Robust Control Systems — Using the Matlab toolbox

J Treurnicht

April 23, 2007

1 Background

The Matlab Robust Control toolbox was developed along two routes: (1) a lean and simple Robust Control toolbox; and (2) a feature-rich Mu toolbox. These two toolboxes are now combined in the new Robust Control Toolbox with most of the existing tools but with a new set of tools that integrate well with the Control Toolbox.

2 Guide to using the Toolbox

2.1 Systems with Uncertainty

To create plants with uncertainty, a number of Matlab commands such as `ureal`, `ucomplex` are used for parameters and `udyn` and `ultidyn` can set up transfer functions.

To set up an uncertain plant transfer function $G_1(s)$ with

$$G_1(s) = \frac{k}{s(s+a)}, \quad k \in [10, 40],\ k_{\text{nom}} = 15,\ a \in [3 \pm 50\%]$$

we can use the commands

```
k1 = ureal('k',15,'Range',[10 40]);    % k vary between 10 and 40, nominal is 15
a1 = ureal('a',3,'percent',50);        % a vary by +/- 50 percent; nominal is 3
%G1 = ss([0 1; 0 -a1],[0; k1],[1 0], []);  % optional
G1 = tf(k1,[1 a1 0]);                  % Transfer function
```

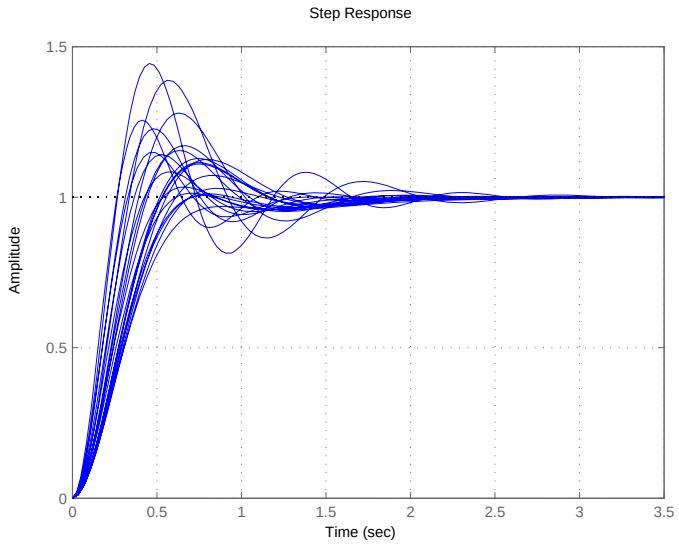
This creates an array of plants including the nominal plant. Adding a controller and closing the loop can be performed normally.

```
D  = tf(2*[1 2],[1 4]);  % Simple lead controller
T1 = feedback(G1*D,1);    % Closed loop
```

The step response can be obtained by the command

```
step(T1),grid
```

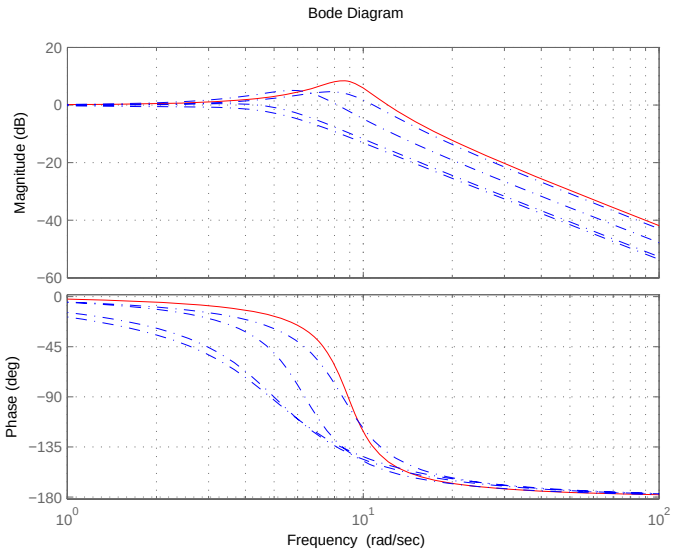
with responses



The calculations are typically slow (many curves), but we can minimize this by extracting the worst case and re-casting the rest into a smaller random set (functions `wcgain, usubs` and `usample`). The following example for the closed loop bode plot shows how

```
[PeakGain,Uwc] = wcgain(T1);  % Worst case gain
Twc   = usubs(T1,Uwc);        % Form worst-case system
Trand = usample(T1,4);        % Form some random systems
bode(Twc,'r',Trand,'b-.',{1,100}),grid
```

with responses



2.2 μ-Toolbox methods

The μ-Toolbox uses a new system description or **S**-matrix. To enter a a transfer function one would use

```
G1 = nd2sys(15,[1 3 0],1);   % nd2sys(num,den,gain)
G2 = pck(A, B, C, D);        % state space to packed (S) format
G3a = abv(G11,G21);          % Stack G11 and G21 on top of each other
G3  = sbs(G3a,G3b);          % Stack G3a and G3b next to each other
Gf  = frsp(G,w);             % Compute frequency response
[u,s,v] = vsvd(Gf);          % Get SVD at all frequencies
vplot('liv,lm',s);           % Plot the singular values on log-log scale
pkvnorm(s);                  % Estimate maximum value from plot
hinfnorm(G,0.0001);          % Accurate norm using a bisection search
```