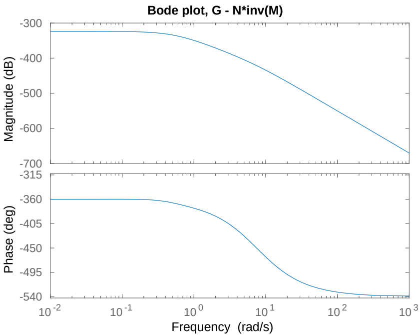
$$J = \int_0^\infty y^T y + u^T u dt$$

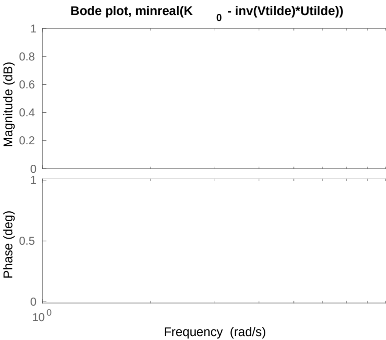
(assuming fictitious noise covariance $0.1I$ for both measurements and states) results in

$$L_1 = \begin{bmatrix} 10.7 \\ -5.68 \end{bmatrix}, \qquad F_1 = \begin{bmatrix} -8.12 & -1.14 \end{bmatrix}$$



Example



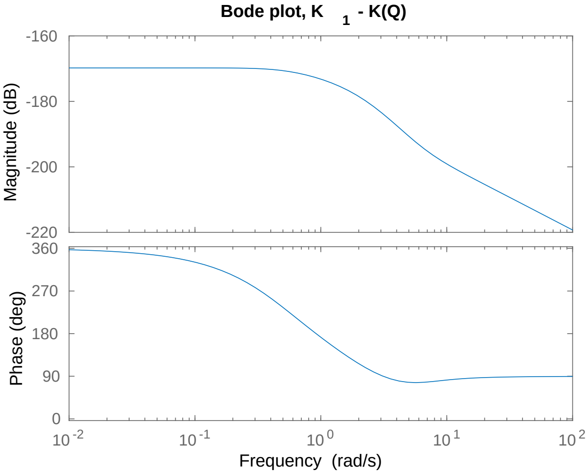


Sometimes Matlab cannot figure out what a zero system is, so you may have to help it by invoking `minreal`

Example


```
 %% Transition from K0 to K1 via Youla parameter
Utilde1 = ss(A + L1*C, -L1, F1, 0);
Vtilde1 = ss(A + L1*C, -B - L1*D, F1, eye(m));
U1 = ss(A + B*F1, -L1, F1, 0);
V1 = ss(A + B*F1, -L1, C + D*F1, eye(p));
tildeFactors = inv([M U1; N V1]);
Vtilde1 = tildeFactors(1:m,1:m);
Q = Vtilde1*(K1 - K0)*V;
KQ = (U + M*Q)*inv(V + N*Q);
```

Example


Bode plot, K 1 - K(Q)
Magnitude (dB)
-160
-180
-200
-220
Phase (deg)
360
270
180
90
0
10 -2
10 -1
10 0
10 1
10 2
Frequency (rad/s)

Conclusions

- Plug-and-Play control was discussed for LTI plants
- New sensors, actuators and subsystems are identified by appropriate system identification algorithms
- Once a new component has been identified, it may be included in the existing control system, preferably in an *automatic* and *reversible* manner
- We discussed *coprime factorizations* and the *Youla-Kucera factorization of stabilizing controllers* and saw how to use it to transition between stabilizing observer-based controllers without losing stability.