ssresp.mixed Analytic and numerical output response example in Matlab

1 In the following example, we explore the output response derived both analytically and numerically in Matlab.

Example ssresp.mixed−1

re: analytic and numerical output response solution in Matlab

Consider a state-space model with the following standard matrices.

```
A = [...
  -1, 3, 5, 7;...
  0, -2, 0, 6;...
  -2, 1, -3, 0;...
  0, 1, 3, -4;...
];
n = length(A); % order
B = [...
  0; 1; 0; 2; ...
];
C = eye(n);
D = zeros([n,1]);
```

Solve for the unit step response output y given the following initial condition.

```
x0 = [2;0;2;0];
```

Analytic solution

We use the solution of Eq. 8:

$$y(t) = C\Phi(t)x(0) + C\int_0^t \Phi(t-\tau)Bu(\tau)d\tau + Du(t). \quad (1)$$

First we need $\Phi(t)$. The "primed" basis requires the eigendecomposition.

```
[M,L] = eig(A);
```

We can find Φ from the primed-basis version Φ', which is easy to compute.

```
Phi_p = @(t) diag(diag(exp(L*t)));
```

Now the basis transformation.

```
M_inv = M^-1; % compute just once, not on every call
Phi = @(t) M*Phi_p(t)*M_inv;
```

Declare symbolic variables.

```
syms T tt
```

Apply Eq. 8.

```
y_sym = C*Phi(tt)*x0 + C*int(Phi(tt-T)*B*1,T,0,tt) + D*1;
```

Convert this to a numerically evaluable function.

```
y_num = matlabFunction(y_sym);
```

Plot it; the result is shown in Fig. mixed.1.

```
figure
t_num = linspace(0,8,200);
plot(t_num,y_num(t_num),'linewidth',1)
xlabel('time (s)')
ylabel('analytic output response')
legend('y_1','y_2','y_3','y_4')
```

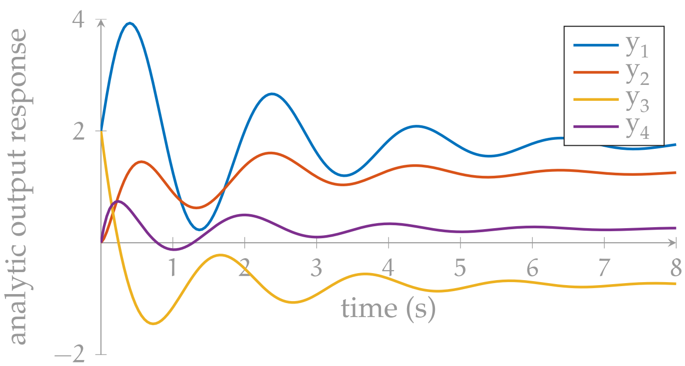


Figure mixed.1: the analytic output response.

Numerical solution

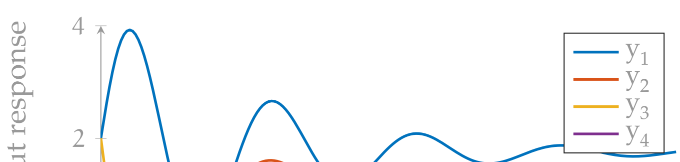
```
sys = ss(A,B,C,D);
```

Using *lsim*

First, use `lsim` to compute the response numerically.

```
u_s = ones(size(t_num)); % a one for every time
y_lsim = lsim(sys,u_s,t_num,x0); % simulate
```

Now plot it; the result is shown in Fig. mixed.2.



```
figure
plot(t_num,y_lsim,'linewidth',1)
xlabel('time (s)')
ylabel('numerical output response')
legend('y_1','y_2','y_3','y_4')
hgsave(h,'figures/temp');
```

Now take the difference between the two solutions and plot the error. As Fig. mixed.3 shows, the differences are minimal.

```
figure
plot(t_num,y_lsim-y_num(t_num).','linewidth',1)
xlabel('time (s)')
ylabel('error in output response')
legend('y_1','y_2','y_3','y_4')
```

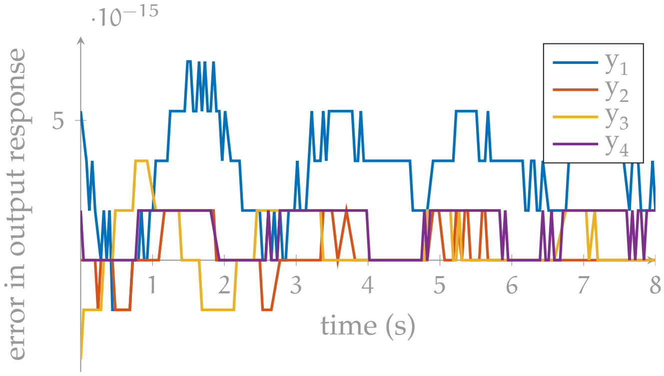


Figure mixed.3: comparison of analytic and numerical output responses.

Using the *step* and *initial* commands with superposition

Just for fun, here's how we could use `step` and `initial` (instead of `lsim`) with superposition to numerically solve.

```
y_step = step(sys,t_num); % forced response
y_initial = initial(sys,x0,t_num); % free response
y_total = y_initial + y_step; % (superposition)
```