Shaping the Dynamic Response

METR 4202: Advanced Control & **Robotics**

Dr Surya Singh -- Lecture # 12

October 22, 2014

metr4202@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~metr4202/

Schedule

Week	Date	Lecture (W: 11:10-12:40, 24-402)
1	30-Jul	Introduction
2	6-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	13-Aug	Robot Kinematics (& Ekka Day)
4	20-Aug	Robot Dynamics & Control
5	27-Aug	Robot Motion
6	3-Sep	Robot Sensing: Perception & Multiple View Geometry
7	10-Sep	Robot Sensing: Features & Detection using Computer Vision
8	17-Sep	Navigation (+ Prof. M. Srinivasan)
9	24-Sep	Localization & Motion Planning + Control
	1-Oct	*Study break*
10	8-Oct	**State-Space Modelling**
11	15-Oct	Shaping the Dynamic Response
12	22-Oct	Linear Observers & LQR
13	29-Oct	Applications in Industry & Course Review

Announcements: Lab 3 Extended

- **Lab 3:**
  - Due Nov 3 or Nov 12
  - Time Signup online
  - Rubric online (soon)

- Cool Robotics Share Site
  ➔ http://metr4202.tumblr.com/
  Twitter: **#metr4202**

Cool Robotics Video Share

Shaping of Dynamic Responses

Let's Generalize This

- Shaping the Dynamic Response
  - A method of designing a control system for a process in which all the state variables are accessible for
  - measurement-the method known as *pole-placement*

- Theory:
  - We will find that in a controllable system, with all the state variables accessible for measurement, it is possible to place the closed-loop poles anywhere we wish in the complex *s* plane!
- Practice:
  - Unfortunately, however, what can be attained in principle may not be attainable in practice. Speeding the response of a sluggish system requires the use of large control signals which the actuator (or power supply) may not be capable of delivering. And, control system gains are **very sensitive** to the location of the open-loop poles

Regulator Design

- Here the problem is to determine the gain matrix G in a linear feedback law $u = -Gx - G_0 x_0$
  - Where: x_0 is the vector of exogenous variables. The reason it is necessary to separate the exogenous variables from the process state x, rather than deal directly with the metastate $\mathbf{x} = \begin{bmatrix} x \\ x_0 \end{bmatrix}$ is that we must assume that the underlying process is controllable.
    - Since the exogenous variables are not true state variables, but additional inputs that cannot be affected by the control action, they cannot be included in the state vector when using a design method that requires controllability.
    - HOWEVER, they can be used in a process for Observability! ∴ when we are doing this as part of the sensing/mapping process!!

^

- The assumption that all the state variables are accessible to measurement in the regulator means that the gain matrix G in is permitted to be any function of the state **x** that the design method requires

$$y = Cx$$
$$u = -G_y y$$
$$u = -G\hat{x}$$

  – Where: $\hat{x}$ is the state of an appropriate dynamic system known as an "observer."

SISO Regulator Design

- Design of a gain matrix

$$G = g' = [g_1, g_2, \ldots, g_k]$$

for the single-input, single-output system

$$\dot{x} = Ax + Bu$$

where

$$B = b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}$$

With the control law $u = -Gx = -g'x$ (6.7) becomes

$$\dot{x} = (A - bg')x$$

- Our objective is to find the matrix G = g' which places the poles of the closed-loop dynamics matrix $A_c = A - bg'$ at the locations desired.

SISO Regulator Design [2]

- One way of determining the gains would be to set up the characteristic polynomial for *Ac:*

$$|sI - A_c| = |sI - A + bg'| = s^k + \bar{a}_1 s^{k-1} + \cdots + \bar{a}_k$$

- The coefficients $a_1, a_2, \ldots, a_k$ of the powers of *s* in the characteristic polynomial will be functions of the *k* unknown gains. Equating these functions to the numerical values desired for $a_1, a_2, \ldots, a_k$ will result in *k* simultaneous equations the solution of which will yield the desired gains $g_1, \ldots, g_k$.

SISO Regulator Design [3]

If the original system is in the companion form given in (3.90), the task is particularly easy, because

$$A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{k-1} & -a_k \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{6.11}$$

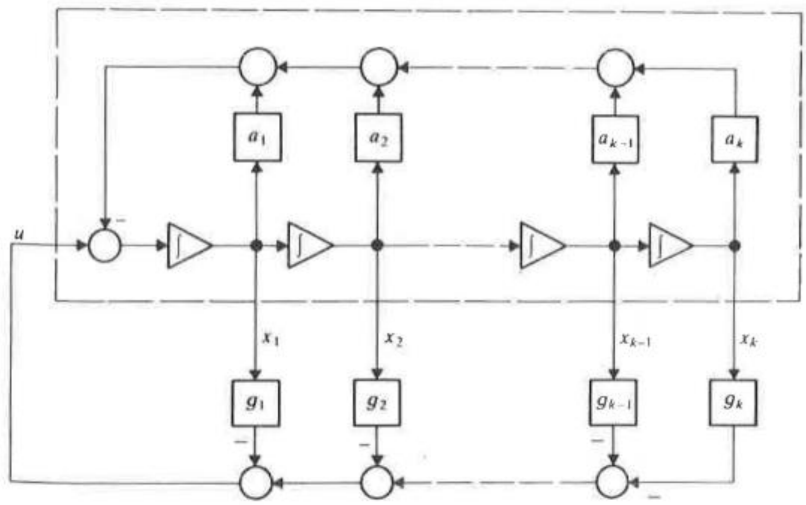
$$bg' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [g_1, g_2, \ldots, g_k] = \begin{bmatrix} g_1 & g_2 & \cdots & g_k \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Hence

$$A_c = A - bg' = \begin{bmatrix} -a_1 - g_1 & -a_2 - g_2 & \cdots & -a_k - g_k \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

The gains $g_1, \ldots, g_k$ are simply added to the coefficients of the open-loop A matrix to give the closed-loop matrix A_c. This is also evident from the block-diagram representation of the closed-loop system as shown in Fig. 6.1.

SISO Regulator Design [4]



SISO Regulator Design [4]

- But how to get this in companion form?

$$\bar{x} = Tx \tag{6.14}$$

Then, as shown in Chap. 3,

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{b}u \tag{6.15}$$

where

$$\bar{A} = TAT^{-1} \quad \text{and} \quad \bar{b} = Tb$$

For the transformed system the gain matrix is

$$\bar{g} = \hat{a} - \bar{a} = \hat{a} - a \tag{6.16}$$

since $\bar{a} = a$ (the characteristic equation being invariant under a change of state variables). The desired control law in the original system is

$$u = -g'x = -g'T^{-1}\bar{x} = -\bar{g}'\bar{x} \tag{6.17}$$

From (6.17) we see that

$$\bar{g}' = g'T^{-1}$$

Thus the gain in the original system is

$$g = T'\bar{g} = T'(\hat{a} - a) \tag{6.18}$$

SISO Regulator Design [5]

In words, the desired gain matrix for a general system is the difference between the coefficient vectors of the desired and actual characteristic equation, premultiplied by the inverse of the transpose of the matrix T that transforms the general system into the companion form of (3.90), the A matrix of which has the form (6.11).

The desired matrix T is obtained as the product of two matrices U and V:

$$T = VU \tag{6.19}$$

The first of these matrices transforms the original system into an intermediate system

$$\dot{\bar{x}} = \bar{A}\bar{x} \tag{6.20}$$

in the second companion form (3.107) and the second transformation U transforms the intermediate system into the first companion form.

Consider the intermediate system

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{b}u \tag{6.21}$$

with $\bar{A}$ and $\bar{b}$ in the form of (3.107). Then we must have

$$\bar{A} = UAU^{-1} \qquad \text{and} \qquad \bar{b} = Ub \tag{6.22}$$

SISO Regulator Design [6]

The desired matrix U is precisely the inverse of the controllability test matrix Q of Sec. 5.4. To prove this fact, we must show that

$$U^{-1}\bar{A} = AU^{-1} \tag{6.23}$$

or

$$Q\bar{A} = AQ \tag{6.24}$$

Now, for a single-input system

$$Q = [b, Ab, \ldots, A^{k-1}b]$$

Thus, with $\bar{A}$ given by (3.107), the left-hand side of (6.23) is

$$Q\bar{A} = [b, Ab, \ldots, A^{k-1}b]\begin{bmatrix} 0 & 0 & \cdots & -a_k \\ 1 & 0 & \cdots & -a_{k-1} \\ 0 & 1 & \cdots & -a_{k-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -a_1 \end{bmatrix}$$

$$= [Ab, A^2b, \ldots, A^{k-1}b, -a_kb - a_{k-1}Ab - \cdots - a_kA^{k-1}b] \tag{6.25}$$

The last term in (6.25) is

$$(-a_kI - a_{k-1}A - \cdots - a_kA^{k-1})b \tag{6.26}$$

SISO Regulator Design [7]

Now, by the Cayley-Hamilton theorem, (see Appendix):

$$A^k = -a_1 A^{k-1} - a_2 A^{k-2} - \cdots - a_k I$$

so (6.26) is $A^k b$. Thus the left-hand side of (6.24) as given by (6.25) is

$$Q\tilde{A} = [Ab, A^2 b, \ldots, A^k b] = A[b, Ab, \ldots, A^{k-1} b] = AQ$$

which is the desired result.

If the system is not controllable, then Q^{-1} does not exist and there is no general method of transforming the original system into the intermediate system (6.21); in fact it is not possible to place the closed-loop poles anywhere one desires. Thus, controllability is an essential requirement of system design by pole placement. If the system is *stabilizable* (i.e., the uncontrollable part is asymptotically stable, as discussed in Chap. 5) a stable closed-loop system can be achieved by placing the poles of the controllable subsystem where one wishes and accepting the pole locations of the uncontrollable subsystem. In order to apply the formula of this section, it is necessary to first separate the uncontrollable subsystem from the controllable subsystem.

The control matrix $\tilde{b}$ of the intermediate system is given by

$$\tilde{b} = Ub \tag{6.27}$$

We now show that

$$\tilde{b} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.28}$$

SISO Regulator Design [8]

Multiply (6.28) by Q to obtain

$$Q\tilde{b} = [b, Ab, \ldots, A^{k-1} b] \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = b$$

which is the same as (6.27), since $Q^{-1} = U$.

The final step is to find the matrix V that transforms the intermediate system (6.21) into the final system (6.15). We must have

$$\bar{x} = V\tilde{x} \tag{6.29}$$

For the transformation (6.28) to hold, we must have

$$\bar{A} = V\tilde{A}V^{-1}$$

or

$$V^{-1}\bar{A} = \tilde{A}V^{-1} \tag{6.30}$$

SISO Regulator Design [9]

The matrix V^{-1} that satisfies (6.30) is the transpose of the upper left-hand k-by-k submatrix of the (triangular Toeplitz) matrix appearing in (3.103)

$$V^{-1} = \begin{bmatrix} 1 & a_1 & \cdots & a_{k-1} \\ 0 & 1 & \cdots & a_{k-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = W \qquad (6.31)$$

To prove this, we note that the left-hand side of (6.30) is

$$V^{-1}\bar{A} = \begin{bmatrix} 1 & a_1 & \cdots & a_{k-1} \\ 0 & 1 & \cdots & a_{k-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_k \\ 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_k \\ 1 & a_1 & \cdots & a_{k-2} & 0 \\ 0 & 1 & \cdots & a_{k-3} & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \qquad (6.32)$$

(Note that the zeros in the first row of $V^{-1}\bar{A}$ are the result of the difference of

SISO Regulator Design [10]

two terms $a_1 - a_1$, $a_2 - a_2$, etc.) and the right-hand side of (6.30) is

$$\tilde{A}V^{-1} = \begin{bmatrix} 0 & 0 & \cdots & -a_k \\ 1 & 0 & \cdots & -a_{k-1} \\ 0 & 1 & \cdots & -a_{k-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -a_k \end{bmatrix} \begin{bmatrix} 1 & a_1 & \cdots & a_{k-1} \\ 0 & 1 & \cdots & a_{k-2} \\ 0 & 0 & \cdots & a_{k-3} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_k \\ 1 & a_1 & \cdots & a_{k-2} & 0 \\ 0 & 1 & \cdots & a_{k-3} & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

which is the same as (6.32). Thus (6.30) is proved.

We also need

$$\bar{b} = V\tilde{b}$$

We will show that

$$\bar{b} = \tilde{b}$$

Consider

$$\tilde{b} = V^{-1}\bar{b}$$

with

$$b = V^{-1}\bar{b} = \begin{bmatrix} 1 & a_1 & \cdots & a_{k-1} \\ 0 & 1 & \cdots & a_{k-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

SISO Regulator Design [11]

Thus $\tilde{b}$ and $\bar{b}$ are the same.

The result of this calculation is that the transformation matrix T whose transpose is needed in (6.18) is the inverse of the product of the controllability test matrix and the triangular matrix (6.31).

The above results may be summarized as follows. The desired gain matrix g, by (6.18) and (6.19), is given by

$$g = (VU)'(\hat{a} - a) \tag{6.33}$$

where

$$V = W^{-1} \quad \text{and} \quad U = Q^{-1}$$

Thus

$$VU = W^{-1}Q^{-1} = (QW)^{-1}$$

Viewing State-Space as a Tool for Solving ODEs Simultaneously

State Space as an ODE

- The basic mathematical model for an LTI system consists of the state differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \qquad \mathbf{x}(t_0) = \mathbf{x}_0$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

- The solution is can be expressed as a sum of terms owing to the initial state and to the input respectively:

$$x(t) = \underbrace{e^{at}x_0}_{\text{zero-input response}} + \underbrace{\int_0^t e^{a(t-\tau)}bu(\tau)d\tau}_{\text{zero-state response}} \qquad y(t) = ce^{at}x_0 + \int_0^t ce^{a(t-\tau)}bu(\tau)d\tau + du(t)$$

- This is a first-order solution similar to what we expect

State Equation Solution: Matrix Exponential

$$x(t) = \boxed{e^{at}x_0} + \int_0^t e^{a(t-\tau)}bu(\tau)d\tau \qquad y(t) = ce^{at}x_0 + \int_0^t ce^{a(t-\tau)}bu(\tau)d\tau + du(t)$$

- The first term can be handled via a Taylor Series

$$e^{\mathbf{A}(t-t_0)} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k(t-t_0)^k = \mathbf{I} + \mathbf{A}(t-t_0) + \frac{1}{2}\mathbf{A}^2(t-t_0)^2 + \frac{1}{6}\mathbf{A}^3(t-t_0)^3 + \ldots$$

→ This case is known as the matrix exponential function
→ Also referred to as the state-transition matrix, denoted by Φ (t, t_0):

$$\mathbf{x}(t) = \mathbf{\Phi}(t)\mathbf{x}_0 + \int_{t_0}^t \mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau$$

- The state-transition matrix satisfies the homogeneous state equation, thus, it represents the free response of the system. That is, it governs the response that is excited by the initial conditions only

Output Equation Solution

- Having the solution for the complete state response, a solution for the complete output equation can be obtained as:

$$\mathbf{y}(t) = \underbrace{\mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0}_{\text{zero-input response: } y_{zi}(t)} + \underbrace{\int_{t_0}^{t} \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \mathbf{D}\mathbf{u}(t)}_{y_{zs}(t)\text{: zero-state response}}$$

State Equation Solution

- Thus, the solution to the unforced system (u=0):

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} \phi_{11}(t) & \cdots & \phi_{11}(t) \\ \phi_{21}(t) & \cdots & \phi_{n2}(t) \\ \vdots & & \vdots \\ \phi_{n1}(t) & \cdots & \phi_{nn}(t) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \end{bmatrix}$$

Note: the term $\phi_{ij}(t)$ can be interpreted as the response of the i^{th} state variable due to an initial condition on the j^{th} state variable when there are zero initial conditions on all other states.

- The solution of the state differential equation can also be obtained using the Laplace transform:

$$L[\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)]$$

$$L[\dot{\mathbf{x}}(t)] = L[\mathbf{A}\mathbf{x}(t)] + L[\mathbf{B}\mathbf{u}(t)]$$

$$s\mathbf{X}(s) - \mathbf{x}_0 = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s)$$

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}_0 + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s)$$

$$\downarrow$$

$$\mathbf{X}(s) = \mathbf{\Phi}(s)\mathbf{x}_0 + \mathbf{\Phi}(s)\mathbf{B}\mathbf{U}(s)$$

$$\Rightarrow \quad L[\mathbf{\Phi}(t)] = \mathbf{\Phi}(s) = [s\mathbf{I} - \mathbf{A}]^{-1} \longrightarrow \mathbf{\Phi}(t) = L^{-1}[s\mathbf{I} - \mathbf{A}]^{-1}$$

Properties of the Matrix Exponential

- Note that $e^{\mathbf{A}t}$ is just a notation used to represent a power series.

$$e^{\mathbf{A}t} \neq [e^{a_{ij}t}]$$

- Example 1: Consider the following 4x4 matrix

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Let's obtain the first terms of the power series:

$$A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad A^3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \qquad A^4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad A^k = 0 \quad \forall k \geq 4$$

The power series contains only a finite number of nonzero terms:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2}\mathbf{A}^2t^2 + \frac{1}{6}\mathbf{A}^3t^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -t & 1 & 0 & 0 \\ \frac{1}{2}t^2 & -t & 1 & 0 \\ -\frac{1}{6}t^3 & \frac{1}{2}t^2 & -t & 1 \end{bmatrix} \neq \left[e^{a_{ij}t}\right] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ e^{-t} & 0 & 0 & 0 \\ 0 & e^{-t} & 0 & 0 \\ 0 & 0 & e^{-t} & 0 \end{bmatrix}$$

Properties of the matrix exponential

- For any real nxn matrix **A**, the matrix exponential $e^{\mathbf{A}t}$ satisfies:

1. $e^{\mathbf{A}t}$ is the unique matrix for which: $\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t} \qquad e^{\mathbf{A}t}\big|_{t=0} = \mathbf{I}(nxn)$
2. For any t_1 and t_2: $e^{\mathbf{A}(t_1+t_2)} = e^{\mathbf{A}t_1}e^{\mathbf{A}t_2}$

   As a consequence: $e^{\mathbf{A}(0)} = e^{\mathbf{A}(t-t)} = e^{\mathbf{A}t}e^{-\mathbf{A}t} = \mathbf{I}$

   Thus, $e^{\mathbf{A}t}$ is invertible for all t, being the inverse: $\left[e^{\mathbf{A}t}\right]^{-1} = e^{-\mathbf{A}t}$
3. For all t, A and $e^{\mathbf{A}t}$ commute with respect to matrix product: $\mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$
4. For all t: $\left[e^{\mathbf{A}t}\right]^T = e^{\mathbf{A}^T t}$
5. For any real nxn matrix B, $e^{(\mathbf{A}+\mathbf{B})t} = e^{\mathbf{A}t}e^{\mathbf{B}t}$ for all t if and only if **AB=BA**
6. Finally, a useful property of the matrix exponential is that it can be reduced to a finite power series involving n scalar analytic functions α_j(t)

$$e^{\mathbf{A}t} = \sum_{k=0}^{n-1} \alpha_k(t)\mathbf{A}^k$$

Using this to Solve State Space Problems

- Example:
  - Solve the following linear second-order ordinary differential

    $\ddot{y}(t)+7\dot{y}(t)+12y(t)=u(t)$
  - Consider the input u(t) is a step of magnitude 3 and the initial conditions $\dot{y}(0)=0.05 \quad y(0)=0.10$

Example: (Back To) Robot Arms

Slides 17-27 Source: R. Lindeke, ME 4135, “Introduction to Control”

Remembering the Motion Models:

- Recall from Dynamics, the Required Joint Torque is:

$$\tau_i = D_i\left(q\right)\ddot{q}_i + C_i\left(q, \dot{q}_i\right) + h\left(q\right) + b\left(\dot{q}_i\right)$$

Dynamical Manipulator Inertial Tensor – a function of position and acceleration

Coupled joint effects (centrifugal and coriolis) issues due to multiple moving joints

Gravitational Effects

Frictional Effect due to Joint/Link movement

Lets simplify the model

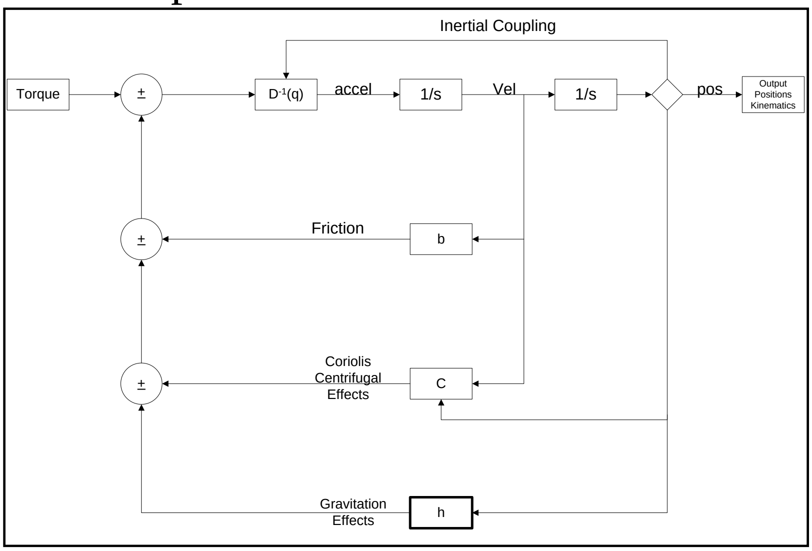
- This torque model is a 2nd order one (in position) lets look at it as a velocity model rather than positional one then it becomes a system of highly coupled 1st order differential equations
- We will then isolate Acceleration terms (acceleration is the 1st derivative of velocity)

$$a = \dot{v} = \ddot{q} = D_i^{-1}(q)\left(\tau_i - C_i(q, \dot{q}_i) - h(q) - b\left(\dot{q}_1\right)\right)$$

Considering Control:

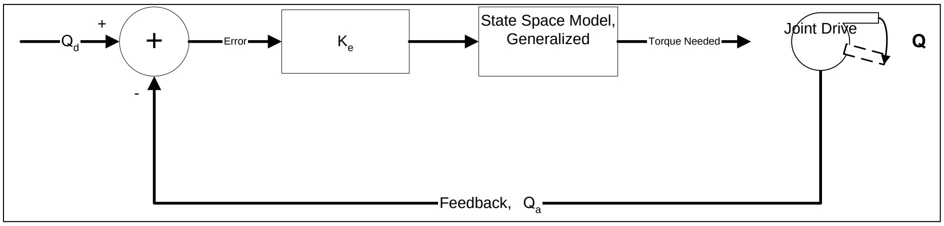
- Each Link's torque is influenced by each other links motion
  - We say that the links are highly coupled
- Solution then suggests that control should come from a simultaneous solution of these torques
- We will model the solution as a "State Space" design and try to balance the torque-in with *positional control*-out – the most common way it is done!
  - But we could also use 'force control' to solve the control problem!

The State-Space Control Model:



Setting up a Real Control

- We will (start) by using positional error to drive our torque devices



- This simple model is called a PE (proportional error) controller

PE Controller:

- To a 1st approximation, $\tau = K_m{*}I$
  - Torque is proportional to <u>motor current</u>

- And the Torque required is a function of 'Inertial' (Acceleration) and 'Friction' (velocity) effects as suggested by our L-E models

$$\tau_m \simeq J_{eq}\ddot{q} + F_{eq}\dot{q}$$

→ Which can be approximated as:

$$K_m I_m = J_{eq}\ddot{q} + F_{eq}\dot{q}$$

Setting up a "Control Law"

- We will use the positional error (as drawn in the state model) to develop our torque control
- We say then for PE control:

$$\tau \propto k_{pe}(\theta_d - \theta_a)$$

- Here, k_{pe} is a "gain" term that guarantees sufficient current will be generated to develop appropriate torque based on observed positional error

Using this Control Type:

- It is a representation of the physical system of a mass on a spring!
- We say after setting our target as a 'zero goal' that:

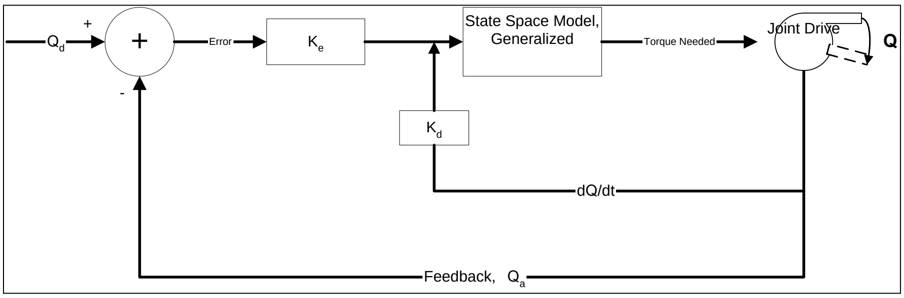
$$-k_{pe} * \theta_a = J\ddot{\theta} + F\dot{\theta}$$

the solution of which is:

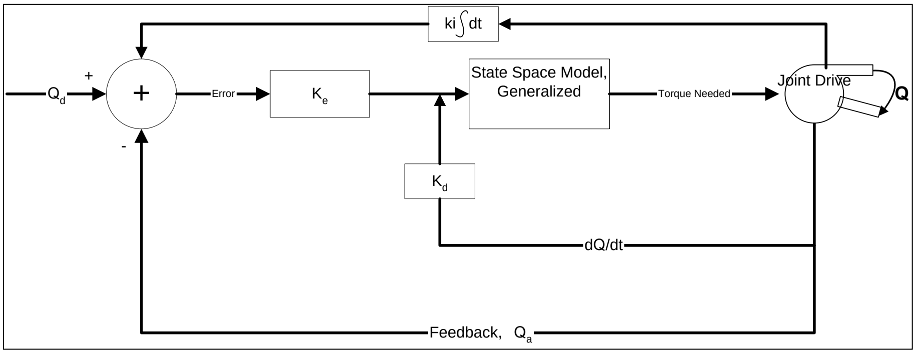
θ_a is a function of the servo feedback as a function of time!

$$\Big(\theta_a = e^{-\left({}^{F}/_{2J}\right)t}\left[C_1 e^{\left({}^{1}/_{2}\right)\omega t} + C_2 e^{-\left({}^{1}/_{2}\right)\omega t}\right]$$

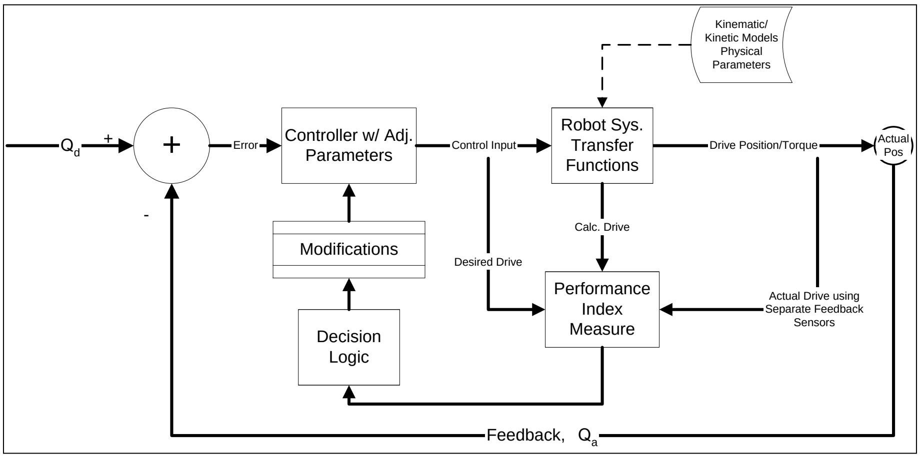
State Space Model of PD:



PID State Space Model:



State Model of Adjustable Controller



Pole Placement

Pole Assignment by State Feedback

- We begin by examining the problem of closed-loop pole assignment. For the moment, we make a simplifying assumption that all of the system states are measured. We will remove this assumption later. We will also assume that the system is completely controllable. The following result then shows that the closed-loop poles of the system can be arbitrarily assigned by feeding back the state through a suitably chosen constant-gain vector.

- Lemma 18.1: Consider the state space nominal model
- Let $\bar{r}(t)$ denote an external signal.

$$\dot{x}(t) = \mathbf{A_o}x(t) + \mathbf{B_o}u(t)$$
$$y(t) = \mathbf{C_o}x(t)$$

- Then, provided that the pair (A0, B0) is completely controllable, there exists

$$u(t) = \bar{r} - \mathbf{K}x(t)$$

$$\mathbf{K} \triangleq [k_0, k_1, \ldots, k_{n-1}]$$

- such that the closed-loop characteristic polynomial is $A_{cl}(s)$, where $A_{cl}(s)$ is an arbitrary polynomial of degree n.

- Note that state feedback does not introduce additional dynamics in the loop, because the scheme is based only on proportional feedback of certain system variables. We can easily determine the overall transfer function from $\bar{r}(t)$ to y(t). It is given by

$$\frac{Y(s)}{\overline{R}(s)} = \mathbf{C_o}(s\mathbf{I} - \mathbf{A_o} + \mathbf{B_o}\mathbf{K})^{-1}\mathbf{B_o} = \frac{\mathbf{C_o}Adj\{s\mathbf{I} - \mathbf{A_o} + \mathbf{B_o}\mathbf{K}\}\mathbf{B_o}}{F(s)}$$

- where

$$F(s) \triangleq \det\{s\mathbf{I} - \mathbf{A_o} + \mathbf{B_o}\mathbf{K}\}$$

- and Adj stands for adjoint matrices.

[Matrix inversion lemma]

- We can further simplify the expression given above. To do this, we will need to use the following results from Linear Algebra.
- (Matrix inversion lemma).
  Consider three matrices A,B,C
  Then, if A + BC is nonsingular, we have that

$$(\mathbf{A}+\mathbf{BC})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{I}+\mathbf{CA}^{-1}\mathbf{B}\right)^{-1}\mathbf{CA}^{-1}$$

- In the case for which B = g $\in \mathbb{R}^n$ and C^T = h $\in \mathbb{R}^n$, the above result becomes

$$\left(\mathbf{A}+gh^T\right)^{-1}=\left(\mathbf{I}-\mathbf{A}^{-1}\frac{gh^T}{1+h^T\mathbf{A}^{-1}g}\right)\mathbf{A}^{-1}$$

- Lemma 18.3: Given a matrix W $\in \mathbb{R}^{n\times n}$ and a pair of arbitrary vectors $\phi_1 \in \mathbb{R}^n$ and $\phi_2 \in \mathbb{R}^n$, then provided that W and are nonsingular,

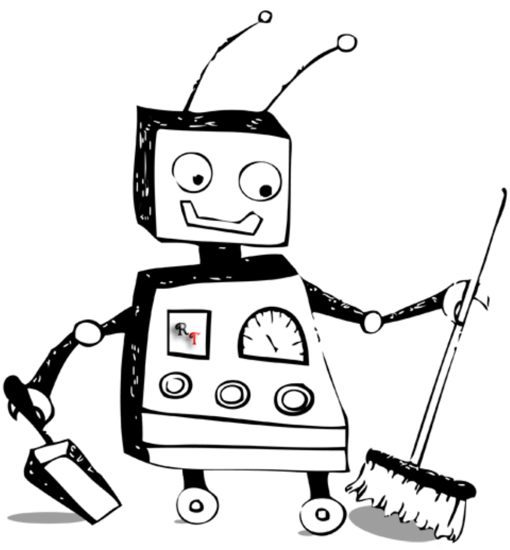
$$W+\phi_1\phi_2^T\ ,$$

- Proof: See the l

$$Adj(W+\phi_1\phi_2^T)\phi_1 = Adj(W)\phi_1$$

$$\phi_2^T Adj(W+\phi_1\phi_2^T) = \phi_2^T Adj(W)$$

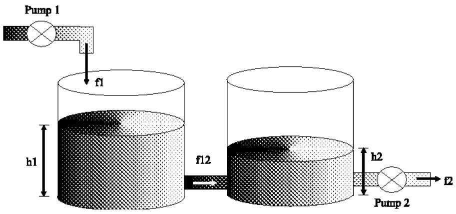
SECaT Time! … Brought To You By the Number 5



Example

Example

- Figure 18.1: Schematic diagram of two coupled tanks



- Water flows into the first tank through pump 1 a rate fi(t) that obviously affects the head of water in tank 1 (denoted by h1(t)). Water flows out of tank 1 into tank 2 at a rate f12(t), affecting both h1(t) and h2(t). Water than flows out of tank 2 at a rate fe controlled by pump 2.
- Given this information, the challenge is to build a virtual sensor (or observer) to estimate the height of liquid in tank 1 from measurements of the height of liquid in tank 2 and the flows f1(t) and f2(t).

- Before we continue with the observer design, we first make a model of the system. The height of liquid in tank 1 can be described by the equation

$$\frac{dh_1(t)}{dt} = \frac{1}{A}(f_i(t) - f_{12}(t))$$

- Similarly, h2(t) is described by

$$\frac{dh_2(t)}{dt} = \frac{1}{A}(f_{12}(t) - f_e)$$

- The flow between the two tanks can be approximated by the free-fall velocity for the difference in height between the two tanks:

$$f_{12}(t) = \sqrt{2g(h_1(t) - h_2(t))}$$

- We can linearize this model for a nominal steady-state height difference (or operating point). Let
- This yields the following linear model:

$$h_1(t) - h_2(t) = \Delta h(t) = H + h_d(t)$$

- where

$$\frac{d}{dt}\begin{bmatrix} h_1(t) \\ h_2(t) \end{bmatrix} = \begin{bmatrix} -k & k \\ k & -k \end{bmatrix} \begin{bmatrix} h_1(t) \\ h_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} f_1(t) - \frac{K\sqrt{H}}{2} \\ f_2(t) + \frac{K\sqrt{H}}{2} \end{bmatrix}$$

$$k = \frac{K}{2\sqrt{H}}$$

- We are assuming that h2(t) can be measured and h1(t) cannot, so we set C = [0 1] and D = [0 0]. The resulting system is both controllable and observable (as you can easily verify). Now we wish to design an observer

$$J = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$$

- to estimate the value of h2(t). The characteristic polynomial of the observer is readily seen to be

$$s^2 + (2k + J_1)s + J_2 k + J_1 k$$

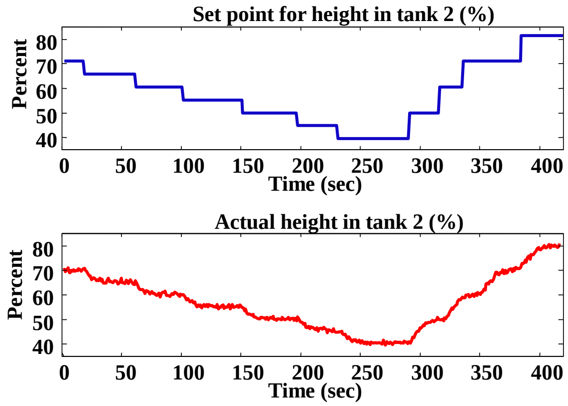
- so we can choose the observer poles; that choice gives us values for J1 and J2.

- If we assume that the operating point is H = 10%, then k = 0.0411. If we wanted poles at s = -0.9291 and s = -0.0531, then we would calculate that J1 = 0.3 and J2 = 0.9. If we wanted two poles at s = -2, then J2 = 3.9178 and J1 = 93.41.

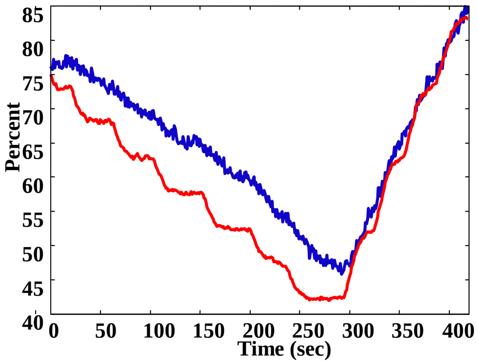
- The equation for the final observer is then

$$\frac{d}{dt}\begin{bmatrix}\hat{h}_1(t)\\ \hat{h}_2(t)\end{bmatrix} = \begin{bmatrix}-k & k\\ k & -k\end{bmatrix}\begin{bmatrix}\hat{h}_1(t)\\ \hat{h}_2(t)\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}f_1(t) - \frac{K\sqrt{H}}{2}\\ f_2(t) + \frac{K\sqrt{H}}{2}\end{bmatrix} + J(h_2(t) - \hat{h}_2(t))$$

- The data below has been collected from the real system shown earlier



- The performance of the observer for tank height is compared below with the true tank height which is actually measured on this system.



Actual height in tank 1 (*blue*), Observed height in tank 1 (*red*)

State-Feedback Control Objectives

- Regulation: Force state x to equilibrium state (usually 0) with a desirable dynamic response.
- Tracking: Force the output of the system y to tracks a given desired output y_d with a desirable dynamic response.

Pole Placement Problem as an Eignenvalue Problem

Choose the state feedback gain to place the poles of the closed-loop system, i.e.,

$$\text{Eigenvalues of } \bar{G} := G - HK$$

At specified locations $\lambda_1^{des}, \ldots, \lambda_n^{des}$

State Feedback Control of a System in CCF

Consider a SISO system in CCF: $\hat{\mathbf{x}}(k+1) = \mathbf{G}_c\hat{\mathbf{x}}(k) + \mathbf{H}_c\mathbf{u}$

$$\mathbf{G}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & -a_2 & -a_1 \end{bmatrix}, \mathbf{H}_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\Phi(s) = |zI - \mathbf{G}| = z^n + a_1 z^{n-1} + \cdots + a_{n-1}z + a_n$$

State Feedback Control

$$\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{r}, \quad \mathbf{K} = \begin{bmatrix} k_1 & \cdots & k_{n1} \end{bmatrix}$$

Closed-Loop CCF System

Closed loop A matrix:

$$\bar{\mathbf{G}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & -a_2 & -a_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & \cdots & k_n \end{bmatrix}$$

$$\bar{\mathbf{G}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_n + k_1 & -a_{n-1} + k_2 & \cdots & -a_2 + k_{n-1} & -a_1 + k_n \end{bmatrix}$$

Choosing the Gain-CCF

Closed-loop Characteristic Equation

$$\Phi(z) = z^n + (a_1 + k_n)z^{n-1} + \cdots + (a_{n-1} + k_2)z + (a_n + k_1)$$

Desired Characteristic Equation:

$$\Phi^{des}(z) = \prod_{i=1}^{n}\left(z - \lambda_i^{des}\right) = z^n + a_1^{des} z^{n-1} + \cdots + a_{n-1}^{des} z + a_n^{des}$$

Control Gains:

$$\boxed{\boldsymbol{K_i = a_{n-i+1}^{des} - a_{n-i+1}, \quad i = 1,2,\ldots,n}}$$

Transformation to CCF

Transform system $\dot{\mathbf{x}} = \mathbf{Gx} + \mathbf{Hu}$ To CCF

$$\hat{\mathbf{x}}^+ = \mathbf{G}_c\hat{\mathbf{x}} + \mathbf{H}_c\mathbf{u} \Rightarrow \begin{cases} \hat{x}_1{}^+ = \hat{x}_2 \\ \hat{x}_2{}^+ = \hat{x}_3 \\ \vdots \\ \hat{x}_n{}^+ = -a_n\hat{x}_1 - a_{n-1}\hat{x}_2 - \cdots - a_1\hat{x}_n + u \end{cases}$$

Where $x^+(k)=x(k+1)$ (for simplicity)

First, find how new state z_1 is related to x:

$$\hat{x}_1 = \mathbf{px}, \quad \mathbf{p} = [p_1 \quad \cdots \quad p_n] \quad \text{(row vector)}$$

Transformed State Equations

<u>Necessary Conditions for p</u>:

$$\begin{cases} \hat{x}_1^{\,+} = \mathbf{p}\mathbf{x}^+ = \mathbf{pGx} + \cancelto{0}{\mathbf{pH}}u = \hat{x}_2 \\ \hat{x}_2^{\,+} = \mathbf{pGx}^+ = \mathbf{pG}^2\mathbf{x} + \cancelto{0}{\mathbf{pGH}}u = \hat{x}_3 \\ \vdots \\ \hat{x}_{n-1}^{\,+} = \mathbf{pG}^{n-2}\mathbf{x}^+ = \mathbf{pG}^{n-1}\mathbf{x} + \cancelto{0}{\mathbf{pG}^{n-2}\mathbf{H}}u = \hat{x}_n \\ \hat{x}_n^{\,+} = \mathbf{pG}^{n-1}\mathbf{x}^+ = \mathbf{pG}^n\mathbf{x} + \cancelto{1}{\mathbf{pG}^{n-1}\mathbf{H}}u \end{cases}$$

$$\mathbf{p}\begin{bmatrix}\mathbf{H} & \mathbf{GH} & \cdots & \mathbf{G}^{n-1}\mathbf{H}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

Vector p can be found if the system is <u>controllable</u>: $\boxed{\mathbf{p} = \mathbf{e}_n^{\,T}\mathbf{M}^{-1}}$

State Transformation Invertibility

State transformation: $\begin{bmatrix}\hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n\end{bmatrix} = \mathbf{Tx} = \begin{bmatrix}\mathbf{p} \\ \mathbf{pG} \\ \vdots \\ \mathbf{pG}^{n-1}\end{bmatrix}\mathbf{x}$

Matrix T is invertible since

$$\begin{bmatrix}\mathbf{p} \\ \mathbf{pG} \\ \vdots \\ \mathbf{pG}^{n-1}\end{bmatrix}\begin{bmatrix}\mathbf{H} & \mathbf{GH} & \cdots & \mathbf{G}^{n-1}\mathbf{H}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \cdots & \mathbf{1} & \mathbf{pG}^n\mathbf{H} \\ \vdots & ⋰ & ⋰ & \vdots \\ \mathbf{1} & \mathbf{pG}^n\mathbf{H} & \cdots & \mathbf{pG}^{2n-2}\mathbf{H}\end{bmatrix}$$

By the Cayley-Hamilton theorem.

Toeplitz Matrix

The Cayley-Hamilton theorem can further be used to show that

$$\mathbf{TM}\begin{bmatrix} a_{n-1} & \cdots & a_1 & 1 \\ a_{n-2} & \cdots & 1 & 0 \\ \vdots & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} = \mathbf{I}$$

Matrix on the right is called Toeplitz matrix

State Transformation Formulas

Formula 1:

$$\mathbf{T} = \begin{bmatrix} \mathbf{p} \\ \mathbf{pG} \\ \vdots \\ \mathbf{pG}^{n-1} \end{bmatrix}, \quad \mathbf{p} = \mathbf{e}_n^{\,T}\mathbf{M}^{-1}$$

Formula 2:

$$\mathbf{T} = \left(\mathbf{M}\begin{bmatrix} a_{n-1} & \cdots & a_1 & 1 \\ a_{n-2} & \cdots & 1 & 0 \\ \vdots & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}\right)^{-1}$$

State Feedback Control Gain Selection

$$\mathbf{u} = -\hat{\mathbf{K}}\hat{\mathbf{x}} + \mathbf{r}, \quad \hat{\mathbf{K}} = \begin{bmatrix} a_n^{des} - a_n - & \cdots & a_1^{des} - a_1 \end{bmatrix}$$

$$\mathbf{u} = -\underbrace{\hat{\mathbf{K}}\mathbf{T}}_{\mathbf{K}} x + \mathbf{r} \Rightarrow K = \begin{bmatrix} a_n^{des} - a_n - & \cdots & a_1^{des} - a_1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{pG} \\ \vdots \\ \mathbf{pG}^{n-1} \end{bmatrix}$$

By Cayley Hamilton: $\mathbf{a}_n \mathbf{I} + \mathbf{a}_{n-1}\mathbf{G} + \cdots + \mathbf{a}_1 \mathbf{G}^{n-1} = -\mathbf{G}^n$

$$\mathbf{K} = \mathbf{p}\left(\mathbf{G}^n + \mathbf{a}_1^{des}\mathbf{G}^{n-1} + \cdots + \mathbf{a}_{n-1}^{des}\mathbf{G} + \mathbf{a}_n^{des}\mathbf{I}\right)$$ or

$$\boxed{\mathbf{K} = \mathbf{e}_n{}^T \mathbf{M}^{-1} \mathbf{\Phi}^{des}(\mathbf{G})}$$

Bass-Gura Formula

$$\mathbf{u} = -\hat{\mathbf{K}}\hat{\mathbf{x}} + \mathbf{r}, \quad \hat{\mathbf{K}} = \begin{bmatrix} a_n^{des} - a_n - & \cdots & a_1^{des} - a_1 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} a_n - a_n^{des} \\ a_{n-1} - a_{n-1}^{des} \\ \vdots \\ a_1 - a_1^{des} \end{bmatrix}^T \left(\mathbf{M} \begin{bmatrix} a_{n-1} & \cdots & a_1 & 1 \\ a_{n-2} & \cdots & 1 & 0 \\ \vdots & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \right)^{-1}$$

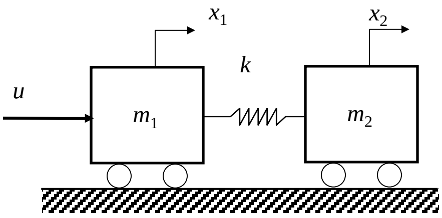
$$\boxed{\mathbf{K} = \begin{bmatrix} a_1^{des} - a_1 \\ \vdots \\ a_{n-1}^{des} - a_{n-1} \\ a_n^{des} - a_n \end{bmatrix}^T \left(\mathbf{M} \begin{bmatrix} 1 & a_1 & \cdots & a_{n-1} \\ 0 & 1 & \cdots & a_{n-2} \\ \vdots & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \right)^{-1}}$$

Double Integrator-Matlab Solution

```
T=0.5; lam=[0;0];
G=[1 T;0 1]; H=[T^2/2;T]; C=[1 0];
K=acker(G,H,lam);
Gcl=G-H*K;
clsys=ss(Gcl,H,C,0,T);
step(clsys);
```


Flexible System Example

Consider the linear mass-spring system shown below:



Parameters:

m_1=m_2=1Kg.
K=50 N/m

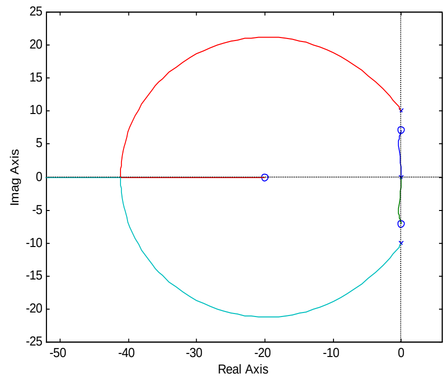
- Analyze PD controller based on a)x_1, b)x_2
- Design state feedback controller, place poles at $-20, -20, 5\sqrt{2}(-1 \pm j)$

Collocated Control

Transfer Function: $G_p = \frac{X_1(s)}{U(s)} = \frac{s^2 + 50}{s^2(s^2 + 100)}$

PD Control: $G_c = K(s + a), \quad a = 20$

Root-Locus



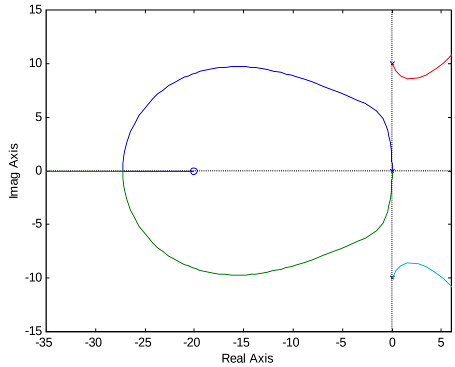
Non-Collocated Control

Transfer Function: $G_p = \frac{X_2(s)}{U(s)} = \frac{50}{s^2(s^2 + 100)}$

PD Control: $G_c = K(s + a), \quad a = 20$

Root-Locu

Unstable



State Model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -50 & 50 & 0 & 0 \\ 50 & -50 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u$$

<u>Discretized Model</u>: x(k+1)=Gx(k)+Hu(k)

$$G = \begin{bmatrix} 0.9975 & 0.0025 & 0.01 & 0 \\ 0.0025 & 0.9975 & 0 & 0.01 \\ -0.4992 & 0.4992 & 0.9975 & 0.0025 \\ 0.4992 & 0.4992 & 0.0025 & 0.9975 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 0 \\ 0.01 \\ 0 \end{bmatrix}$$

Open-Loop System Information

Controllability matrix:

$$\mathbf{M} = \left[\mathbf{H} \quad \mathbf{GH} \quad \mathbf{G(GH)} \quad \mathbf{G(G^2H)} \right]$$

$$\mathbf{M} = \begin{bmatrix} 0 & 0.0001 & 0.0002 & 0.0003 \\ 0 & 0 & 0 & 0 \\ 0.01 & 0.0099 & 0.0098 & 0.0097 \\ 0 & 0.001 & 0.0002 & 0.003 \end{bmatrix}$$

Characteristic equation:

$|zI-G|=(z-1)^2(z^2-1.99z+1)=z^4-3.99z^3+5.98z^2-3.99z+6$

State Feedback Controller

Characteristic Equations:

$|zI-G|=(z-1)^2(z^2-1.99z+1)=z^4-3.99z^3+5.98z^2-3.99z+6$

$$\Phi^{des}(s)=(z-0.8187)^2\left((z-0.9294)^2+0.0658^2\right)$$
$$\Phi^{des}(s)=z^4-3.4963z^3+4.5822z^2-2.6675z+0.5819$$

$$K=\begin{bmatrix}-3.4963+3.99\\4.5822-5.98\\-2.6675+3.99\\0.5819-6\end{bmatrix}^T\left(M\begin{bmatrix}1&-3.99&5.98&-3.99\\0&1&-3.99&5.98\\0&0&1&-3.99\\0&0&0&-3.99\end{bmatrix}\right)^{-1}$$

$$\boxed{\begin{aligned}&K=[757.00\quad -144.17\quad 45.54\quad 105.75]\\&u=-757x_1+144.17x_2-45.54x_3-105.75x_4+r\end{aligned}}$$

Matlab Solution

```
%System Matrices
m1=1; m2=1; k=50; T=0.01;
syst=ss(A,B,C,D);
A=[0 0 1 0;0 0 0 1;-50 50 0 0;50 -50 0 0];
B=[0; 0; 1; 0];
C=[1 0 0 0;0 1 0 0]; D=zeros(2,1);
cplant=ss(A,B,C,D);

%Discrete-Time Plant
plant=c2d(cplant,T);
[G,H,C,D]=ssdata(plant);
```

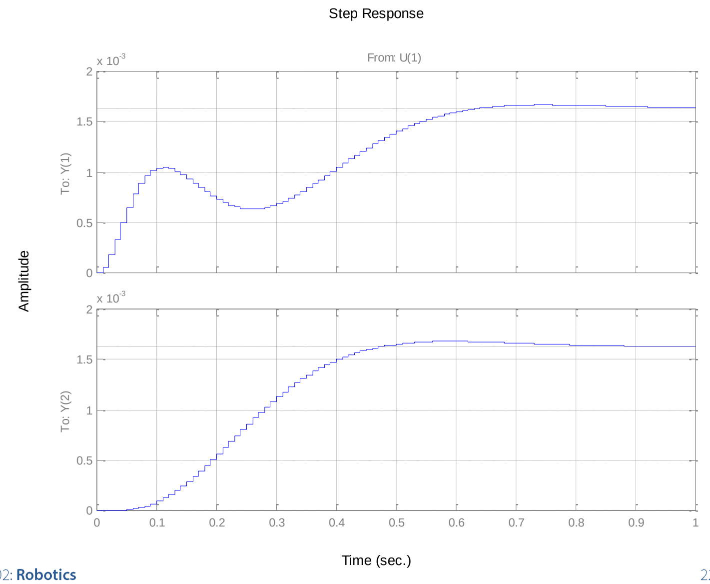
Matlab Solution

```
%Desired Close-Loop Poles
pc=[-20;-20;-5*sqrt(2)*(1+j);-
5*sqrt(2)*(1-j)];
pd=exp(T*pc);

% State Feedback Controller
K=acker(G,H,pd);

%Closed-Loop System
clsys=ss(G-H*K,H,C,0,T);
grid
step(clsys,1)
```


Time Respone



Steady-State Gain

Closed-loop system: $x(k+1)=G_{cl}x(k)+Hr(k)$, $Y=Cx(k)$

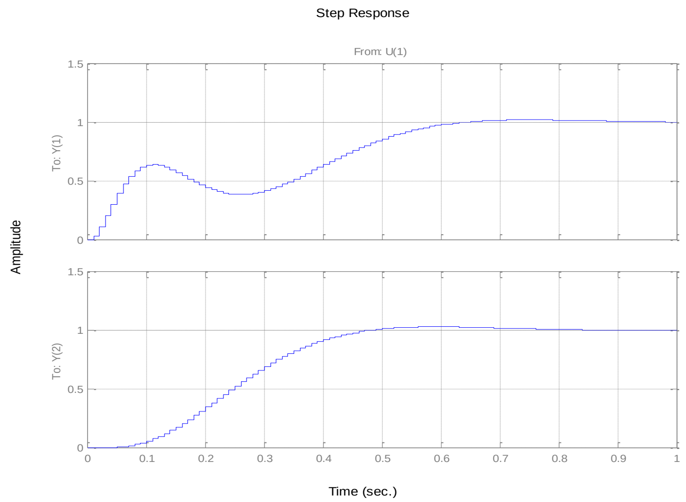
$Y(z)=C(zI-G_{cl})^{-1}H\ R(z)$

If $r(k)=r.1(k)$ then $y_{ss}=C(I-G_{cl})^{-1}H$

Thus if the desired output is constant

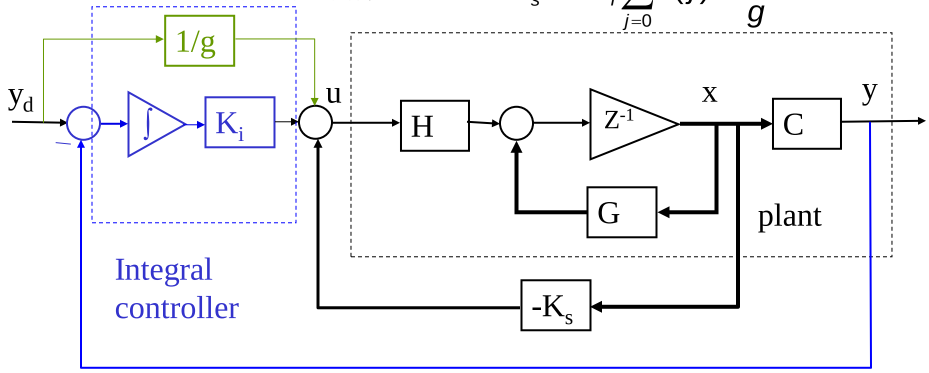
$r=y_d/\text{gain}$, $\text{gain}= C(I-G_{cl})^{-1}H$

Time Response



Integral Control

Control law: $$\mathbf{u} = -\mathbf{K}_s\mathbf{x} - \mathbf{K}_I\sum_{j=0}^{k-1} e(j) + \frac{y_d}{g}$$



Automatically generates reference input r!

Closed-Loop Integral Control System

Plant: $$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

Control: $\mathbf{u} = \mathbf{r} - \mathbf{K}_s\mathbf{x} - \mathbf{K}_I v(k),\ \mathbf{e} = \mathbf{y}_d - \mathbf{y}$

Integral state: $\mathbf{v}(k+1) = \mathbf{v}(k) - \mathbf{e}(k)$

Closed-loop system

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{v}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{C} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{v}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{H} \\ \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{K}_s & \mathbf{K}_I \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{H}\boldsymbol{r} \\ -\boldsymbol{y}_d \end{bmatrix}$$

Double Integrator-Matlab Solution

```
T=0.5; lam=[0;0;0];
G=[1 T;0 1]; H=[T^2/2;T]; C=[1 0];
Gbar=[G zeros(2,1);C 1];
Hbar=[H;0];
K=acker(Gbar,Hbar,lam);
Gcl=Gbar-Hbar*K;
yd=1; r=0; %unknown gain
clsys=ss(Gcl,[H*r;-yd],[C 0;K],0,T);
step(clsys);
```


Closed-Loop Step Response