Review Material: CSE Course

Stability for Linearized Systems

To study stability of a linearized system we must remember that it can be derived from its (linearized) matrix A:

- The linearized system is **Asymptotically Stable** if and only if $\forall i : \Re(\lambda_i(A)) < 0$
- The linearized system is **Unstable** if and only if $\exists i : \Re(\lambda_i(A)) > 0$
- **No conclusion can be drawn** if and only if $\forall i : \Re(\lambda_i(A)) \leq 0$

Zeta Transform

If we want to compute the solution of a **discrete time dynamical system** we need to apply the **zeta transform** of the state space representation. We obtain $H(z) = C(zI - A)^{-1}B + D$. To perform the partial fraction expansion and thus the anti-transform we need to proceed as follow:

1. Compute $\tilde{F}(z) = \frac{F}{z}$
2. Compute the PFE of $\tilde{F}(z)$
3. Compute $F(z) = \tilde{F}(z) \cdot z$
4. Antitransform $F(z)$

Using matlab **this procedure can be simplified** by using the statement *residuez*

Recall that in the presence of complex conjugate poles $F(z) = \frac{Rz}{z-\lambda} + \frac{R^* z}{z-\lambda^*}, \lambda = \sigma + j\omega = \nu e^{j\theta}$ the antitransform is $f(k) = 2M\nu^k \cos(\theta k + \phi), M = |R|, \phi = \angle R$

```
% define system matrices
A=[3 0;-3.5 -0.5];
B=[1;2];
C=[1 -1];
%define initial condition
x0=[1;-2];
%define z-transform variable
z=tf('z',1);
%define z-transform of u(k)
U=2*z/(z-1);    % u(k)=2epsilon(k)

X_i = minreal(zpk((z*inv(z*eye(2)-A)*x0)),1e-3);

[num_X_i_1,den_X_i_1] = tfdata(X_i(1),'v'); %1st state
[r1,p1,k1]=residuez(num_X_i_1,den_X_i_1);

[num_X_i_2,den_X_i_2] = tfdata(X_i(2),'v'); %2nd state
[r2,p2,k2]=residuez(num_X_i_2,den_X_i_2);

%compute zero-state state response
X_f = minreal(zpk((inv(z*eye(2)-A)*B)*U),1e-3);

% compute residues of PFE
[num_X_f_1,den_X_f_1] = tfdata(X_f(1),'v'); %1st state
[r1,p1,k1]=residuez(num_X_f_1,den_X_f_1);

[num_X_f_2,den_X_f_2] = tfdata(X_f(2),'v'); %2nd state
[r2,p2,k2]=residuez(num_X_f_2,den_X_f_2);
```

Transfer Function for Discrete Time Dynamical Systems

```
T = 1;
z = tf('z',T);
H = 1/(z^2-1.7*z+0.72);
zpk(H);     % Zero-Pole-Gain form

[zH,pH,kH]=zpkdata(H,'v');  % To compute zeroes, poles and gain for a given tf
```

Natural modes for Discrete Time Dynamical Systems

The natural modes of the system $x(k+1) = Ax(k) + Bu(k)$ are:

- For the i-th distinct eigenvalue λ_i with minimal polynomial multiplicity $\mu_i^{'}$ are $m_{ij}(k) = \binom{k}{\mu_i^{'}-1}\lambda_i^{k-\mu_i^{'}+1} = \frac{k(k-1)\dots(k-(\mu_i^{'}-2))}{(\mu_i^{'}-1)!}\lambda_i^{k-\mu_i^{'}+1}$
- For a couple of complex conjugate poles of the form $\lambda = \sigma + j\omega = \nu e^{j\theta}$ having minimal polynomial multiplicity $\mu_i^{'}$ are $m_{ij}(k) = \binom{k}{\mu_i^{'}-1}\nu_i^{k-\mu_i^{'}+1}\cos(\theta(k-\mu_i^{'}-1)+\phi)$

Modal Analysis

Denote with $\lambda_i(A)$ the i-th eigenvalue of matrix A then:

- The natural mode associated with eigenvalue λ_i is **bounded** if $|\lambda_i(A)| = 1, \mu^{'}(\lambda_i(A)) = 1$
- The natural mode associated with eigenvalue λ_i is **convergent** if $|\lambda_i(A)| < 1$
- The natural mode associated with eigenvalue λ_i is **divergent** if $|\lambda_i(A)| > 1 \ OR \ |\lambda_i(A)| = 1, \ \mu^{'}(\lambda_i(A)) > 1$

Digital to Analog Conversion

To retrieve the original signal, after it has been sampled, we need to use an **ideal low pass filter**. In reality it's not possible to create such a thing, but the closest is the **Zero-Order-Hold** (ZOH) filter, which has a transfer function $G_{ZOH}(s) = \frac{1-e^{-Ts}}{s}$

Digital Control Design by Emulation

The main steps of the emulation design method are described in detail in the following sections.

Emulation 1: Choose suitable Sampling Time T

To choose a viable sampling time we need to take various aspects into account:

- **Sampling Theorem**: the sampling frequency must at least equal twice the value of the highest significant frequency in the signal. the highest significant frequency of all the signals in the loop is the system bandwidth ω_B. So we have $\omega_S > 2\omega_B \rightarrow \omega_S > 3\omega_C$ (for a well-dampened system $\omega_B > 1.5\omega_C$
- **ZOH Filter**: the D/A converter introduces a phase lag of $\angle G_{ZOH}(j\omega) = -\frac{\omega T}{2}$. In order to limit the phase lag at the cross-over frequency to small values (e.g. $-10°$ to $-5°$) we obtain $18\omega_C < \omega_S < 36\omega_C$
- **Suitable sampling of the transient**: a suitable number of samples must considered for describing the behavior of the transient phase typically 10 to 50 samples can be employed for a suitable description of the transient behavior for a well damped system (e.g. z = 0.6) we have: $11.42\omega_C < \omega_S < 57.2\omega_C$
- **Final Pratical Rule of Thumb**: provided that the chosen values satisfies HW and cost limitations choose T within the interval $20\omega_C < \omega_S < 50\omega_C$

Emulation 2: Analog Controller Design

If it is known in advance that the controller has to be realized through a digital computer, the design of $C_0(s)$ is performed taking into account the dynamics introduced by the A/D and the ZOH D/A transfer function. Thus, the analog controller design should be performed considering as plant transfer function $G^{'}(s) = G_{A/D}(s)G_{ZOH}(s)G(s) = \frac{1}{T}\frac{1-e^{-Ts}}{s}G(s)$. **Remark** : in MatLab environment, use the 1st order Padé approximation

Emulation 3: Discretization of Analog Controller

The simplest and most effective way to discretize a continuous time controller $C_0(s)$ is to consider its state space representation and suppose to integrate it through a numeric method, using the bilinear transformation $s = \frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}$. We obtain $C(z) = C_0(s)|_{s=\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}}$

- **Backward Euler** for $\alpha = 1$
- **Forward Euler** for $\alpha = 0$
- **Tustin** for $\alpha = 0.5$

The Tustin approximation provides the best frequency response matching of the analog controller.

We have another option, the **matched pole-zero** method (MPZ), which consists in applying the transformation $z = e^{sT}$. This method is, however, **worse in performance with respect to the Tustin approximation**. If we use the matlab statement *c2d(C_0, T, METHOD)* we can get the discretized version of the controller transfer function. The *METHOD* parameter is defaulted to 'zoh', so **we need to specify 'tustin'**

```
C = c2d(C_0, T, 'tustin');
```

Emulation 5: Anti-Aliasing Filter Design

We can employ a **Butterworth Filter** of the form $F(s) = \frac{1}{B_n(s)} \rightarrow s' = \frac{s}{\omega_f}$, with order:

- Order 1: $B_n(s) = (s' + 1)$
- Order 2: $B_n(s) = (s'^2 + 1.414s' + 1)$
- Order 3: $B_n(s) = (s'^2 + s' + 1)(s' + 1)$
- Order 4: $B_n(s) = (s'^2 + 0.765s' + 1)(s'^2 + 1.848s' + 1)$

It is required that the cut-off frequency ω_f satisfies $\omega_B < \omega_f < \frac{\omega_s}{2}$. The filter can be designed by imposing a given attenuation γ so that $|F(j\omega)| \leq \gamma, \forall\omega \geq \omega_h$ (usually $\omega_h = \frac{\omega_s}{2}$. We have $\omega_f = \omega_h(\frac{\gamma^2}{1-\gamma^2})^{\frac{1}{2n}} \approx \omega_h\gamma^{\frac{1}{n}}$ for $\gamma \ll 1$. The **order of the filter** is chosen n order to limit the phase lag introduced at ω_c using the normalized filter phase diagram.

[B,A] = BUTTER(N,Wf) designs an Nth order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). The coefficients are listed in descending powers of z. The cutoff frequency Wf must be 0.0 < Wf < 1.0, with 1.0 corresponding to half the sample rate.

```
% Design analog Butterworth filter
% In this case, Wf is in [rad/s] and it can be greater than 1.0
[B,A] = BUTTER(N,Wf,'s')
```

Control of LTI systems through Static State Feedback

In the presence of an input such as $u(t) = -Kx(t) + Nr(t)$ the dynamical properties of the (controlled) system depend on the eigenvalues of matrix $A - BK$ (K is referred to as the state gain).A suitable choice of K allows us to modify the system eigenvalues and improve the dynamic properties of the system (e.g. stabilize it). The reference gain matrix N can be chosen to modify the dc-gain of the controlled system.

Is it always possible to find a matrix K able to arbitrarily assign the eigenvalues of $A - BK$? **Yes** if and only if $\rho(M_r) = \rho([B \ AB \ \ldots \ A^{n-1}B]) = n$, where M_r is the **reachability matrix**. Typically, K is computed by placing the eigenvalues (poles) of the controlled system to obtain, besides asymptotic stability, good damping and rapidity properties of the transient. N can be chosen to make unitary the dc-gain of the controlled system guaranteeing zero steady state tracking error in the presence of a constant reference signal $r(t) = \bar{r}\epsilon(t)$.

To obtain the eigenvalues to be assigned by the state feedback controller, the computed values of ζ and ω_n are converted to the corresponding couple of complex conjugate values $\lambda_{1,2} = \sigma_0 \pm j\omega_0 = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$. What do we do if the system dimension is greater than 2? The additional eigenvalues are chosen with a **faster time constant** $\tau_{add} \ll \frac{1}{\zeta\omega_n}$. This means that all the eigenvalues $\lambda_3, \lambda_4, \ldots, \lambda_n$ will all be coincident ($\lambda_3 = \lambda_4 = \cdots = \lambda_n$) and of the form $\lambda_i = -a \cdot \zeta\omega_n$ where a is a **constant value chosen between 5 and 10**

```
Mr = ctrb(A,B);
rho_Mr = rank(Mr);
lambda_des = [-1 -2];
K = place(A,B,lambda_des);       % place is used for different eigenvalues

lambda_des = [-1 -1];
K = acker(A,B,lambda_des);       % acker is used for coincident eigenvalues

sys_cont_c = ss(A-B*K,B,C,0);
N = 1/dcgain(sys_cont_c);
```

State Observers

For an LTI continuous time system, A state feedback control law of the form $u(t) = -Kx(t) + Nr(t)$ can be evaluated only when the system state can be measured. A state observer is a "device" that provides an estimate $\hat{x}$ of the system state x exploiting the knowledge of the system input u and the measurement of the output y. For a continuous time dynamic system, the **state estimation error** is defined as $e(t) = \hat{x}(t) - x(t)$. An observer such that $\lim_{t\to\infty} |e(t)| = 0$ is referred to as an **asymptotic state observer**.

We can create a system such that $\dot{e}(t) = (A - LC)e(t)$. A suitable choice of L allows us to modify the observer eigenvalues and improve its dynamic properties. L is referred to as the **observer gain**. However, is it always possible to find a matrix L able to arbitrarily assign the eigenvalues of $A - LC$? **Yes** if and only if $\rho(M_o) = \rho([C \ AC \ \ldots \ A^{n-1}C]^T) = n$, where M_o is the **observability matrix**. The L matrix eigenvalues are usually chosen as coincident, with negative real part and null imaginary part

```
Mo = obsv(A, C);
rho_Mo = rank(Mo);

lambda_obsv_des = [-1 -2];
L = place(A',C',lambda_obsv_des)';

lambda_obsv_des = [-1 -1];
L = acker(A',C',lambda_obsv_des)';
```

Design Procedure for State-Feedback Controller using Estimated States

Thanks to the **separation principle** explained in the slides, we can separate the design of the state-controller and state-observer

1. Check for reachability and observability of the system
2. Compute the state feedback gain K to assign the eigenvalues of A-BK to the desired locations
3. Compute the dc-gain correction matrix N
4. Compute the observer gain L to assign the eigenvalues of A-LC to the desired locations. Typically the observer eigenvalues are chosen such that **the corresponding time constants are faster than those imposed by the control law** $\tau_{A-LC} \ll \tau_{A-BK}$. This means that they will be of the form $\lambda_{obs,i} = -a\zeta\omega_n$, where a is a constant value chosen between 5 and 10 times the largest real part of the state feedback controller eigenvalues

```
assert(rank(ctrb(A,B)) == rank(A));
assert(rank(obsv(A,C)) == rank(A));

s_hat = 0.08;
t_s = 4;
alpha = 1;

zeta = abs(log(s_hat))/(sqrt(pi^2+log(s_hat)^2));
omega_n = log((alpha/100)^(-1))/(zeta*t_s);

lambda_1 = -zeta*omega_n + 1i*omega_n*sqrt(1-zeta^2);
lambda_2 = -zeta*omega_n - 1i*omega_n*sqrt(1-zeta^2);
lambda_des = [lambda_1 lambda_2];

K = place(A,B,lambda_des);

sys_ctrb = ss(A-B*K, B, C, D);
N = 1/dcgain(sys_ctrb);

lambda_obs = -zeta * omega_n * 5;               % 5 Times faster than the state controller
lambda_obs_des = [lambda_obs lambda_obs];

L = acker(A', C', lambda_obs_des)';

% Simulation
sys_x = ss(A,B,eye(length(A)),D);
sys_obs = ss(A-L*C, [B L], eye(length(A)), D);
```

Review of Stability

	LTI System	Linearized LTI System (check $\forall$ equilibrium solution)	Discrete Time LTI System
Internally Bounded (check $\lambda_i(A)$)	$\forall i : \Re\{\lambda_i(A)\} \leq 0$, $\forall i \quad \| \quad \Re\{\lambda_i(A)\} = 0, \mu_i^{'} = 1$		$\forall i : \|\lambda_i\| \leq 1$, $\forall i \quad \| \quad \|\lambda_i\| = 1, \mu_i^{'} = 1$
Asymptotically Stable (check $\lambda_i(A)$)	$\forall i : \Re\{\lambda_i(A)\} < 0$	$\forall i : \Re\{\lambda_i(A)\} < 0$	$\forall i : \|\lambda_i(A)\| < 1$
BIBO Stable (check $H(s)$)	$\forall i : \Re\{p_i\} < 0$, with p_i being the i-th pole of $H(s)$		$\forall i : \|p_i\| < 1$, with p_i being the i-th pole of $H(s)$