Matave Control Toolbox

Applied Control Engineering Toolbox for MATLAB and GNU Octave

Version 2.0

Model building

Function name	*Description*	*Status*	*MIMO*	*TF/SS*	*Discrete*
tf	Crate transfer function model	Done	No	N/N	Y
zpk	Create zero-pole-gain model	Done	No	N/N	Y
ss	Create state space model	Done	Yes	N/N	Y

Model transformation

Function name	*Description*	*Status*	*MIMO*	*TF/SS*	*Discrete*
minreal	Minimal realization	Done	Yes	Y/Y	Y
balreal	Balanced realization	Done	Yes	N/Y	Y
modred	Model reduction	Done	Yes	N/Y	Y
append	Append systems	Done	Yes	Y/Y	Y
feedback	Feedback model	Done	Yes	Y/Y	Y
series	Serial model	Done	Yes	Y/Y	Y
parallel	Parallel model	Done	Yes	Y/Y	Y

Model data access

Function name	*Description*	*Status*	*MIMO*	*TF/SS*	*Discrete*
dcgain	Get the low frequency gain	Done	Yes	Y/Y	Y
pzmap	Plot poles and zeros	Done	Yes	Y/Y	Y
damp	Get the damping	Done	Yes	Y/Y	Y
pole	Get poles	Done	Yes	Y/Y	Y
zero	zeros for SISO	Done	No	Y/Y	Y
tzero	zeros for MIMO	Done	Yes	N/Y	Y

Model conversions

Function name	*Description*	*Status*	*MIMO*	*TF/SS*	*Discrete*
c2d	Convert continuous to discrete	Done	Yes	Y/Y	N
c2dt	Convert continuous to discrete with delay	Done	Yes	Y/Y	N
d2c	Convert discrete to continuous	Done	Yes	Y/Y	Y
d2d	Rediscrete the model	Done	Yes	Y/Y	Y
tf2ss	Transfer function to state space	Done	No	Y/N	Y
ss2tf	State space to transfer function	Done	Yes	N/Y	Y

Frequency domain analysis

Function name	**Description**	**Status**	*MIMO*	*TF/SS*	*Discrete*
evalfr	Get one frequency	Done	Yes	Y/Y	Y
freqresp	Get multiple frequencies	Done	Yes	Y/Y	Y
bode	Bode diagram	Done	Yes	Y/Y	Y
bodemag	Bode diagram without phase	Done	Yes	Y/Y	Y
nyquist	Nyquist diagram	Done	Yes	Y/Y	Y
sigma	Singular value diagram	Done	Yes	Y/Y	Y
margin	Stability margins	Done	Yes	Y/Y	Y
allmargin	Show all margin	Done	Yes	Y/Y	Y
sensitivity	Show sensitivity margins	Done	Yes	Y/Y	Y
db2mag	Convert dB to magnintude	Done	Yes	Y/Y	Y
mag2db	Conver magnintude to dB	Done	Yes	Y/Y	Y
rlocus	Root locus plot	Done	Yes	Y/Y	Y
dBdrop	Find the frequency at 3 dB drop	Done	Yes	Y/Y	Y

Time domain analysis

Function name	*Description*	*Status*	*MIMO*	*TF/SS*	*Discrete*
gensig	Generate signals	Done	No	N/N	Y
impulse	Impulse response	Done	Yes	Y/Y	Y
step	Step response	Done	Yes	Y/Y	Y
ramp	Ramp response	Done	Yes	Y/Y	Y
initial	Response with initial conditions	Done	Yes	N/Y	Y
lsim	Linear simulation response	Done	Yes	Y/Y	Y

Singel variable control

Function name	***Description***	***Status***	***MIMO***	***TF/SS***	***Discrete***
pid	Parallel PID controller	Done	No	N/N	Y
pipd	Serial PID controller	Done	No	N/N	Y
loop	Loopshaping controller	Done	No	Y/N	Y
acker	Acker formula	Done	No	N/Y	Y

Multivariable control

Function name	***Description***	***Status***	***MIMO***	***TF/SS***	***Discrete***
lqr	Linear quadratic regulator	Done	Yes	N/Y	Y
lqe	Linear quadratic estimator	Done	Yes	N/Y	Y
Lqi	Linear quadratic integral	Done	Yes	N/Y	Y
reg	Generates the LQ-model	Done	Yes	N/Y	Y
lqgreg	Generates the Gaussian LQG-model	Done	Yes	N/Y	Y

Matrix equations

Function name	***Description***	***Status***	***MIMO***	***TF/SS***	***Discrete***
lyap	Solve Lyapunov equation	Done	Y	N/N	Y
are	Solve algibraic riccati equation	Done	Y	N/Y	Y
obsv	Observbility matrix	Done	Y	N/Y	Y
ctrb	Controllbility matrix	Done	Y	N/Y	Y
gram	Gramian	Done	Y	N/Y	Y
hsvd	Hankel singular values	Done	Y	N/Y	Y
covar	Covaraiance matrix	Done	Y	N/Y	Y

LQR + LQI:

Assume that we have a regular state space model

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

The LQR control law

$$u = -L\,x + Li\,x_i$$

with integral action LQI

$$\dot{x}_i = r - y = r - Cx - Du$$
$$x_i = \int \dot{x}_i = \int (r - Cx - Du)$$

This will create the augmented state space model

$$\begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} B \\ -D \end{bmatrix} [u] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [r]$$
$$y = Cx + Du$$

With the LQR and LQI control law. The augmented state space model will then be:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} (A - BL) & BLi \\ (DL - C) & -DLi \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [r]$$
$$[y] = \begin{bmatrix} (C - DL) & DLi \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix}$$

LQR + LQE:

Assume a state space gaussian model

$$\dot{x} = Ax + Bu + B_d d$$
$$y = Cx + Du + B_n n$$

And a state state space gaussian observer

$$\dot{\hat{x}} = A\hat{x} + Bu + Ke$$
$$\hat{y} = C\hat{x} + Du$$
$$e = y - \hat{y} = Cx + Du + B_n n - C\hat{x} - Du = Cx + B_n n - C\hat{x}$$

The LQR control law

$$u = -Lx + r$$

One important step is to create one error state of diffrence between real state and estimated state

$$\tilde{x} = x - \hat{x}$$

This will change the control law to a new control law

$$u = -L\hat{x} + r = L\tilde{x} - Lx + r$$

Because

$$\tilde{x} - x = -\hat{x}$$

Insert the new control law into the state space gaussian model

$$\dot{x} = Ax + B(L\tilde{x} - Lx + r) + B_d d$$
$$\dot{x} = (A - BL)x + BL\tilde{x} + Br + B_d d$$

Insert the new control law to the output of the state space gaussian model

$$y = Cx + D(L\tilde{x} - Lx + r) + B_n n$$
$$y = (C - DL)x + DL\tilde{x} + Dr + B_n n$$

Insert the new control law into the state state space gaussian observer

$$\dot{\hat{x}} = A\hat{x} + Bu + Ke$$
$$\dot{\hat{x}} = A\hat{x} + B(L\tilde{x} - Lx + r) + K(Cx + B_n n - C\hat{x})$$
$$\dot{\hat{x}} = A\hat{x} + BL\tilde{x} - BLx + Br + KCx + KB_n n - KC\hat{x}$$

Assume now

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$$
$$\dot{\tilde{x}} = [(A - BL)x + BL\tilde{x} + r + B_d d] - [A\hat{x} + BL\tilde{x} - BLx + r + KCx + KB_n n - KC\hat{x}]$$
$$\dot{\tilde{x}} = Ax - BLx + BL\tilde{x} + r + B_d d - A\hat{x} - BL\tilde{x} + BLx - r - KCx - KB_n n + KC\hat{x}$$
$$\dot{\tilde{x}} = Ax + B_d d - A\hat{x} - KCx - KB_n n + KC\hat{x}$$

Remember

$$\tilde{x} = x - \hat{x}$$

Which results:

$$\dot{\tilde{x}} = Ax + B_d d - A\hat{x} - KCx - KB_n n + KC\hat{x}$$
$$\dot{\tilde{x}} = A(x - \hat{x}) + B_d d - KC(x - \hat{x}) - KB_n n$$
$$\dot{\tilde{x}} = (A - KC)\tilde{x} + B_d d - KB_n n$$

The whole state space model will then be:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} (A - BL) & BL \\ 0 & (A - KC) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B & B_d & 0 \\ 0 & B_d & -KB_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$
$$y = \begin{bmatrix} (C - DL) & DL \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} Dr & 0 & B_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$

Because the most important equations are:

$$\dot{x} = (A - BL)x + BL\tilde{x} + Br + B_d d$$
$$\dot{\tilde{x}} = (A - KC)\tilde{x} + B_d d - KB_n n$$
$$y = (C - DL)x + DL\tilde{x} + Dr + B_n n$$

LQR + LQE + LQI:

Assume a state space gaussian model

$$\dot{x}=Ax+Bu+B_d d$$
$$y=Cx+Du+B_n n$$

And a state state space gaussian observer

$$\dot{\hat{x}}=A\hat{x}+Bu+Ke$$
$$\hat{y}=C\hat{x}+Du$$
$$e=y-\hat{y}=Cx+Du+B_n n-C\hat{x}-Du=Cx+B_n n-C\hat{x}$$

The LQR control law

$$u=-L\hat{x}+Li\,x_i$$

with integral action LQI

$$\dot{x}_i=r-y=r-Cx-Du-B_n n$$
$$x_i=\int\dot{x}_i=\int(r-Cx-Du-B_n n)$$

One important step is to create one error state of diffrence between real state and estimated state

$$\tilde{x}=x-\hat{x}$$

This will change the control law to a new control law

$$u=-L\hat{x}+Li\,x_i=L\tilde{x}-L\,x+Li\,x_i$$

Because

$$\tilde{x}-x=-\hat{x}$$

Insert the new control law into the integral state

$$\dot{x}_i=r-y=r-Cx-D(L\tilde{x}-L\,x+Li\,x_i)-B_n n$$
$$\dot{x}_i=r-y=r-Cx-DL\tilde{x}+DL\,x-DLi\,x_i-B_n n=r+(DL-C)x-DL\tilde{x}-DLi\,x_i-B_n n$$

Insert the new control law into the state space gaussian model

$$\dot{x}=Ax+B(L\tilde{x}-L\,x+Li\,x_i)+B_d d$$
$$\dot{x}=(A-BL)x+BL\tilde{x}+BLi\,x_i+B_d d$$

Insert the new control law to the output of the state space gaussian model

$$y=Cx+D(L\tilde{x}-L\,x+Li\,x_i)+B_n n$$
$$y=(C-DL)x+DL\tilde{x}+DLi\,x_i+B_n n$$

Insert the new control law into the state state space gaussian observer

$$\dot{\hat{x}}=A\hat{x}+Bu+Ke$$
$$\dot{\hat{x}}=A\hat{x}+B(L\tilde{x}-L\,x+Li\,x_i)+K(Cx+B_n n-C\hat{x})$$
$$\dot{\hat{x}}=A\hat{x}+BL\tilde{x}-BL\,x+BLi\,x_i+KCx+KB_n n-KC\hat{x}$$

Assume now

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$$
$$\dot{\tilde{x}} = [(A - BL)x + BL\tilde{x} + BLi\,x_i + B_d d] - [A\hat{x} + BL\tilde{x} - BL x + BLi\,x_i + KCx + KB_n n - KC\hat{x}]$$
$$\dot{\tilde{x}} = Ax - BLx + BL\tilde{x} + BLi\,x_i + B_d d - A\hat{x} - BL\tilde{x} + BL x - BLi\,x_i - KCx - KB_n n + KC\hat{x}$$
$$\dot{\tilde{x}} = Ax + B_d d - A\hat{x} - KCx - KB_n n + KC\hat{x}$$

Remember

$$\tilde{x} = x - \hat{x}$$

Which results

$$\dot{\tilde{x}} = Ax + B_d d - A\hat{x} - KCx - KB_n n + KC\hat{x}$$
$$\dot{\tilde{x}} = A(x - \hat{x}) + B_d d - KC(x - \hat{x}) - KB_n n$$
$$\dot{\tilde{x}} = (A - KC)\tilde{x} + B_d d - KB_n n$$

The whole state space model will then be:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} (A-BL) & BL & BLi \\ 0 & (A-KC) & 0 \\ (DL-C) & -DL & -DLi \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \\ x_i \end{bmatrix} + \begin{bmatrix} 0 & B_d & 0 \\ 0 & B_d & -KB_n \\ 1 & 0 & -B_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$
$$y = \begin{bmatrix} (C-DL) & DL & DLi \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \\ x_i \end{bmatrix} + \begin{bmatrix} 0 & 0 & B_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$

Because the most important equations are:

$$\dot{x} = (A - BL)x + BL\tilde{x} + BLi\,x_i + B_d d$$
$$\dot{\tilde{x}} = (A - KC)\tilde{x} + B_d d - KB_n n$$
$$\dot{x}_i = r + (DL - C)x - DL\tilde{x} - DLi\,x_i - B_n n$$
$$y = (C - DL)x + DL\tilde{x} + DLi\,x_i + B_n n$$