VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

XMUT315 Control Systems Engineering

Demo 1: Systems Modelling

Matlab Commands Summary

abs(x)	Obtain absolute value of x.
acker(A, B, poles)	Find gains for pole placement.
angle(x)	Compute the angle of x in radians.
atan(x)	Compute arctan(x).
axis([xmin, xmax, ymin, ymax])	Define range on axes of a plot.
bode(G, w)	Make a Bode plot of transfer function $G(s)$ over a range of frequencies, ω. Field ω is optional.
break	Exit loop.
c2d(G, T, 'tustin')	Convert $G(s)$ to $G(z)$ using the Tustin transformation. T is the sampling interval.
c2d(G, T, 'zoh')	Convert $G(s)$ in cascade with a zero-order hold to $G(z)$. T is the sampling interval.
canon(S, 'modal')	Convert an LTI state-space object, S, to parallel form.
clear	Clear variables from workspace.
clf	Clear current figure.
conv([a b c d], [e f g h])	Multiply $(as^3 + bs^2 + cs + d)$ by $(es^3 + fs^2 + gs + h)$.
ctrb(A, B)	Find controllability matrix.
d2c(G, 'zoh')	Convert $G(z)$ to $G(s)$ in cascade with a zero-order hold.
dcgain(G)	Find dc gain for $G(s)$ (that is, $s = 0$), or $G(z)$ (that is, $z = 1$).
eig(A)	Find eigenvalues of matrix $\mathbf{A}$.
end	End the loop.
exp(a)	Obtain e^a.
feedback(G, H, sign)	Find $T(s) = G(s)/[1 \pm G(s)H(s)]$. Sign $= -1$ or is optional for negative feedback systems. Sign $= +1$ for positive feedback systems.

`grid on`	Put grid lines on a graph.
`hold off`	Turn off graph hold; start new graph.
`imag(P)`	Form a matrix of the imaginary parts of the components of matrix **P**.
`input('str')`	Permit variable values to be entered from the keyboard with prompt `str`.
`interp1(x, y, x1)`	Perform table lookup by finding the value of y at the value of $x = x_1$.
`inv(P)`	Find the inverse of matrix **P**.
`length(P)`	Obtain dimension of vector **P**.
`log(x)`	Compute natural log of x.
`log10(x)`	Compute log to the base 10 of x.
`margin(G)`	Find gain and phase margins, and gain and phase margin frequencies of transfer function, $G(s)$. Return [Gain margin, Phase margin, 180° frequency, 0 dB frequency].
`max(P)`	Find the maximum component of **P**.
`minreal(G, tol)`	Cancel common factors from transfer function $G(s)$ within tolerance, tol. If `tol` field is blank, a default value is used.
`ngrid`	Superimpose grid over a Nichols plot.
`nichols(G, w)`	Make a Nichols plot of transfer function $G(s)$ over a range of frequencies, ω. Field ω is optional.
`nyquist(G, w)`	Make a Nyquist diagram of transfer function $G(s)$ over a range of frequencies, ω. Field ω is optional.
`obsv(A, C)`	Find observability matrix.
`ord2(wn, z)`	Create a second-order system, $G(s) = 1/[s^2 + 2\zeta\omega_n s + \omega_n^2]$.
`pade(T, n)`	Obtain nth order Padé approximation for delay, T.
`pause`	Pause program until any key is pressed.
`plot(t1, y1, t2, y2, t3, y3)`	Plot $y1$ versus $t1$, $y2$ versus $t2$, and $y3$ versus $t3$ on the same graph.
`pole(G)`	Find poles of LTI transfer function object, $G(s)$.
`poly([-a -b -c])`	Form polynomial $(s + a)(s + b)(s + c)$.
`polyval(P, a)`	Find polynomial $P(s)$ evaluated at a, that is, $P(a)$.
`rank(A)`	Find rank of matrix **A**.
`real(P)`	Form a matrix of the real parts of the components of matrix **P**.
`residue(numf, denf)`	Find residues of $F(s)$ = numf/denf.
`rlocfind(GH)`	Allow interactive selection of points on a root locus plot for loop gain, $G(s)H(s)$. Return value for K and all closed-loop poles at that K.
`rlocus(GH, K)`	Plot root locus for loop gain, $G(s)H(s)$, over a range of gain, K. The K field is optional.

`roots(P)`	Find roots of polynomial, P.
`semilogx(w, P1)`	Make a semilog plot of P_1 versus $\log_{10}(\omega)$.
`series(G1, G2)`	Find $G_1(s)G_2(s)$.
`sgrid(z, wn)`	Overlay z(ζ) and wn(ω_n) grid lines on a root locus.
`sin(x)`	Find $\sin(x)$.
`sqrt(a)`	Compute $\sqrt{a}$.
`ss2tf(A, B, C, D, 1)`	Convert a state-space representation to a transfer function. Return `[num, den]`.
`ss(A, B, C, D)`	Create an LTI state-space object, S.
`ss(G)`	Convert an LTI transfer function object, $G(s)$, to an LTI state-space object.
`ssdata(S)`	Extract **A**, **B**, **C**, and **D** matrices from LTI state-space object, S.
`step(G1, G2, .. Gn, t)`	Plot step responses of $G_1(s)$ through $G_n(s)$ on one graph over a range of time, t. Field t is optional as are fields G_2 through G_n.
`subplot(xyz)`	Divide plotting area into an x by y grid with z as the window number for the current plot.
`tan(x)`	Find tangent of x radians.
`text(a, b, 'str')`	Put `str` on graph at graph coordinates, $x = a, y = b$.
`tf2ss(numg, deng)`	Convert $G(s)$ = `numg/deng` to state space in controller canonical form. Return `[A, B, C, D]`.
`vpa(a, D)`	Calculate a with D digits and convert to a symbolic with D digits.
`tf2zp(numg, deng)`	Convert $G(s)$ = `numg/deng` in polynomial form to factored form. Return `[zeros, poles, gains]`.
`tf(numg, deng, T)`	Create an LTI transfer function, $G(s)$ = `numg/deng`, in polynomial form. T is the sampling interval and should be used only if G is a sampled transfer function.
`tf(G)`	Convert an LTI transfer function, $G(s)$, to polynomial form.
`tfdata(G, 'v')`	Extract numerator and denominator of an LTI transfer function, $G(s)$, and convert values to a vector. Return `[num, den]`.
`title('str')`	Put title `str` on graph.
`xlabel('str')`	Put label `str` on x axis of graph.
`ylabel('str')`	Put label `str` on y axis of graph.
`zgrid`	Superimpose z(ζ) and wn (ω_n) grid curves on a z-plane root locus.
`zgrid([ ],[ ])`	Superimpose the unit circle on a z-plane root locus.
`zp2tf([-a -b]', [-c -d]', K)`	Convert $F(s) = K(s+a)(s+b)/(s+c)(s+d)$ to polynomial form. Return `[num, den]`.

`zpk(numg, deng, K, T)`	Create an LTI transfer function, $G(s)$ = `numg/deng`, in factored form. T is the sampling interval and should be used only if G is a sampled transfer function.
`zpk(G)`	Convert an LTI transfer function, $G(s)$, to factored form.