

# Matave Control Toolbox

Applied Control Engineering Toolbox for MATLAB and GNU Octave

Version 4.0

## Model building

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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
pade	Internal time delay to model	Done	Yes	Y/N	Y

## Model data access

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

<b>Function name</b>	<b>Description</b>	<b>Status</b>	<b>MIMO</b>	<b>TF/SS</b>	<b>Discrete</b>
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
findmaxgain	Compute the max gain limit	Done	Yes	Y/Y	Y
dBdrop	Find the frequency at 3 dB drop	Done	Yes	Y/Y	Y

## Time domain analysis

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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
nlsim	Nonlinear simulation response	Done	Yes	Y/Y	Y

## Singel variable control

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
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

## Compensators

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>MIMO</i>	<i>TF/SS</i>	<i>Discrete</i>
smithpredict	Otto Smith delay compensator	Done	No	Y/N	Y
imc	Disturbance compensator	Done	No	Y/N	Y

## Miscellaneous

<i>Function name</i>	<i>Description</i>	<i>Status</i>	<i>Internet connection</i>
updatematavecontrol	Update the Matavecontrol library	Done	Y

## LQR + LQI:

Assume that we have a regular state space model

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

$$y = Cx + Du$$

The LQR control law

$$u = -Lx + Li x_i$$

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:

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

$$[y] = [(C - DL) \quad DLi] \begin{bmatrix} x \\ x_i \end{bmatrix}$$

## LQR + LQE:

Assume a state space gaussian model

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

And a state space gaussian observer

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

The LQR control law

$$u = -Lx + r$$

One important step is to create one error state of difference 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

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

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:

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

$$y = [(C - DL) \quad DL] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + [Dr \quad 0 \quad B_n] \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$

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

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

And a state space gaussian observer

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

The LQR control law

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

with integral action LQI

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

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

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

This will change the control law to a new control law

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

Because

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

Insert the new control law into the integral state

$$\begin{aligned}\dot{x}_i &= r - y = r - Cx - D(L\tilde{x} - Lx + Li x_i) - B_n n \\ \dot{x}_i &= r - y = r - Cx - DL\tilde{x} + DLx - DLi x_i - B_n n = r + (DL - C)x - DL\tilde{x} - DLi x_i - B_n n\end{aligned}$$

Insert the new control law into the state space gaussian model

$$\begin{aligned}\dot{x} &= Ax + B(L\tilde{x} - Lx + Li x_i) + B_d d \\ \dot{x} &= (A - BL)x + BL\tilde{x} + BLi x_i + B_d d\end{aligned}$$

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

$$\begin{aligned}y &= Cx + D(L\tilde{x} - Lx + Li x_i) + B_n n \\ y &= (C - DL)x + DL\tilde{x} + DLi x_i + B_n n\end{aligned}$$

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

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

Assume now

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

Remember

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

Which results

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

The whole state space model will then be:

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

Because the most important equations are:



$$\begin{aligned}\dot{x} &= (A - BL)x + BL\tilde{x} + BLix_i + B_d d \\ \dot{\tilde{x}} &= (A - KC)\tilde{x} + B_d d - KB_n n \\ \dot{x}_i &= r + (DL - C)x - DL\tilde{x} - DLix_i - B_n n \\ y &= (C - DL)x + DL\tilde{x} + DLix_i + B_n n\end{aligned}$$