

Für ein dynamisches System $4 \frac{d^3 \alpha}{dt^3} + c_2 \frac{d^2 \alpha}{dt^2} + c_1 \frac{d \alpha}{dt} = 12 \sin(0,2u)$

mit der Eingangsgröße u , der Ausgangsgröße α und den beiden Parametern c_1, c_2 gebe man die Realisierung in Matlab als M-Block an.

```

function varargout = teleskop(flag, t, x, u, p)
% teleskop: Klausurbeispiel (execution part)
% [nu,ny,acc,Ts,xo] = ('init' , t,[],[], p)
% [y, x] = ('output', t, x, u, p)
% ('terminate',t, x)
% Definitions:
% flag : M-block mode: 'init', 'output', 'terminate'
% t : simulation time in s (not used)
% x : inner state; for size see xo
% u(nu=1) :
% p-struct:
%
%
%
% y(ny=1) :
%
% acc : (direct access)
%
% Editor:
%
switch flag
case 'init' % [nu,ny,acc,Ts,xo] = ('init', t,[],[], p)
    varargout{1} = % fill in the number of inputs
    varargout{2} = % fill in the number of outputs
    varargout{3} = % if direct access behavior then set to 1 else 0
    varargout{4} =
    varargout{5} = % initial state

case 'output' % [y, x] = ('output', t, x, u, p)
    varargout{1} = % output equation
    % ..... add here the possible feedthrough quotas to the outputs
    [tt,xx] = ode23(@fxdot,[0 p.Ts],x.old,[],p,u);
    x.old=xx(end,:);
    varargout{2} =

case 'terminate' % ('terminate',t, x)
    % do some finalizing things
otherwise % Unexpected flags
    error(['Unexpected flag: ' flag])
end

% dynamic model
function [xdot]=fxdot(t,x,p,u)

```

return