Controllability Analysis and Matlab Software

M.S. Govatsmark
T. Larsson
S. Skogestad

Department of Chemical Engineering
Norwegian University of Science and Technology

Outline

1. Matlab and Toolboxes
2. Controllability Analysis
3. Distillation Column Example

1. Matlab Software for Control

- Simulation
  - Ode suite
  - Simulink
- Analysis and Design
  - **Control System Toolbox**
  - **Robust Control Toolbox**
  - μ**-Analysis and Synthesis Toolbox**
  - Optimization Toolbox
  - Model Predictive Control Toolbox
- Other relevant tools
  - System Identification Toolbox
  - Signal Processing Toolbox
  - LMI Control Toolbox
  - Polynomial Toolbox
  - QFT Toolbox

See Matwork's homepage (www.mathworks.com) for additional information

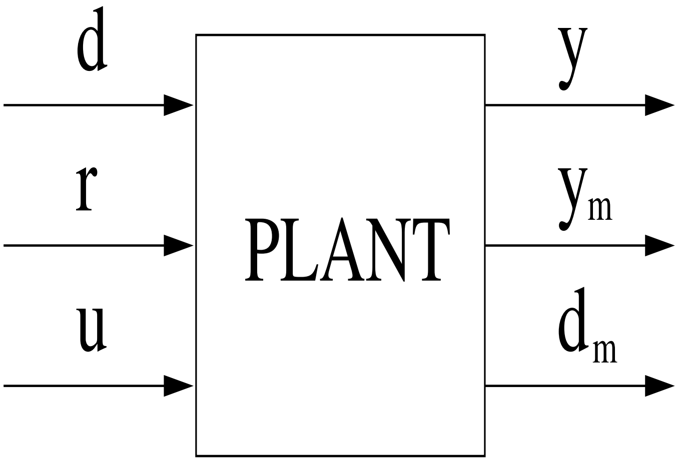
- **Control System Toolbox**
  For modeling, analysis, and design of control systems.
  - Several LTI model types supported
  - Can include time delay
  - Model order reduction
  - Model dynamics (poles, zeros)
  - Frequency response (bode, sigma)
  - LQG design, pole placement
  - Linear simulation (step, lsim)
- **Robust Control Toolbox**
  For the analysis and design of robust multi-variable control systems. Use the matrix functions in Control toolbox.
  - Robust model reduction
  - H_2 and H_∞ design
  - LQG and LQG loop transfer recovery
  - μ-synthesis
- **μ Toolbox**
  For performing robust control design:
  - Model reduction
  - H_2 and H_∞ design
  - μ-synthesis
  - sysic

Robust and μ toolbox looks like competing products.

2. What is Controllability ?

The ability to achieve acceptable control performance for a system.

Controllability is independent of the controller and is only a property of the plant (or process) alone.



More exactly:

To keep the the outputs (y) within specified bounds or displacements from their references (r), in spite of unknown, but known bounded variations, such as disturbances (d) and plant changes, using the available inputs (u) and available measurements (y_m and d_m).

Controllable system:

There exists a controller (connecting the plant measurements and plant inputs) that yields acceptable performance for all the expected plant variations.

Why do Controllability Analysis?

To find out what a control performance can be expected of a system.

Try to answer the following three question:

1. **How well can the plant be controlled?**
   - How easy the plant actually is to control
2. **What control structure should be used?**
   - What variable to measure / control
   - Which variable to manipulate
   - How are the variables best paired together
3. **How might the process be changed to improve the control?**
   - Relax output specifications
   - Replace or move actuators
   - Extra measurements
   - Add extra equipment to damp disturbances
   - Change plant dynamics and time delay

How to do Controllability Analysis

1. **"Simulation approach"**
   - Performance is assessed by exhaustive simulations, which require a specific controller design and specific values of disturbances and set-point changes.
   - Not know if a fundamental property to the plant
2. **"Rigorous approach"**
   - Mathematically formulation of the control objective, disturbances, model uncertainty and then synthesis controllers to see if the objectives are met (e.g. H_∞-, H_2- or even better: μ-controller).
   - Difficult and time consuming
3. **"Simple tools approach"**
   - Use controllability tools based on scaled models of $G(j\omega)$ and $G_d(j\omega)$ to get a rough idea of how easy the is to control (PRESENTED HERE)
   - Only a linear tool

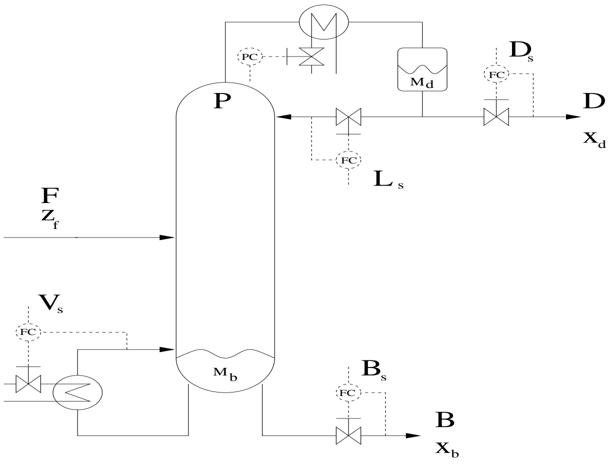
A Procedure for Controllability Analysis

Basis: Minimal realization of a linear, time-invariant, scaled model

1. **Check functional controllability ($rank(G(s) = n_y$)**
   - Ability to control the outputs independently
2. **Compute the poles and zeros and associated directions**
   - Large RHP-poles far from the origin is bad.
   - Bad with small, pinned RHP-zeros close to the origin.
3. **Compute the frequency response for $G_d(j\omega)$**
   - Get a lower bandwidth limit (ω_d) for control
4. **Compute steady-state and frequency response of $G(j\omega)$ and the RGA-matrix**
   - RGA-elements close to the crossover frequency is critical.
   - Large RGA-elements may expect sensitivity to uncertainty.
5. **Compute the singular values of $G(j\omega)$ with associated directions**
   - Impossible with independent control of outputs when $\underline{\sigma}(G(j\omega)) < 1$
6. **Compute the magnitude to elements in $G^H G_d$**
   - Input saturation if the elements are bigger than 1
7. **Compatible requirements with respect to RHP-poles, RHP-zeros and disturbances?**
   - E.g. require that $|y_z g_d(z)| \leq 1$ for each disturbance and RHP-zeros.
8. **Compute condition number**
   - Small condition number indicate no problem with uncertainty
9. **Compute PRGA and CLDG if interested in decentralized control**

3. Example
"simple tools" - controllability analysis

(www.chembio.ntnu.no/users/skoge/book/matlab_m/cola/cola.html)



No.	Manipulated variables	Nominal value	Allowed variation
u_1	Reflux flow (L)	2.7063	1
u_2	Boilup flow (V)	3.2063	1
u_3	Distillate flow (D)	0.5	1
u_4	Bottom product flow (B)	0.5	1

No.	Disturbance	Nominal value	Expected variation
d_1	Feed flow (F)	0.5	0.2
d_2	Feed composition (z_f)	1.0	0.1

No.	Controlled variables	Nominal value	Accepted deviation
y_1	Distillate composition (x_D)	0.99	0.01
y_2	Bottom prod. composition (x_B)	0.01	0.01
y_3	Condenser holdup (M_D)	0.5	0.5
y_4	Boiler holdup (M_B)	0.5	0.5

Linearized, Scaled Model

$$\begin{bmatrix} x_D \\ x_B \\ M_D \\ M_B \end{bmatrix} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} L \\ V \\ F \\ z_F \\ D \\ B \end{bmatrix} \tag{1}$$

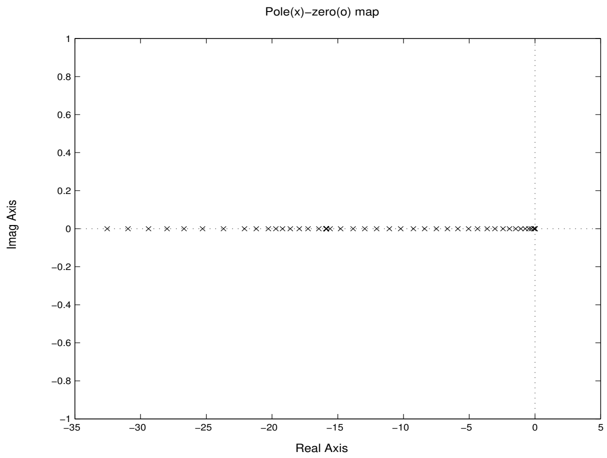
A, B, C, D: A given state space model.

Matlab:

```
>> sysc = ss(A,B,C,D);     %Control
>> sysr = mksys(A,B,C,D); %Robust
>> sysr = mksys(A,B(:,1:4),B(:,5:6),...
C(1:2,:),C(3:4,:), ...
D(1:2,1:4),D(1:2,5:6), ...
D(3:4,1:4),D(3:4,5:6),'tss'); %Robust - ``two ports''
>> sysm = pck(A,B,C,D);    %Mu
>>
>> size(sysc)         %Control
State-space model with 4 outputs, 6 inputs, and 82 states.
```

Robust control toolbox allows for “two ports”.

Poles with Associated Directions



Poles ($p = \lambda, (A - \lambda I)x = 0$):

$$p^T = \begin{bmatrix} 0 & 0 & -0.0052 & -0.0830 & .. \end{bmatrix} \tag{2}$$

⋆ **Necessary to stabilize poles in origo.**

Pole output directions ($y_p = Cx, (A - \lambda I)x = 0$):

$$p: \; 0 \;\; 0 \;\; -0.0052 \;\; -0.0830 \;\; .... \tag{3}$$

$$y_p = \begin{bmatrix} 0 & 0 & -0.6270 & -0.7793 & .... \\ 0 & 0 & -0.7790 & 0.6266 & .... \\ 0 & 1 & 0 & 0 & .... \\ 1 & 0 & 0 & 0 & .... \end{bmatrix} \begin{array}{c} x_D \\ x_B \\ M_D \\ M_B \end{array} \tag{4}$$

⋆ **The condenser holdup (M_D) and the reboiler holdup (M_B) must be stabilized / controlled.**

Pole input directions ($u_p = B^T x, (A^T - \lambda I)x = 0$):

$$p: \quad 0 \quad 0 \quad -0.0052 \quad -0.0830 \quad ....$$

$$u_p = \begin{array}{cc} \begin{bmatrix} 0.5735 & -0.5774 & 0.7056 & 0.1012 & .... \\ -0.5735 & 0.5774 & -0.7012 & -0.7618 & .... \\ 0.1147 & 0 & 0.0703 & 0.6360 & .... \\ 0 & 0 & 0.0744 & 0.0698 & .... \\ 0 & -0.5774 & 0 & 0 & .... \\ -0.5735 & 0 & 0 & 0 & .... \end{bmatrix} & \begin{array}{c} L \\ V \\ F \\ z_F \\ D \\ B \end{array} \end{array} \tag{5}$$

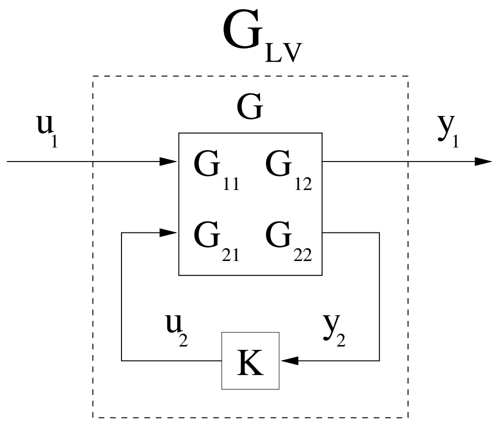
⋆ **The distillate flow (D) may be used to control the condenser holdup (M_D)**

⋆ **The bottom product flow (B) may be used to control the reboiler holdup (M_B).**

Matlab:

```
>> pole(sysc));'
>> [R DD1] = eig(A);  yp    = C*R;
>> [L DD2] = eig(A'); up    = B'*L;
>> ypn=yp'./(sqrt(sum(yp.*yp)')*ones(1,4));
>> upn=up'./(sqrt(sum(up.*up)')*ones(1,6));
```

System Manipulation 1

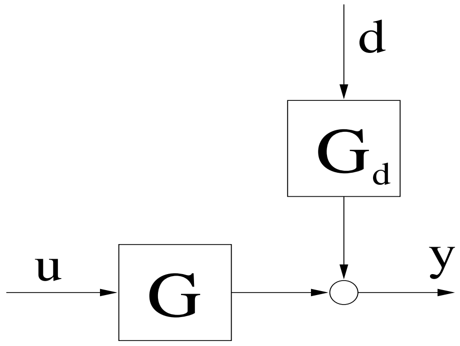


Level control:

$$G_{LV} = G_{11} - G_{12}K(I - G_{22}K)^{-1}G_{21} = LFT(G, -K)$$

System Manipulation 2

Split the inputs in manipulable variables (u) and disturbances (d):
$y = Gu + G_d d$



⋆ **Stable plant with no RHP-zeros.**

Matlab:

```
>> K=diag([-10, -10]); sysm = starp(sysm,-K,2,2);
```

Steady State Gain and RGA

Relative gain at steady state:

$$G(0) = \begin{bmatrix} 87.5403 & -86.1756 \\ 108.4596 & -109.8244 \end{bmatrix} \tag{6}$$

Disturbance gain at steady state:

$$G_d(0) = \begin{bmatrix} 7.8789 & 8.8125 \\ 11.7211 & 11.1875 \end{bmatrix} \tag{7}$$

Steady-state RGA:

$$RGA(0) = G(0) \times (G^{-1}(0))^T = \begin{bmatrix} 35.9419 & -34.9419 \\ -34.9419 & 35.9419 \end{bmatrix} \tag{8}$$

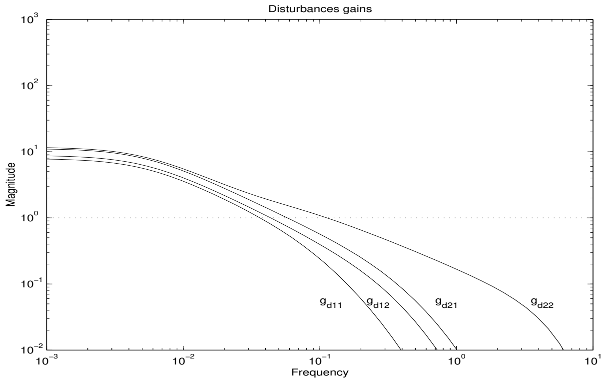
⋆ **Big elements in the steady state RGA indicate strong interactions between the loops**

⋆ **Negative off-diagonal elements indicate that should pair on the diagonal elements**

Matlab:

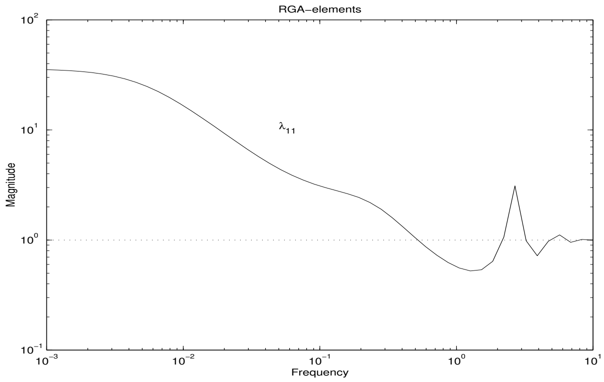
```
>> gw0 = frsp(sysm,0);
>> gwd0= frsp(sysmd,0);
>> RGA0= veval('.*',gw0,vpinv(vtp(gw0)));
```

Disturbance Gains as Function of Frequency



⋆ **To reject disturbances control is necessary for frequencies lower than $0.1s^{-1}$.**

RGA-matrix as Function of Frequency:

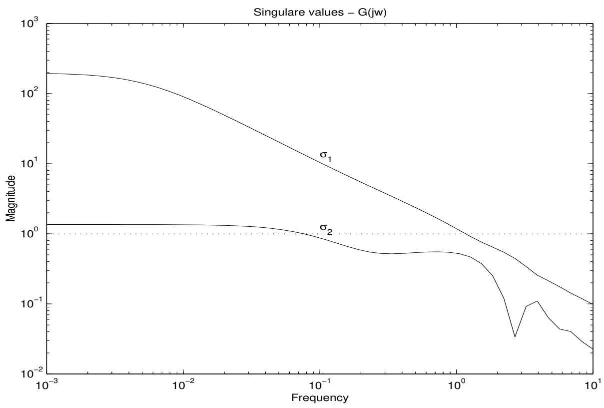


⋆ **The diagonal RGA-elements at necessary crossover ($\omega \approx 0.1s^{-1}$) are about 3. It seems sensible to pair on the diagonal.**

Matlab:

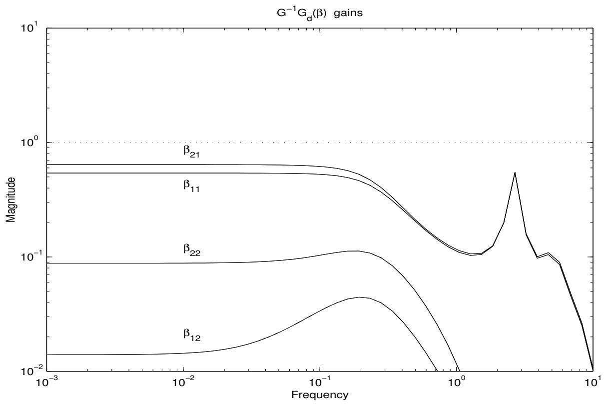
```
>> gwd = frsp(sysmd,w); vplot('liv,lm',gwd,1,':');
>> RGA = veval('.*',gw,vpinv(vtp(gw))); vplot('liv,lm',RGA,1,':');
```

Singular Values to $G(j\omega)$



⋆ **No problem with independent control of the outputs for frequencies lower than $0.1s^{-1}$.**

Input Saturation

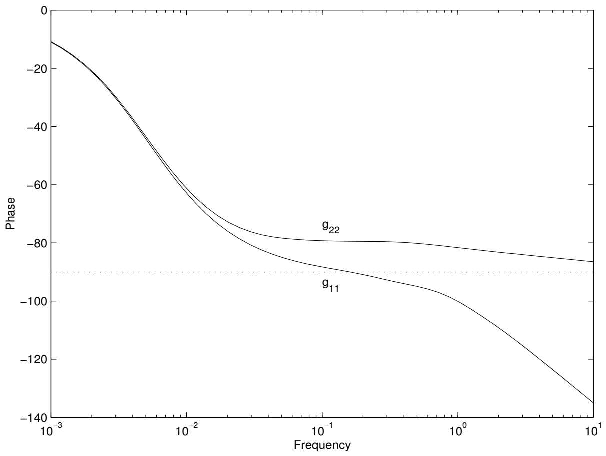


⋆ **All elements in $G^{-1}G_d$ less than 1 indicate no problem with input saturation.**

Matlab:

```
>>vplot('liv,lm',vsvd(gw),1,':');
>>vplot('liv,lm',mmult(minv(gw),gwd),1,':');
```

What Will Limit the Bandwidth for the Two Diagonal Controllers:



Achievable bandwidth will be limited by -180^o.

⋆ **Upper band width limits for one of the diagonal controllers are $0.2s^{-1}$, so to design two diagonal controllers should be possible.**

Matlab:

```
>> vplot('liv,p',mmult(frsp(-1,w),sel(gw,2,2)),sel(gw,1,1))
```

Will a Diagonal Controller Be Sufficient?

Performance and disturbance rejection of the individual loop is affected by the other loop.

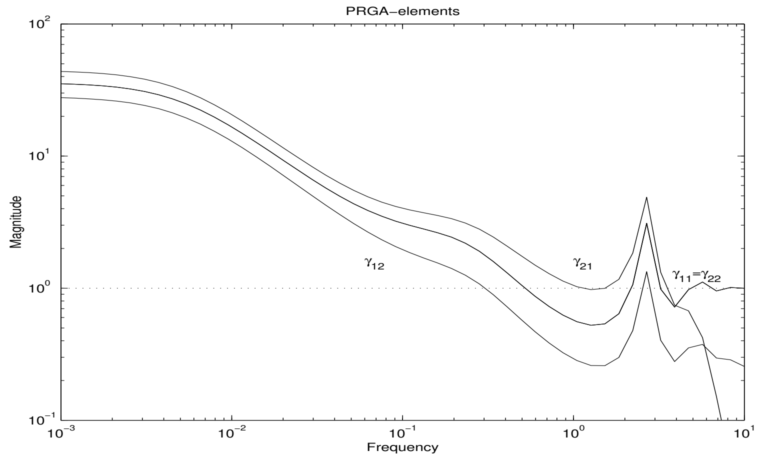
Performance Relative Gain Array

$$\Gamma(s) = \tilde{G}(s)G^{-1}(s), \tilde{G}(s) = diag(G(s)) \tag{9}$$

The PRGA

- diagonal elements equals the RGA elements.
- The off diagonal element gives interactions from the other loops, under feedback control.

PRGA as function of frequency:



⋆ **Small interactions for higher frequencies than $4s^{-1}$.**

⋆ **For lower frequencies than $4s^{-1}$ a reference change in output 1 will affect the output 2.**

Matlab:

```
>> gwdiag=vdiag(vdiag(gw));
>> prga  =mmult(gwdiag,minv((gw)));
```

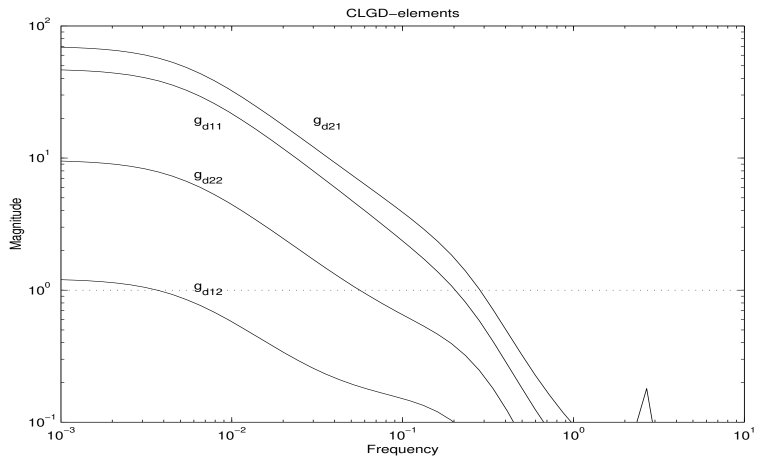
Closed-loop Disturbance Gain (CLDG)

$$\tilde{G}_d(s) = \Gamma(s) G_d(s) \tag{10}$$

CLDG gives the apparent disturbance gain as seen when the loops are under diagonal control.

Can show: Performance: $|1 + L_i| > \max\limits_{k,j} \{|\tilde{G}_{dik}|, |\gamma_{ij}|\}$

CLDG as function of frequency:



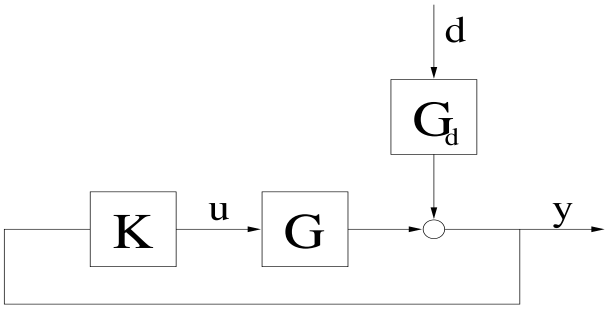
⋆ **Effective control is necessary for frequencies lower than $0.3s^{-1}$.**

Matlab:

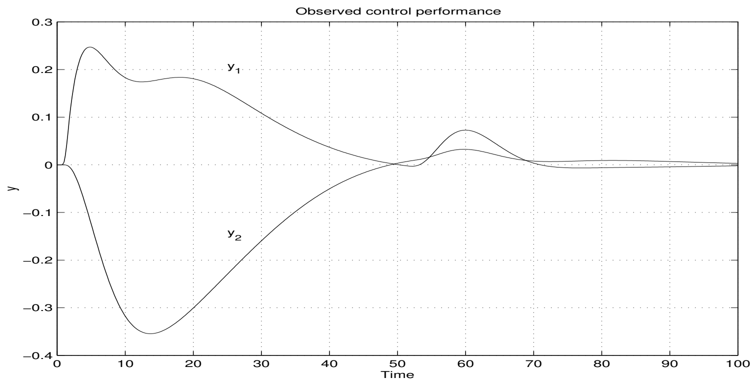
```
>> cldg=mmult(prga,gwd);
```

System Manipulation 3

Closing the loops $y = (I + GK)^{-1} G_d d$:



Observed control performance:



Matlab:

```
>> k1 = nd2sys([3.76 1], [3.76 1.e-4],  0.261);
>> k2 = nd2sys([3.31 1], [3.31 1.e-4], -0.375);
>> K = daug(k1,k2);
>> GK = mmult(sysm,K);
>> S = minv(madd(eye(2),GK));
>> SGd = mmult(S,sysmd);
>> Sf=frsp(S,w);
>> d = transp(vpck([1 0; 1 1],[0; 50]));
>> y = trsp(SGd,d,100,0.2);
interpolating input vector (zero order hold)
>> vplot(y);
```

Concluding Remarks / Summary

- The **controllability** of a system is its ability to achieve acceptable control performance.
- **Controllability analysis** of a system is performed to find out what control performance can be expected.
- Three different approach for doing controllability analysis: **"Simulation"**, **"Rigorous"** and **"Simple Tools"** approach.
- A **procedure for controllability analysis** based on "simple tools" is presented and demonstrated on a distillation column example.
- **Matlab** with its toolboxes is a useful tool for performing controllability analysis.
  - Control System Toolbox is closely integrated with standard Matlab.
  - But: Control System Toolbox is not sufficient for robust analysis and design.
  - Robust Control Toolbox and μ Synthesis Toolbox have good tools for robust analysis and design.
  - μ Synthesis Toolbox contains some very useful functions (e.g. sel.m and vplot.m) and seems best for controllability analysis purposes.