Sate Space analysis

1 Theoretical aspects

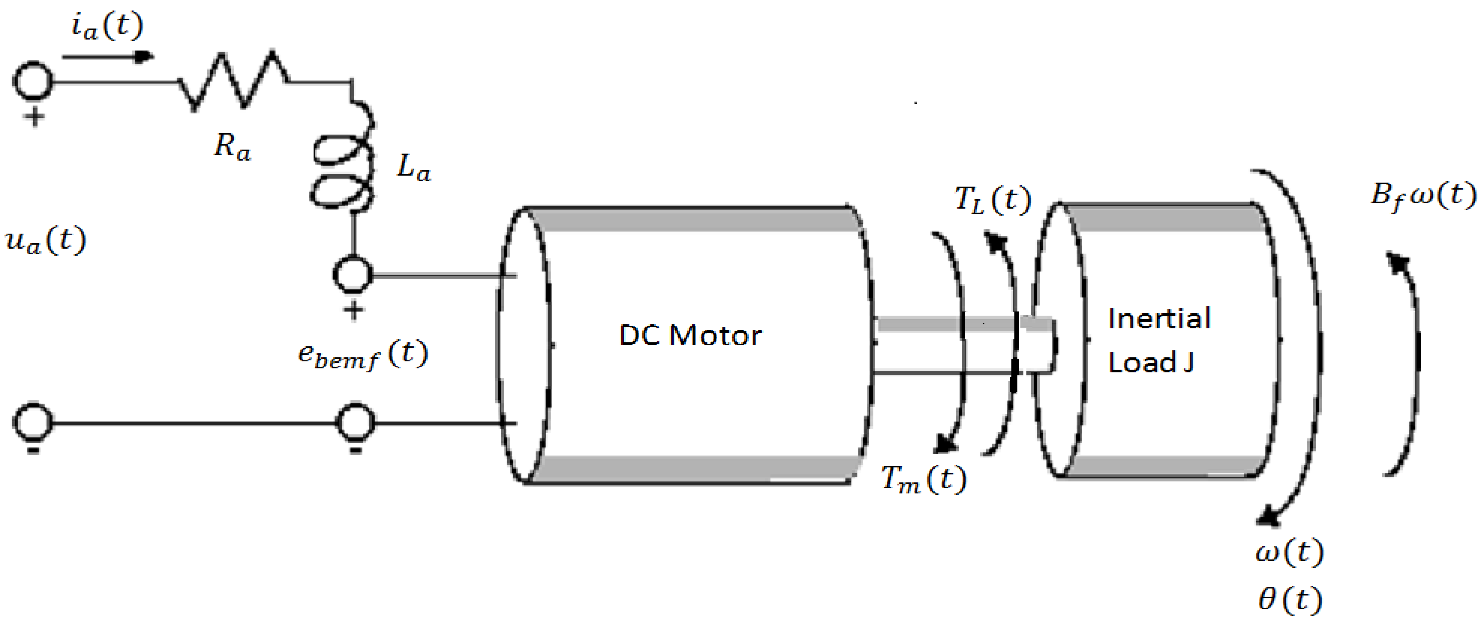
- Brushed Direct Current Motor (BDCM)
- State space analysis.
- States feedback; Ackermann's algorithm.

2 Aims

State space analysis of BDC motor;
Computing a state feedback using Ackermann's algorithm;

3 Brushed Direct Current Motors

A simple model of a DC motor driving an inertial load shows the angular rate of the load, $\omega(t)$, as the output and applied voltage, $u_a(t)$ as the input. The ultimate goal of this example is to control the angular rate by varying the applied voltage. The figure below shows a simple model of the DC motor.



The adopted model is a MIMO one, having the input vector compound of the supply voltage, $u_a(t)$ and the load torque, $T_L(t)$. The state space vector contains the current through the supply circuit, $i_a(t)$, the angular velocity $\omega(t)[\frac{rad}{sec}]$ and the rotor position, $\theta(t)[deg]$. Having the state space vector with the following form:

$x = \begin{bmatrix} i_a \\ \omega \\ \theta \end{bmatrix}$, the state space MIMO model results as:
$$\begin{cases} \dot{x} = \begin{pmatrix} -\frac{R_a}{L_a} & -\frac{K_e}{L_a} & 0 \\ \frac{K_T}{J} & -\frac{B_f}{J} & 0 \\ 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} \frac{K_{PWM}}{L_a} & 0 \\ 0 & -\frac{1}{J} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_a \\ T_L \end{pmatrix} \\ y = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} u_a \\ T_L \end{pmatrix} \end{cases}$$

The motor torque, T_m(t) and the back electromotive force, $e_{bemf}(t)$ do not explicitly appear in the final model. If speed control is the target of this simulation, the supply voltage will be considered in terms of PWM signal: $u_a(t) = K_{PWM} * u_c(t)$.

3.1 *Handling state space model using Matlab*

The model parameters for the considered process (brushed direct current motor) have the next values:

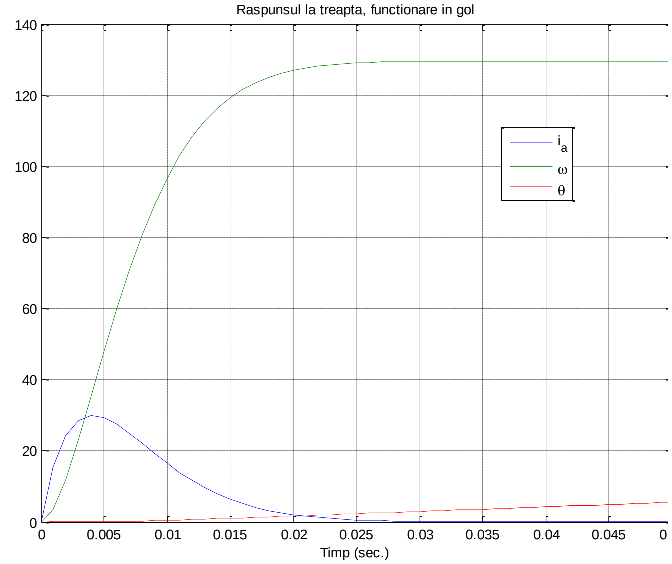
```
Ra=0.92; La=2e-3; Ke=0.2960; Kt=0.2939; Bf=3.35e-4;J=7.01e-4;Kpwm=38.4615;
```

The four matrices for state space complete representation are introduced by declaring all the elements.

```
A=[-Ra/La, -Ke/La, 0; Kt/J, -Bf/J, 0;0, 1, 0];
B=[Kpwm/La, 0; 0 -1/J;0 0];
C=[0, 1, 0];
D=[0, 0];
```

Time response simulation for a supply voltage as a unity step signal and no load variation $\{T_L(t)=0\}$ can be reach if using the next matlab script:

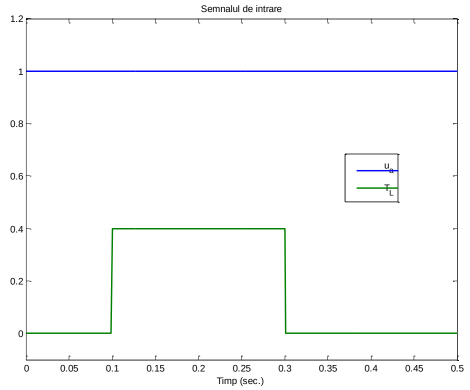
```
Tmax=max(real(abs(1/eig(A))))
% the highest time constant
T_stationar=6*Tmax;
pas=T_staţionar/100;
t=0:pas:T_stationar;

ua=ones(1,length(t));Tl=ua*0;
u=[ua;Tl];

sist=ss(A,B,C,D);
[y,t,x]=lsim(sist,u,t,[0, 0, 0]);

plot(t,x); ;grid
legend('i_a','\omega','\theta')
xlabel('Timp (sec.)');
title('No load step response')
```



3.2 *BDC Motor under load variations conditions*

Having a convenient value for the load torque T_L=0.4Nm, running the next script will offer suggestive plots for evaluating the effects of load variation over the motor physical states.

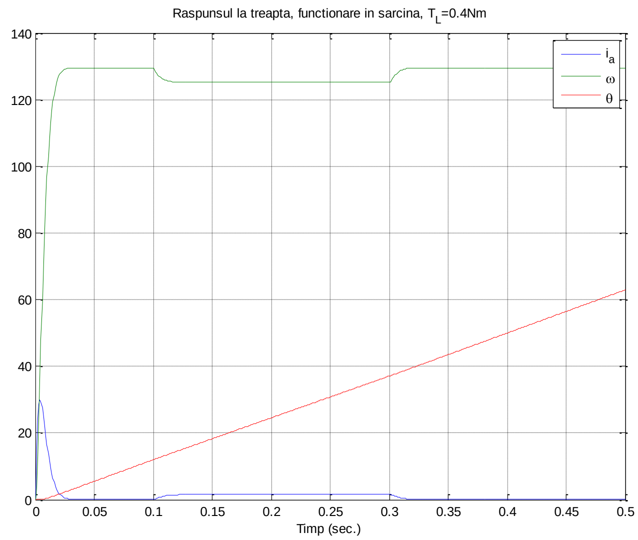
If the input signal is considered as in the figure below:

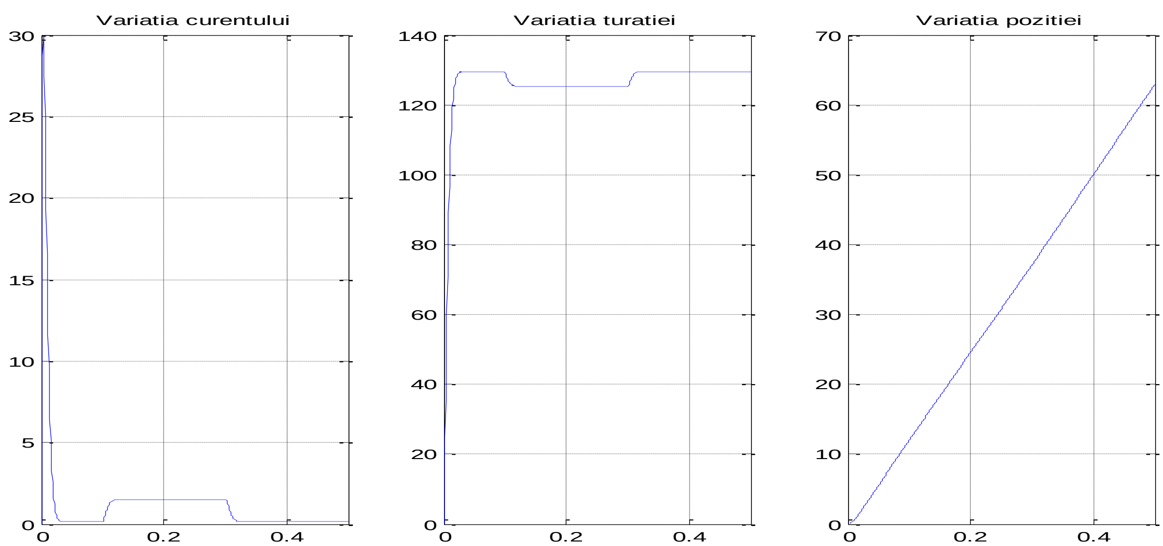


The next script will treat the load variation effects:

```
t=0:1e-3:0.5;
ua=ones(1,length(t));
Tl1=0.1;Tl2=0.3;
Tl=0.4*(zeros(1,length(t))+t<=Tl2);
u=[ua;Tl];
plot(t,u');grid;
axis([0 0.5 -0.1 1.2]);
legend('u_a','T_L');
title('Semnalul de intrare') ;
xlabel('Timp (sec.)') ;
```

In the next 4 figures the internal behaviours (the evolution of the three states) are presented.





3.3 *Eliminating the position steady state error*

To reduce the position steady state error, the open loop gain, Fx, can be computed as:

$$\varepsilon_{SS} = \lim_{t\to\infty} \varepsilon(t) = \lim_{t\to\infty} \left(\omega^{sp}(t) - \omega(t)\right) \overset{TVF}{=} \lim_{s\to 0} sL(1(t) - \omega(t)) = \lim_{s\to 0} s\left(\frac{1}{s} - H_{\omega\omega^{sp}}(s)\frac{1}{s}\right) = \lim_{s\to 0}\left(1 - H_{\omega\omega^{sp}}(s)\right) = \lim_{s\to 0}\left(1 - C(sI - A)^{-1} B_{u_c} F_x\right)$$

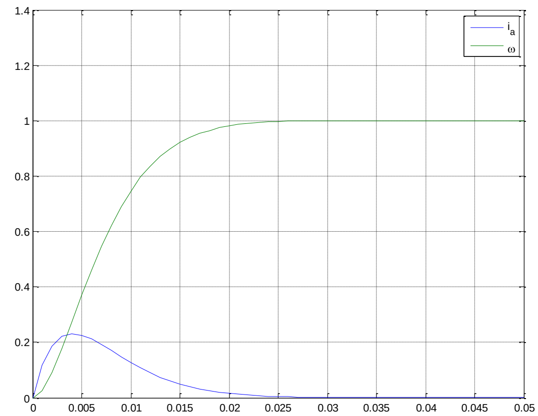
$$F_x = -\left(C(-A)B\right)^{-1} = \left(CAB\right)^{-1}$$

4 Speed control

For speed (angular velocity) control, the state space model is reduced by eliminating the third state variable (the rotor position). In this conditions, the new state space will have the next (second order) form:

```
A=[-Ra/La, -Ke/La; Kt/J, -Bf/J];
B=[Kpwm/La, 0; 0 -1/J];
C=[0, 1];
D=[0, 0];
```

The open loop gain for reducing the position steady state error can be computed using the next script



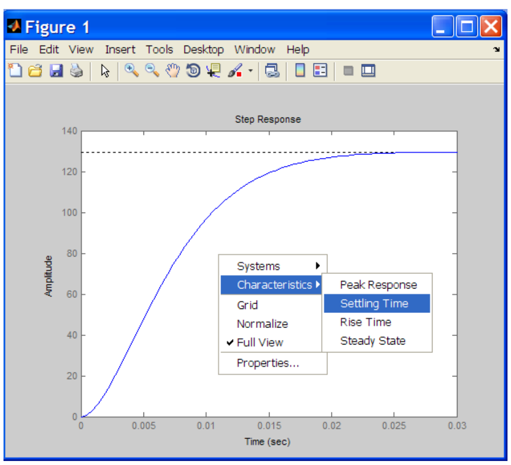

```
sist=ss(A,B,C,D);
t=0:1e-3:0.05;
u=[ones(1,length(t));zeros(1,length(t))];
Fx=inv((C*inv(-A)*B(:,1)));
[y,t,x]=lsim(sist,u*Fx,t);
plot(t,x);legend('i_a','\omega');grid
```

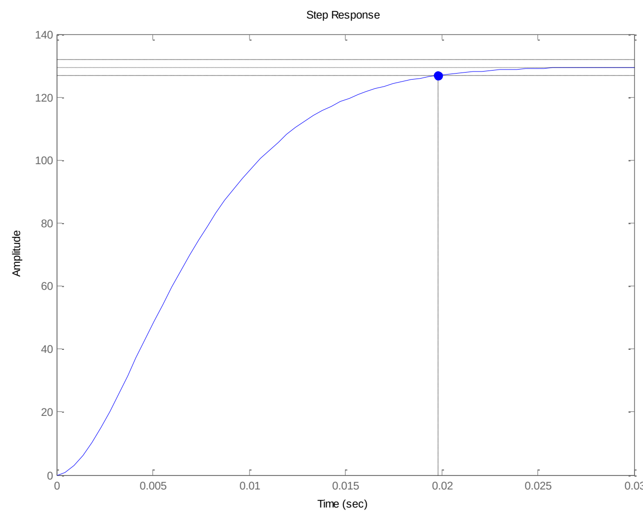
% in the figure on the left, it can be visualized the zero steady state position error.
Modify the above script if the set point for the angular velocity is 150 rad/sec. instead of 1 rad/sec.

4.1 *Computing state feedback using Ackermann's algorithm*

Using pole placement techniques, the settling time can be reduced maintaining also de restrictions regarding the nominal (operating) point. If imposing a settling time decreased by 10 % (or $t_s = 0.0178\ sec.$).

If looking to the step response (next two figures, **step(A,B,C,D,1)**), the settling time results $t_s = 0.0198\ sec.$):





Reducing the settling time by 10 % can be achieved by proper pole placement. Looking at the values of the two poles:

```
eig(A)
ans =
  1.0e+002 *
  -2.3024 + 0.9623i
  -2.3024 - 0.9623i
```

By calling the **eig** function, the dynamics is determined by a pair of complex conjugated poles having the real part equal to -230.24. If the real parts are decreased by 10% of its initial values, than the new poles can be declared as follows:

```
p=eig(A)+eig(A)/10
p =
  1.0e+002 *
  -2.5326 + 1.0585i
  -2.5326 - 1.0585i
```

Before computing the state feedback, the controllability propriety must be verified. The controllability matrix and its rank can be obtained in Matlab using the next 2 lines:

```
Co=ctrb(A,B( :, 1)) ;
R=rank(Co)
ans =
     2
```

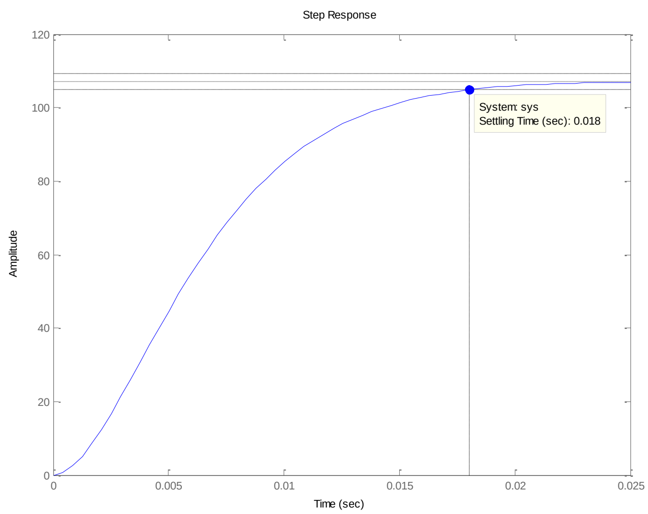
Due to the equality between the dimensions of the Co matrix and its rank, the system results fully controllable meaning that using a specific control signal the system can be passed to a desired state. If the process is fully controllable, (i.e. both states are controllable) pole placement techniques can be applied.

Using Ackermann's algorithm, the state feedback vector Kx can be obtained in Matlab as follows:

```
Kx=acker(A,B(:,1),p)
Kx =
    0.24       0.0016
```

The closed loop system with state feedback will be determined by the next four matrices:

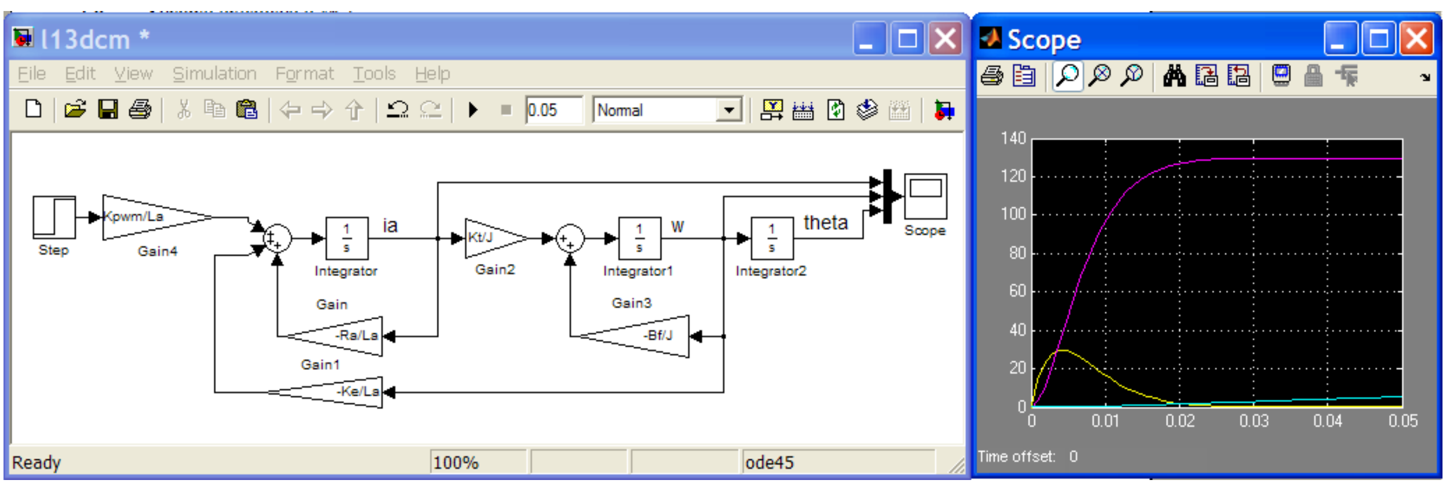
```
Ar=A-B(:,1)*Kx; Br=B;  Cr=C; Dr=D;
```



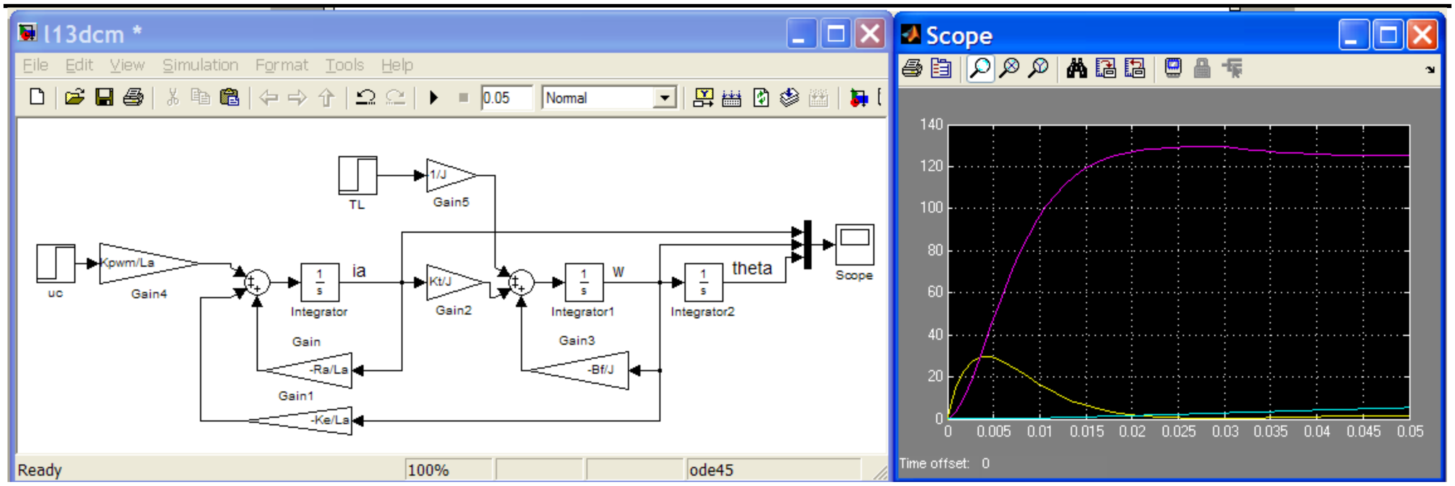
4.2 *Block diagram approach using Simulink*

For a better understanding, the block diagram representation will be used further on to accommodate to pole placement techniques.

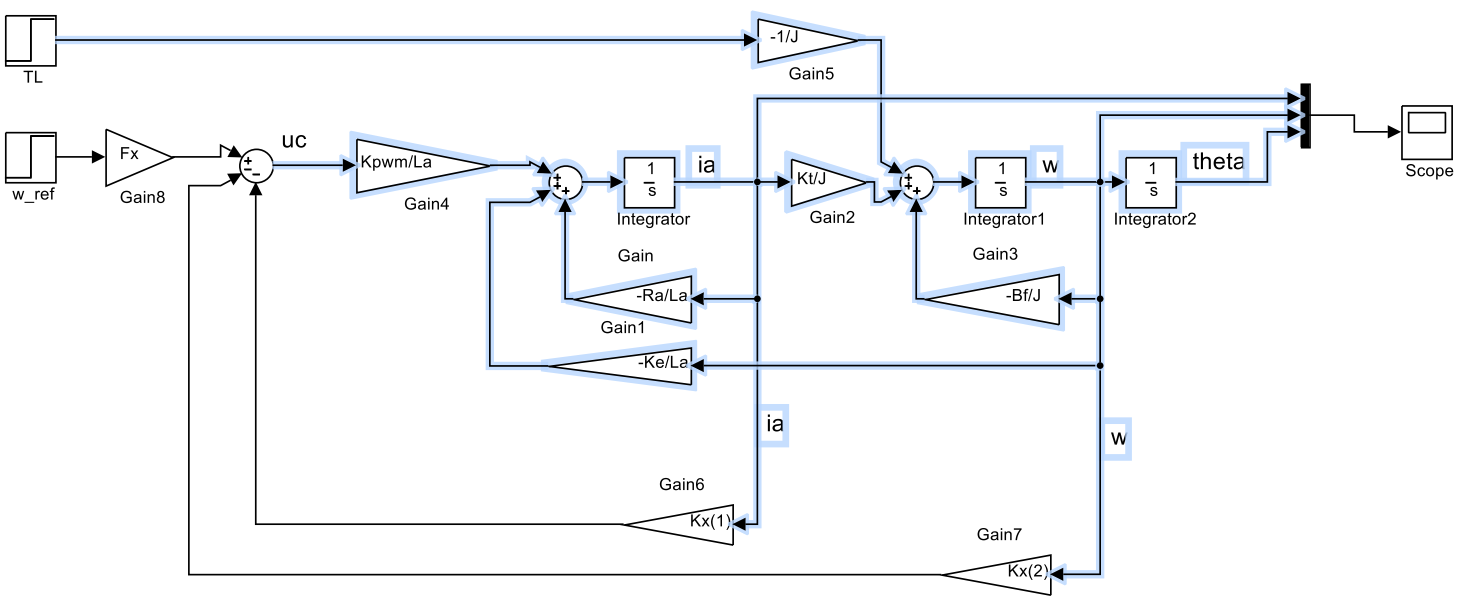
First step consist in designing the state space of the BDC motor using basic elements (integrators, summing points, multipliers by a constant). If using the last state space representation (of second order) giving access to the current and speed (angular velocity), the Simulink structure results as in the next figure:



If adding also the second input signal (load torque variation) by setting its value to -0.4 [Nm] and a step time at 0.03 sec., in the next figures the effects of load variation can be also obtained using Simulink.



In order to close to loop, select only the blocks modelling the process and create a subsystem. Add the state feedback as in the picture bellow and verify if the settling time decreases by 10 %. Using the same Simulink model, simulate and compare the evolution of the angular velocity for the open and closed loop case.



4.2.1 Questions

1. Why the steady state error is different from zero after closing the loop?
2. Which are the affected performances if changing only the imaginary part of the poles?
3. Is it possible to eliminate the perturbation (load variation) using pole placement? Justify the answer!