



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ηλεκτρονικά Ισχύος

3^η Εργασία

Λογισμικό

MATLAB

Θωμάς Χατζής 2018030134

Χρήστος Παυλόπουλος 2018030139

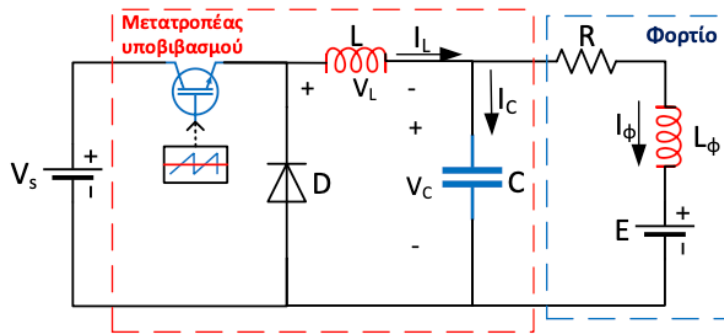
Μάριος Σαλίνας 2018030049

Ερώτημα 1

Στο Σχήμα 1 απεικονίζεται μια διάταξη μετατροπέα υποβιβασμού συνεχούς ρεύματος που τροφοδοτεί R-L φορτίο με εσωτερική τάση E. Η λειτουργία του συστήματος χωρίζεται σε δύο φάσεις, ανάλογα με την κατάσταση του διακόπτη.

Στην πρώτη φάση ο διακόπτης είναι κλειστός. Το χρονικό διάστημα που ο διακόπτης είναι κλειστός ορίζεται από το duty cycle. Η διόδος είναι ανάστροφα πολωμένη, λόγω της αντίστροφης πολικότητας που έχει με την πηγή. Η πηγή κατά την Φ1 τροφοδοτεί το πηνίο, το πυκνωτή και το φορτίο. Στην δεύτερη φάση ο διακόπτης είναι ανοικτός. Κατά την Φ2, η διόδος είναι ορθά πολωμένη, έτσι ώστε να αποφευχθεί η απότομη διακοπή του ρεύματος στο πηνίο και να υπάρχει συνεχές ρεύμα.

Παρακάτω περιγράφονται οι εξισώσεις κατά τις φάσεις λειτουργίας του κυκλώματος:



Σχήμα 1

Για την Φάση 1:

- $V_L(t) = V_s - V_c(t) = L \cdot \frac{dI_L(t)}{dt}$
- $V_c(t) = R \cdot I_\phi(t) + L_\phi \cdot \frac{dI_\phi(t)}{dt} + E$
- $I_C(t) = C \cdot \frac{dV_C(t)}{dt}$
- $I_L(t) = I_C(t) + I_\phi(t)$

Οι παραπάνω εξισώσεις μετατράπηκαν σε σύστημα πινάκων:

$$\frac{d}{dt} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1}{L} \\ 0 & -\frac{R}{L_\varphi} & \frac{1}{L_\varphi} \\ \frac{1}{C} & -\frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{L_\varphi} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s \\ E \end{bmatrix}$$

$$\begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s \\ E \end{bmatrix}$$

Για την Φάση 2:

- $-V_c(t) = L \cdot \frac{dI_L(t)}{dt}$
- $V_c(t) = R \cdot I_\varphi(t) + L_\varphi \cdot \frac{dI_\Phi(t)}{dt} + E$
- $I_C(t) = C \cdot \frac{dV_C(t)}{dt}$
- $I_L(t) = I_C(t) + I_\Phi(t)$

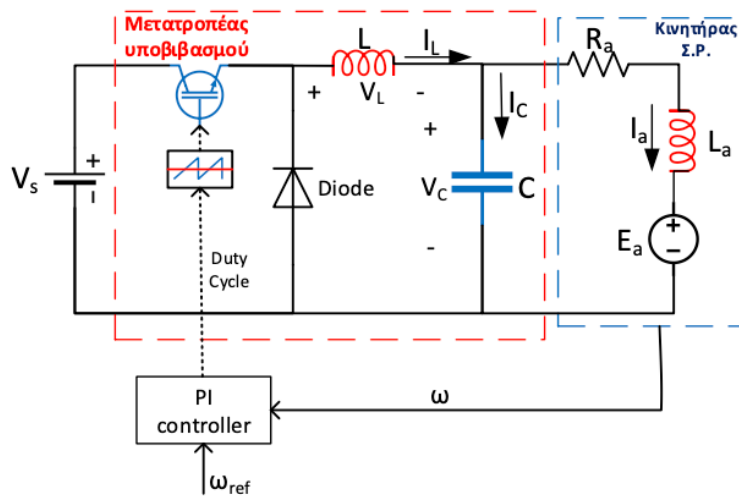
Οι παραπάνω εξισώσεις μετατράπηκαν σε σύστημα πινάκων:

$$\frac{d}{dt} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1}{L} \\ 0 & -\frac{R}{L_\varphi} & \frac{1}{L_\varphi} \\ \frac{1}{C} & -\frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{L_\varphi} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s \\ E \end{bmatrix}$$

$$\begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_L(t) \\ I_\Phi(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s \\ E \end{bmatrix}$$

Ερώτημα 2

Στο 2^ο σχήμα παρατηρείται ότι το κύκλωμα είναι ίδιο με αυτό του Σχήματος 1 με την διαφορά ότι πλέον η τάση E_a είναι μεταβλητή και υπάρχει ένας ΠΙ ελεγκτής. Η τάση E_a είναι μεταβλητή και εξαρτάται από την γωνιακή ταχύτητα ω του κινητήρα. Ο ΠΙ ελεγκτής ρυθμίζει το duty cycle του ηλεκτρονικού διακόπτη, ανάλογα με το σφάλμα που προκύπτει από την σύγκριση των στροφών του κινητήρα ω και των στροφών αναφοράς ω_{ref} , έτσι ώστε τελικά να διατηρούνται οι στροφές κατά το δυνατόν σταθερές. Έτσι, θα χρησιμοποιηθούν οι εξισώσεις για την ανάλυση και μοντελοποίηση του προηγούμενου ερωτήματος και κάποιες επιπλέον.



Σχήμα 2

Για την μοντελοποίηση του κινητήρα DC χρησιμοποιήθηκαν οι παρακάτω εξισώσεις που περιγράφουν την λειτουργία του:

- $J \frac{d\omega(t)}{dt} = T_e(t) - T_L(t)$
- $E_a(t) = k_e \cdot \omega(t)$
- $T_e(t) = k_T \cdot I_a(t)$
- $D(t) = K \cdot (\omega_{ref}(t) - \omega(t)) + K_1 \cdot \int (\omega_{ref}(t) - \omega(t)) \cdot dt$

Όπου

- ω οι στροφές του κινητήρα
- T_e η ηλεκτρική ροπή του κινητήρα
- T_L η ροπή του φορτίου
- E_a η αντιηλεκτρεγερτική δύναμη στο τύλιγμα του δρομέα
- R_a η αντίσταση του τυλίγματος οπλισμού
- L_a η αυτεπαγωγή του τυλίγματος οπλισμού
- J σταθερά αδράνειας
- k_T η σταθερά ροπής του κινητήρα
- k_e η σταθερά τάσης του κινητήρα
- D το duty cycle του διακόπτη

Τέλος, έγινε η μετατροπή του συστήματος εξισώσεων σε σύστημα διακριτού χρόνου με $T_s = 10^{-6}$, με την χρήση της συνάρτησης c2d του Matlab.

Από το σύστημα της μορφής:

$$\frac{d}{dt}x = A_2x + B_2u$$

$$y = C_2x + D_2u$$

Μετατράπηκε σε:

$$x(k+1) = A_{2,d}x(k) + B_{2,d}u(k)$$

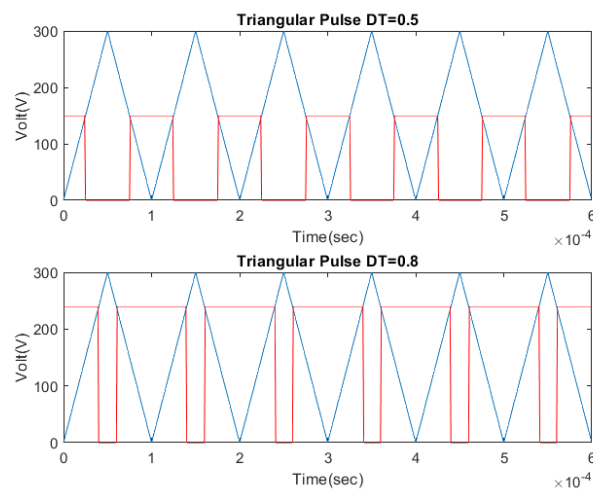
$$y(k) = C_{2,d}x(k) + D_{2,d}u(k)$$

Ερώτημα 3

Από την εκφώνηση της άσκησης δίνονται τα παρακάτω δεδομένα για το Σχήμα 1:

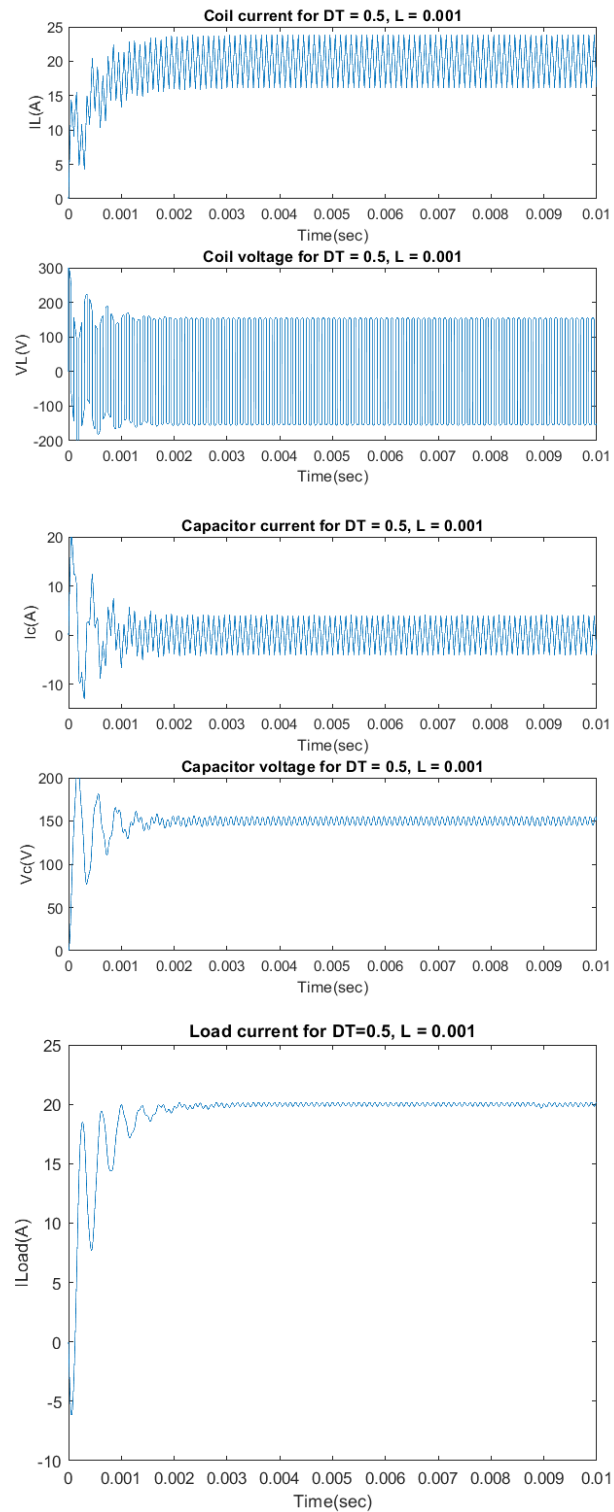
- $V_S=300V$
- $L=10^{-3}H$
- $C=10^{-5}F$
- $f=10kHz$
- $\omega_{ref} = 80 \text{ rad/s}$

Για διαφορετικό Duty Cycle 0.5 και 0.8 αντίστοιχα παρατίθενται τα παρακάτω διαγράμματα με την παρουσία τριγωνικού παλμού:

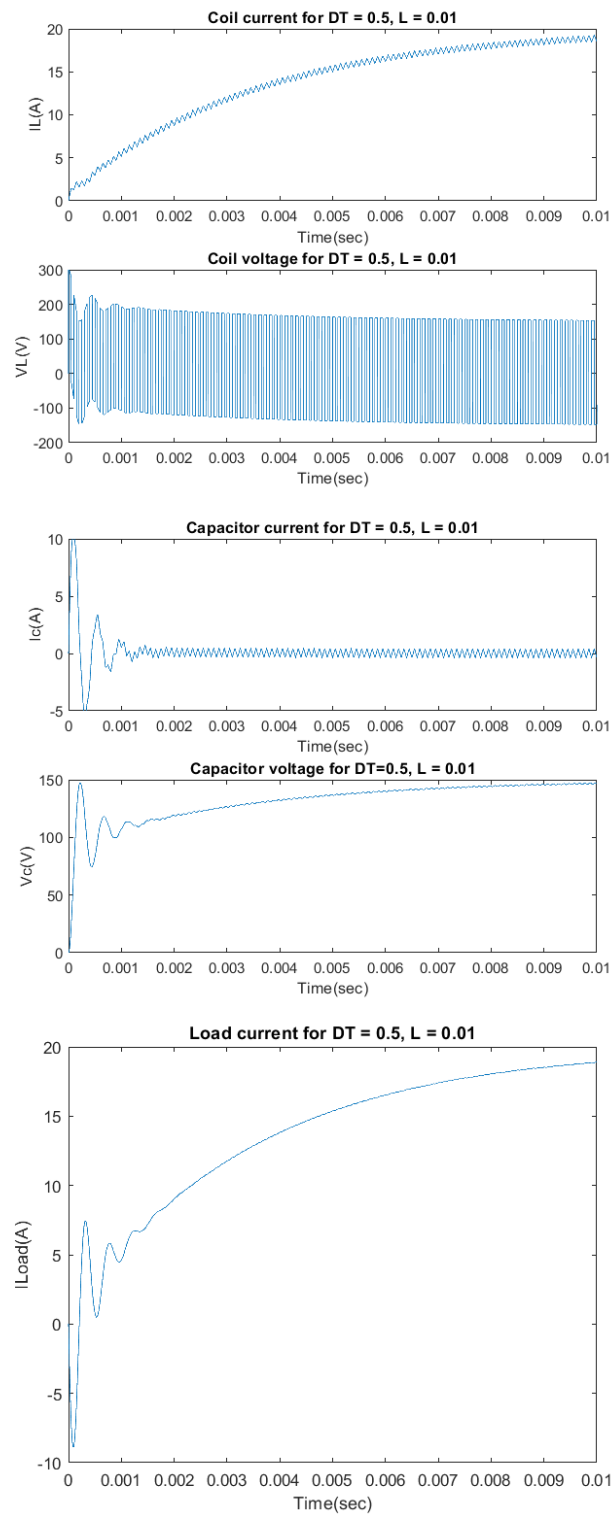


Στην συνέχεια σχεδιάστηκαν οι κυματομορφές των τάσεων και των ρευμάτων του κυκλώματος για τις παρακάτω 3 περιπτώσεις.

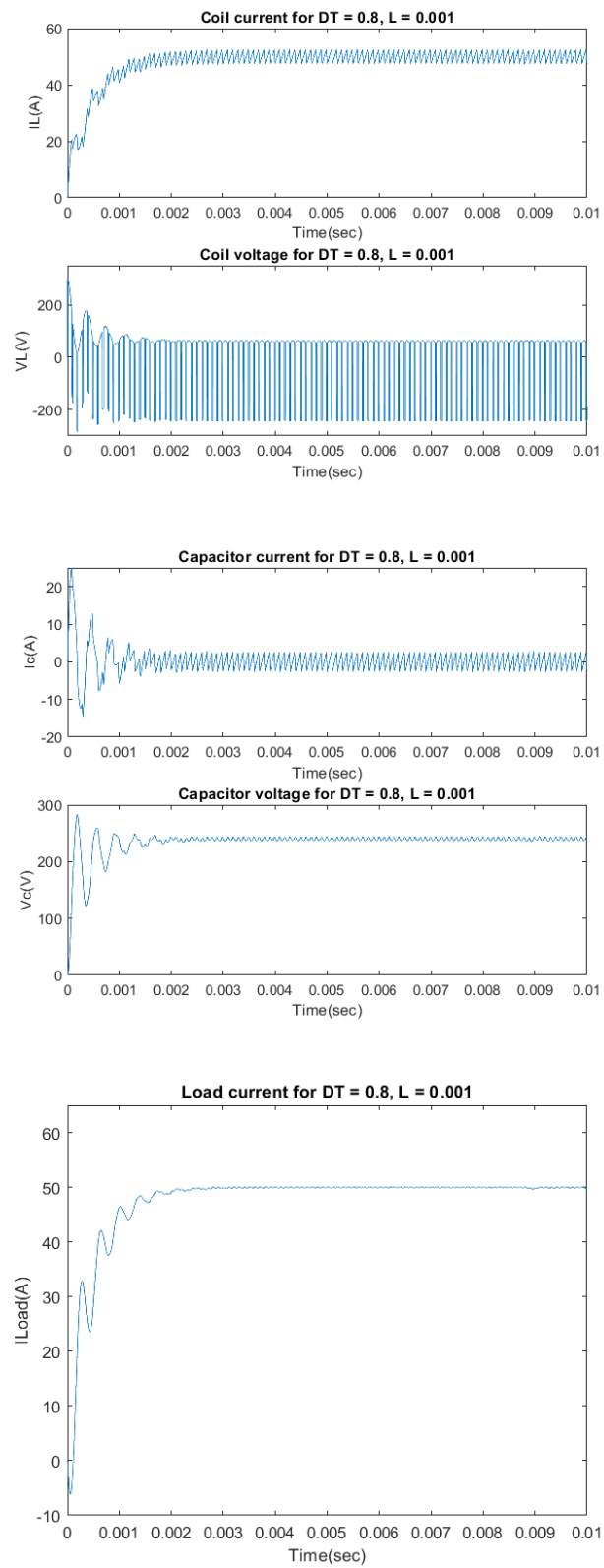
3.1 Duty Cycle = 0.5, $L = 10^{-3} H$



3.2 Duty Cycle = 0.5, $L = 10^{-2} H$



3.3 Duty Cycle = 0.8, $L = 10^{-3} H$



3.4 Σχολιασμός αποτελεσμάτων

Παρακάτω σχολιάζεται ο έλεγχος τριών περιπτώσεων διαφορετικών τιμών Duty Cycle και L για το κύκλωμα του 1^{ου} Σχήματος. Διακρίνονται δύο περιπτώσεις για τον σχολιασμό τους. Η πρώτη περίπτωση συναντάται όταν το L είναι το ίδιο αλλά το Duty Cycle διαφορετικό. Παρατηρείται ότι για μικρές τιμές του Duty Cycle υπάρχει μικρό μέγιστο πλάτος, αλλά μεγάλο πλάτος ταλάντωσης στο ρεύμα του πηνίου. Στο ρεύμα του πυκνωτή είναι φανερό ότι για μικρό Duty Cycle το πλάτος ταλάντωσης είναι μεγάλο. Όσο για το φορτίο διαπιστώνεται ότι για το μεγαλύτερο Duty Cycle το ρεύμα είναι αυξημένο και το πλάτος ταλάντωσης μικρό. Η δεύτερη περίπτωση συναντάται για διαφορετικό L και ίδιο Duty Cycle. Παρατηρείται ότι για μικρές τιμές του L το σχήμα της τάσης του πηνίου έχει μεγαλύτερη καμπυλότητα. Τέλος, για το ρεύμα του πυκνωτή διαπιστώνεται μεγάλο πλάτος ταλάντωσης για μικρές τιμές του L , ενώ για την τάση του παρατηρείται γρηγορότερη σταθεροποίηση για την μικρή τιμή του L .

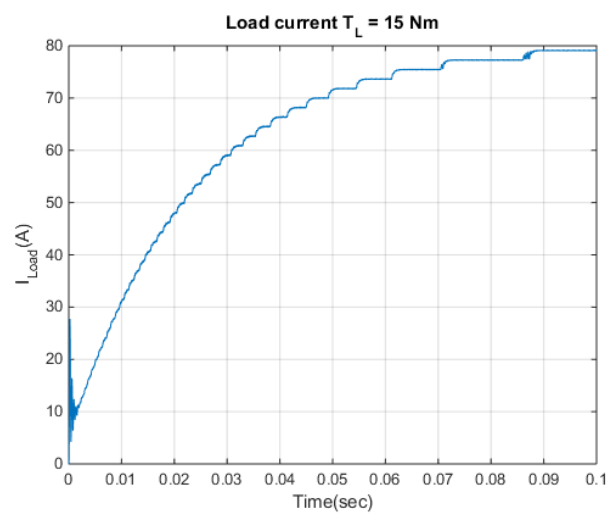
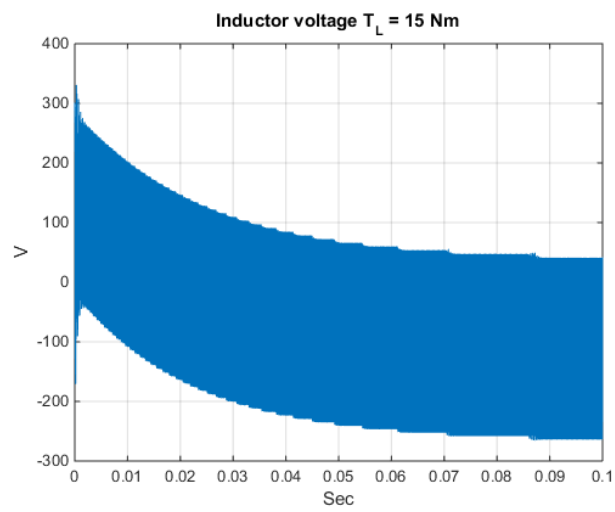
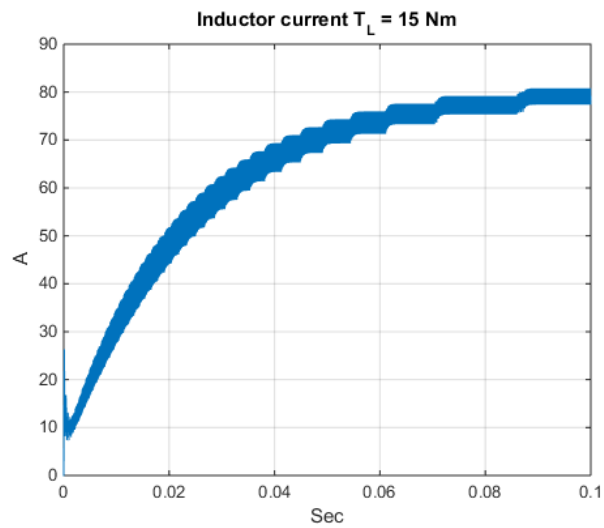
Ερώτημα 4

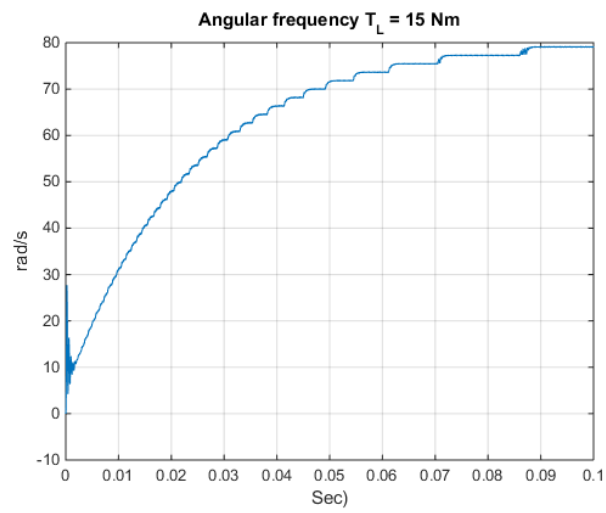
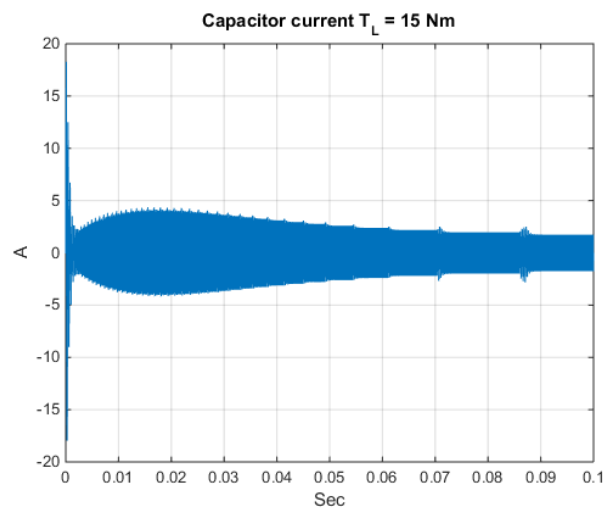
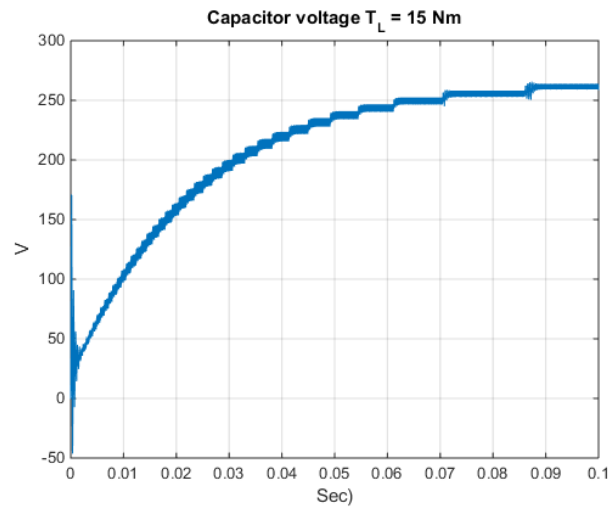
Από την εκφώνηση της άσκησης δίνονται τα παρακάτω δεδομένα για το Σχήμα 2.

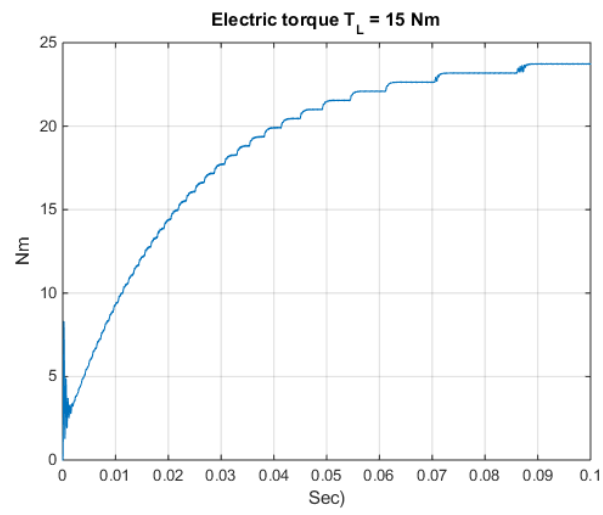
- $V_S=300V$
- $L=10^{-3}H$
- $C=10^{-5}F$
- $f=10kHz$
- $\omega_{ref} = 80 \text{ rad/s}$

Στην συνέχεια σχεδιάστηκαν οι κυματομορφές των τάσεων, των ρευμάτων, των στροφών και της ηλεκτρικής ροπής του κυκλώματος για τις παρακάτω 2 περιπτώσεις.

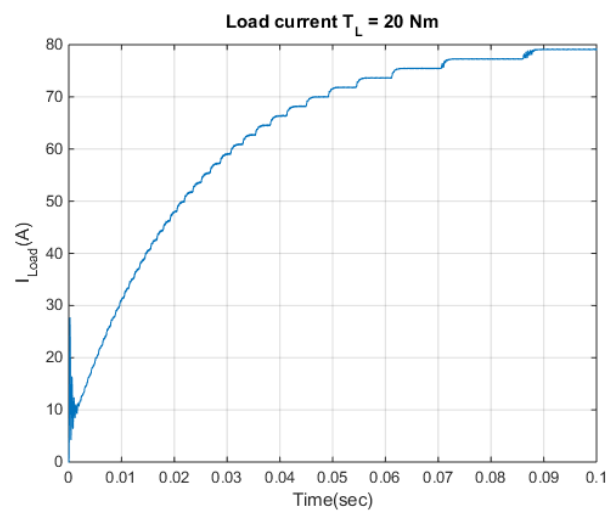
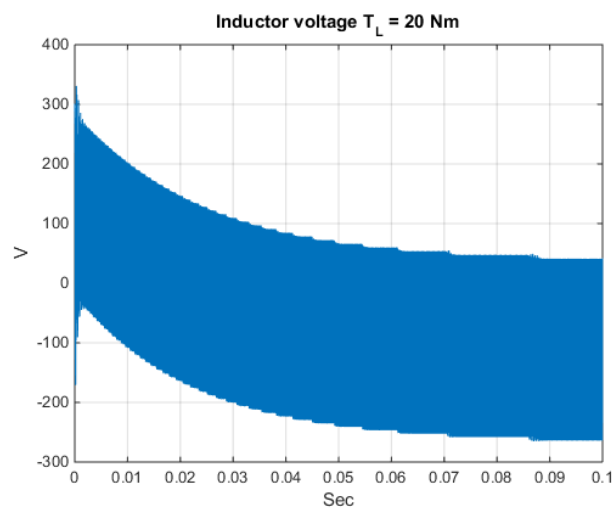
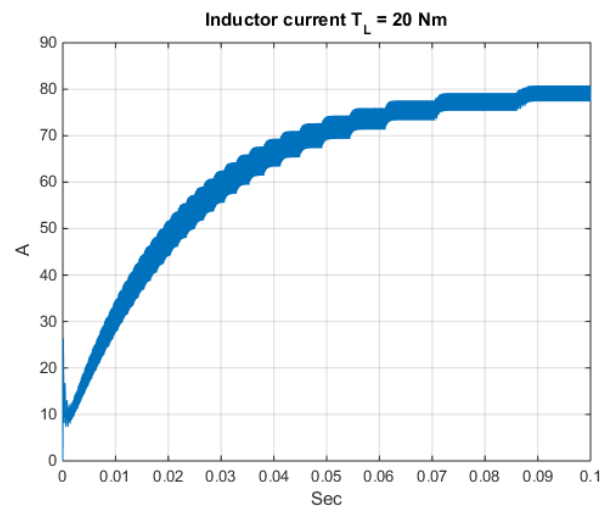
4.1 $T_L=15Nm$, αρχικές στροφές κινητήρα $\omega=0\text{ rad/s}$

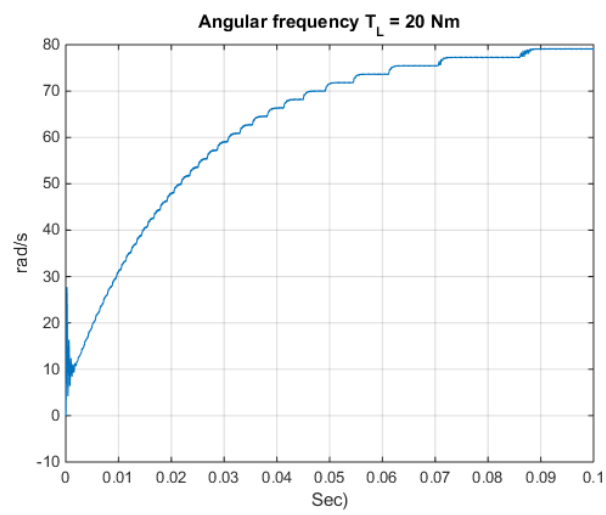
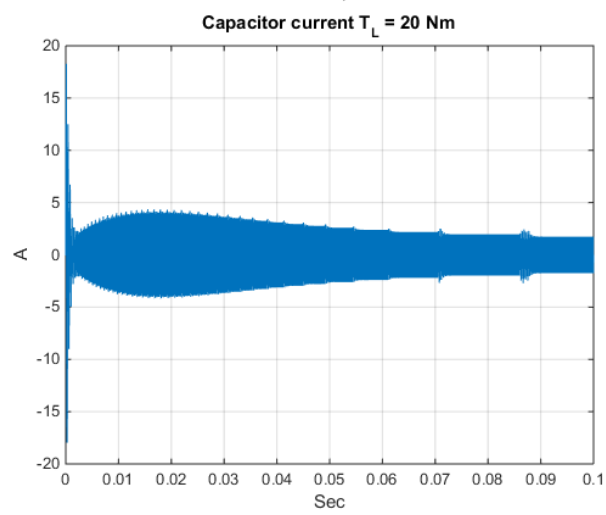
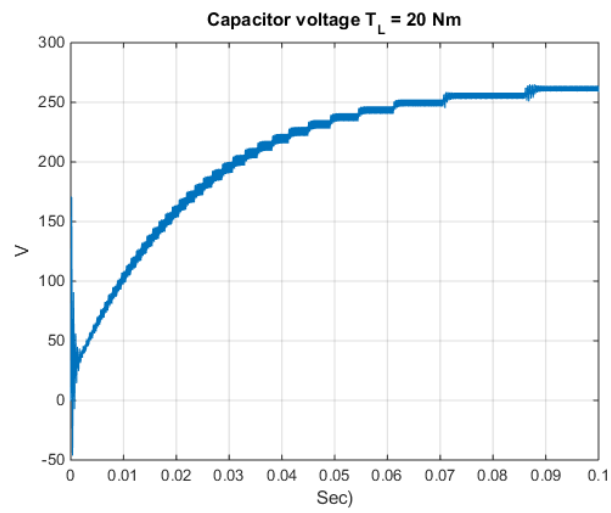


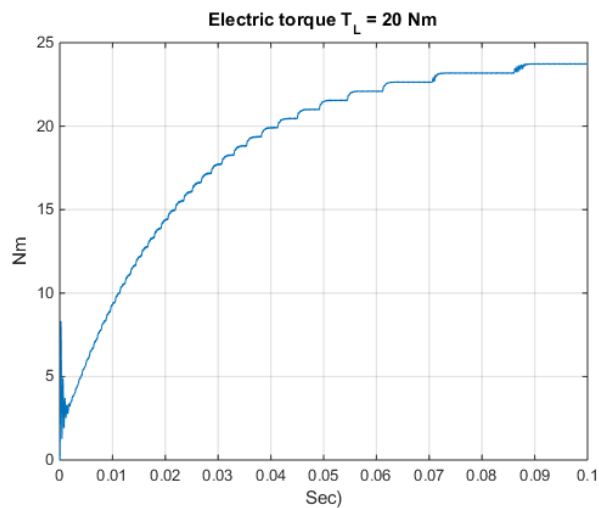




4.2 $T_L=20Nm$, αρχικές στροφές κινητήρα $\omega=0\text{ rad/s}$







4.4 Σχολιασμός Αποτελεσμάτων

Παρακάτω σχολιάζεται ο έλεγχος δύο περιπτώσεων διαφορετικών τιμών ροπής φορτίου για το κύκλωμα του 2^{ου} Σχήματος. Στην 1^η περίπτωση η ροπή είναι ίση με 15 Nm και το σύστημα έχει σταθερές τιμές ρευμάτων και τάσεων στα στοιχεία του κυκλώματος στην μόνιμη του κατάσταση. Αυξάνοντας την ροπή στα 20 Nm στην 2^η περίπτωση, παραμένουν σταθερές οι τιμές του κυκλώματος απλά είναι αυξημένες σε σύγκριση με το προηγούμενο ερώτημα έτσι ώστε να μπορέσει να ανταποκριθεί το σύστημα στο μεγαλύτερο φορτίο. Τέλος, πρέπει να σημειωθεί ότι αύξηση της ροπής του φορτίου έχει επίπτωση στις στροφές του κινητήρα, καθώς αυτές μειώνονται.

Ερώτημα 5

Κώδικας Matlab

Ερώτημα 3

```
close all;
clear all;
clc;

%% Square pulse generation
Fs = 10000;
T = 6*1/Fs;
t = 0:1/(100*Fs):T-1/(100*Fs);

pulse_tr = 150*sawtooth(2*pi*Fs*t,0.5)+150;
pulse_1 = 150*ones(600,1);
pulse_2 = 240*ones(600,1);

for i = 1:1:600
    %DT=0.5
    if pulse_tr(i)>= pulse_1(i)
        Vdc_1(i)= 0;
    else
        Vdc_1(i)= pulse_1(i);
    end

    %DT=0.8
    if pulse_tr(i)>= pulse_2(i)
        Vdc_2(i)= 0;
    else
        Vdc_2(i)= pulse_2(i);
    end
end

figure;
subplot(2,1,1);
plot(t,pulse_tr)
hold on;
plot(t,Vdc_1,'r')
title('Triangular Pulse DT=0.5');
xlabel('Time(sec)');
ylabel('Volt(V)');

subplot(2,1,2);
plot(t,pulse_tr)
hold on;
plot(t,Vdc_2,'r')
title('Triangular Pulse DT=0.8');
xlabel('Time(sec)');
ylabel('Volt(V)');

%% DT = 0.5, L = 0.001

%Parameters
R = 3;           %Resistance
L = 0.0005;      %Inductor
C = 10^(-5);     %Capacitor
L1 = 10^(-3);    %Inductor A
L2 = 10^(-2);    %Inductor B
L3 = 10^(-3);    %Inductor C

%State space model
%Matrixes
A1=[0 0 -1/L1 ; 0 -R/L 1/L ; 1/C -1/C 0 ];
```

```

B1=[1/L1 0 ; 0 -1/L ; 0 0];

A2=[0 0 -1/L2 ; 0 -R/L 1/L ; 1/C -1/C 0 ];
B2=[1/L2 0 ; 0 -1/L ; 0 0];

A3=[0 0 -1/L3 ; 0 -R/L 1/L ; 1/C -1/C 0 ];
B3=[1/L3 0 ; 0 -1/L ; 0 0];

%Matrix B for phase 2
B = [0 0 ; 0 -1/L ; 0 0];
%C and D stay the same
C=[1 0 0; 0 1 0; 0 0 1];
D=[0 0 ; 0 0 ; 0 0];

%Phase 1 discrete system
sys1_1 = ss(A1,B1,C,D);
sys1_1_dis=c2d(sys1_1,10^(-6));

A11_dis = sys1_1_dis.A;
B11_dis = sys1_1_dis.B;
C11_dis = sys1_1_dis.C;
D11_dis = sys1_1_dis.D;

%Phase 2 discrete system
sys1_2 = ss(A1,B,C,D);
sys1_2_dis=c2d(sys1_2,10^(-6));

A12_dis = sys1_2_dis.A;
B12_dis = sys1_2_dis.B;
C12_dis = sys1_2_dis.C;
D12_dis = sys1_2_dis.D;

%Initializing everything
V = 90*ones(10000,1);
Vs = 300*ones(10000,1);
Vd = -300*ones(10000,1);
Id = 0*ones(10000,1);
VL = 0*ones(10000,1);
Vc(1)=0;
If(1)=0;
Il(1)=0;
t = 0;

%for-loop
for p = 1:1:10000

    if((mod(t,1/Fs))<=1/(Fs*2))
        %Phase 1 calculation
        Il(p+1) = A11_dis(1)*Il(p)+ A11_dis(4)*If(p) + A11_dis(7)*Vc(p) +
        B11_dis(1)*Vs(p) + B11_dis(4)*V(p);
        If(p+1) = A11_dis(2)*Il(p)+ A11_dis(5)*If(p) + A11_dis(8)*Vc(p) +
        B11_dis(2)*Vs(p) + B11_dis(5)*V(p);
        Vc(p+1) = A11_dis(3)*Il(p)+ A11_dis(6)*If(p) + A11_dis(9)*Vc(p) +
        B11_dis(3)*Vs(p) + B11_dis(6)*V(p);
        Vd(p+1) = -300;
        Id(p+1) = 0;
        VL(p+1) = Vs(p+1) - Vc(p+1);
    else
        %Phase 2 calculation
        Il(p+1) = A12_dis(1)*Il(p)+ A12_dis(4)*If(p) + A12_dis(7)*Vc(p) +
        B12_dis(1)*Vs(p) + B12_dis(4)*V(p);
        If(p+1) = A12_dis(2)*Il(p)+ A12_dis(5)*If(p) + A12_dis(8)*Vc(p) +
        B12_dis(2)*Vs(p) + B12_dis(5)*V(p);
        Vc(p+1) = A12_dis(3)*Il(p)+ A12_dis(6)*If(p) + A12_dis(9)*Vc(p) +
        B12_dis(3)*Vs(p) + B12_dis(6)*V(p);
        Vd(p+1) = 0;
        Id(p+1) =Il(p+1);
    end
end

```

```

        VL(p+1) = -Vc(p+1);
    end;
    t= t+ 1/(100*Fs);
end;

t = 0:1/(100*Fs):10000/(100*Fs);

%Displaying requested plots
figure;
subplot(2,1,1);
plot(t,IL);
title('Coil current for DT = 0.5, L = 0.001');
axis([0 0.01 0 25]);
xlabel('Time(sec)');
ylabel('IL(A)');

subplot(2,1,2);
plot(t,VL);
title('Coil voltage for DT = 0.5, L = 0.001');
axis([0 0.01 -200 300]);
xlabel('Time(sec)');
ylabel('VL(V)');

figure;
subplot(2,1,1);
plot(t,IL-If);
title('Capacitor current for DT = 0.5, L = 0.001');
axis([0 0.01 -15 20]);
xlabel('Time(sec)');
ylabel('Ic(A)');

subplot(2,1,2);
plot(t,Vc);
title('Capacitor voltage for DT = 0.5, L = 0.001');
axis([0 0.01 0 200]);
xlabel('Time(sec)');
ylabel('Vc(V)');

figure;
plot(t,If);
title('Load current for DT=0.5, L = 0.001');
axis([0 0.01 -10 25]);
xlabel('Time(sec)');
ylabel('ILoad(A)');

%% DT = 0.5, L = 0.01

%State space model
%Phase 1 discrete system
sys2_1 = ss(A2,B2,C,D);
sys2_1_dis=c2d(sys2_1,10^(-6));

A21_dis = sys2_1_dis.A;
B21_dis = sys2_1_dis.B;
C21_dis = sys2_1_dis.C;
D21_dis = sys2_1_dis.D;

%Phase 2 discrete system
sys2_2 = ss(A2,B,C,D);
sys2_2_dis=c2d(sys2_2,10^(-6));

A22_dis = sys2_2_dis.A;
B22_dis = sys2_2_dis.B;
C22_dis = sys2_2_dis.C;
D22_dis = sys2_2_dis.D;

%Initializing
V = 90*ones(20000,1);

```

```

Vs = 300*ones(20000,1);
Vd = -300*ones(20000,1);
Id = 0*ones(20000,1);
VL = 0*ones(20000,1);
Vc(1)=0;
If(1)=0;
Il(1)=0;
t = 0;

for p = 1:1:20000
    if((mod(t,1/Fs))<=1/(Fs*2))
        %Phase 1 calculations
        Il(p+1) = A21_dis(1)*Il(p)+ A21_dis(4)*If(p) + A21_dis(7)*Vc(p) +
B21_dis(1)*Vs(p) + B21_dis(4)*V(p);
        If(p+1) = A21_dis(2)*Il(p)+ A21_dis(5)*If(p) + A21_dis(8)*Vc(p) +
B21_dis(2)*Vs(p) + B21_dis(5)*V(p);
        Vc(p+1) = A21_dis(3)*Il(p)+ A21_dis(6)*If(p) + A21_dis(9)*Vc(p) +
B21_dis(3)*Vs(p) + B21_dis(6)*V(p);
        Vd(p+1) = -300;
        Id(p+1) = 0;
        VL(p+1) = Vs(p+1) - Vc(p+1);
    else
        %Phase 2 calculations
        Il(p+1) = A22_dis(1)*Il(p)+ A22_dis(4)*If(p) + A22_dis(7)*Vc(p) +
B22_dis(1)*Vs(p) + B22_dis(4)*V(p);
        If(p+1) = A22_dis(2)*Il(p)+ A22_dis(5)*If(p) + A22_dis(8)*Vc(p) +
B22_dis(2)*Vs(p) + B22_dis(5)*V(p);
        Vc(p+1) = A22_dis(3)*Il(p)+ A22_dis(6)*If(p) + A22_dis(9)*Vc(p) +
B22_dis(3)*Vs(p) + B22_dis(6)*V(p);
        Vd(p+1) = 0;
        Id(p+1) = Il(p+1);
        VL(p+1) = -Vc(p+1);
    end;
    t= t+ 1/(100*Fs);
end;

t = 0:1/(100*Fs):20000/(100*Fs);

%Displaying requested plots
figure;
subplot(2,1,1);
plot(t,Il);
title('Coil current for DT = 0.5, L = 0.01');
axis([0 0.01 0 20]);
xlabel('Time(sec)');
ylabel('IL(A)');

subplot(2,1,2);
plot(t,VL);
title('Coil voltage for DT = 0.5, L = 0.01');
axis([0 0.01 -200 300]);
xlabel('Time(sec)');
ylabel('VL(V)');

figure;
subplot(2,1,1);
plot(t,Il-If);
title('Capacitor current for DT = 0.5, L = 0.01');
axis([0 0.01 -5 10]);
xlabel('Time(sec)');
ylabel('Ic(A)');

subplot(2,1,2);
plot(t,Vc);
title('Capacitor voltage for DT=0.5, L = 0.01');
axis([0 0.01 0 150]);
xlabel('Time(sec)');
ylabel('Vc(V)');

```

```

figure;
plot(t,If);
title('Load current for DT = 0.5, L = 0.01');
axis([0 0.01 -10 20]);
xlabel('Time(sec) ');
ylabel(' ILoad(A) ');

%% DT = 0.8, L = 0.001

%State space model
%Phase 1 discrete system
sys3_1 = ss(A3,B3,C,D);
sys3_1_dis=c2d(sys3_1,10^(-6));

A31_dis = sys3_1_dis.A;
B31_dis = sys3_1_dis.B;
C31_dis = sys3_1_dis.C;
D31_dis = sys3_1_dis.D;

%Phase 2 discrete system
sys3_2 = ss(A3,B,C,D);
sys3_2_dis=c2d(sys3_2,10^(-6));

A32_dis = sys3_2_dis.A;
B32_dis = sys3_2_dis.B;
C32_dis = sys3_2_dis.C;
D32_dis = sys3_2_dis.D;

%Initializing
V = 90*ones(20000,1);
Vs = 300*ones(20000,1);
Vd = -300*ones(20000,1);
Id = 0*ones(20000,1);
VL = 0*ones(20000,1);
Vc(1)=0;
If(1)=0;
Il(1)=0;
t = 0;

for p = 1:1:20000

    if((mod(t,1/Fs))<=(1/Fs)*0.8)
        %Phase 1 calculations
        Il(p+1) = A31_dis(1)*Il(p)+ A31_dis(4)*If(p) + A31_dis(7)*Vc(p) +
        B31_dis(1)*Vs(p) + B31_dis(4)*V(p);
        If(p+1) = A31_dis(2)*Il(p)+ A31_dis(5)*If(p) + A31_dis(8)*Vc(p) +
        B31_dis(2)*Vs(p) + B31_dis(5)*V(p);
        Vc(p+1) = A31_dis(3)*Il(p)+ A31_dis(6)*If(p) + A31_dis(9)*Vc(p) +
        B31_dis(3)*Vs(p) + B31_dis(6)*V(p);
        Vd(p+1) = -300;
        Id(p+1) = 0;
        VL(p+1) = Vs(p+1) - Vc(p+1);

    else
        %Phase 2 calculations
        Il(p+1) = A32_dis(1)*Il(p)+ A32_dis(4)*If(p) + A32_dis(7)*Vc(p) +
        B32_dis(1)*Vs(p) + B32_dis(4)*V(p);
        If(p+1) = A32_dis(2)*Il(p)+ A32_dis(5)*If(p) + A32_dis(8)*Vc(p) +
        B32_dis(2)*Vs(p) + B32_dis(5)*V(p);
        Vc(p+1) = A32_dis(3)*Il(p)+ A32_dis(6)*If(p) + A32_dis(9)*Vc(p) +
        B32_dis(3)*Vs(p) + B32_dis(6)*V(p);
        Vd(p+1) = 0;
        Id(p+1) = Il(p+1);
        VL(p+1) = -Vc(p+1);
    end;
    t= t+ 1/(100*Fs);
end;

```

```

t = 0:1/(100*Fs):20000/(100*Fs);

%Displaying requested plots
figure;
subplot(2,1,1);
plot(t,Il);
title('Coil current for DT = 0.8, L = 0.001');
axis([0 0.01 0 60]);
xlabel('Time(sec)');
ylabel('IL(A)');

subplot(2,1,2);
plot(t,VL);
title('Coil voltage for DT = 0.8, L = 0.001');
axis([0 0.01 -300 350]);
xlabel('Time(sec)');
ylabel('VL(V)');

figure;
subplot(2,1,1);
plot(t,Il-If);
title('Capacitor current for DT = 0.8, L = 0.001');
axis([0 0.01 -20 25]);
xlabel('Time(sec)');
ylabel('Ic(A)');

subplot(2,1,2);
plot(t,Vc);
title('Capacitor voltage for DT = 0.8, L = 0.001');
axis([0 0.01 0 300]);
xlabel('Time(sec)');
ylabel('Vc(V)');

figure;
plot(t,If);
title('Load current for DT = 0.8, L = 0.001');
axis([0 0.01 -10 65]);
xlabel('Time(sec)');
ylabel('ILoad(A)');

```

Ερώτημα 4

Υποερώτημα 4.1

```

%% 4.1
clear all;
close all;
clc;

Vs = 300; % Source voltage
C = 0.00001; % Capacitance
R = 3; % Resistance
L = 0.001; % Induction
fs = 10000; % Frequency
Ts = 1/fs; % Period

dt = 0.000001; % Step
n = 1000; % Amount of periods
t = 0:dt:n*Ts-dt; % Time

Kp = 0.001;
Ki = 0.5;

```

```

%Motor parameters
Ra = 3; % Resistance
La = 0.0005; % Induction
J = 0.005; % kgm^2
Kt = 0.3; % Nm/A
Ke = 0.3; % v/rad/s
TL = 15; % Torque
omega = 80; % Reference value

pulse = (sawtooth(2*pi*fs*t,(1/2))+1)/2;

% Phase 1 state space
A1 = [0 0 -1/L; 0 -Ra/La 1/La; 1/C -1/C 0];
B1 = [1/L 0; 0 -1/La; 0 0];
C1 = eye(3);
D1 = zeros(3,2);

sys1 = ss(A1,B1,C1,D1);
sys_dis1 = c2d(sys1,dt);

sysa = sys_dis1.A;
sysb = sys_dis1.B;
sysc = sys_dis1.C;
sysd = sys_dis1.D;

% Phase 2 state space
A2 = [0 0 -1/L; 0 -Ra/La 1/La; 1/C -1/C 0];
B2 = [0 0; 0 -1/La; 0 0];
C2 = eye(3);
D2 = zeros(3,2);

sys2 = ss(A2,B2,C2,D2);
sys_dis2 = c2d(sys2,dt);

sys2a = sys_dis2.A;
sys2b = sys_dis2.B;
sys2c = sys_dis2.C;
sys2d = sys_dis2.D;

% Motor state space
A_motor = Kt/J;
B_motor = -1/J;
C_motor = Ke;
D_motor = 0;

sysmotor = ss(A_motor,B_motor,C_motor,D_motor);
sys_dismotor = c2d(sysmotor,dt);

motora = sys_dismotor.A;
motorb = sys_dismotor.B;
motorc = sys_dismotor.C;
motord = sys_dismotor.D;

% Controller state space
Ac = 0;
Bc = 1;
Cc = Ki;
Dc = Kp;

sysc = ss(Ac,Bc,Cc,Dc);
sysc_dis = c2d(sysc,dt);

cona=sysc_dis.A;
conb=sysc_dis.B;
conc=sysc_dis.C;
cond=sysc_dis.D;

```

```

x1 = 0;
x2 = 0;
g = 100;
v = 0;
j = 1;

% Helping vector
IL = zeros(size(t));
VL = zeros(size(t));
Va = zeros(size(t));
Ia = zeros(size(t));
Vc = zeros(size(t));
Ic = zeros(size(t));
Te = zeros(size(t));
wm = zeros(size(t));
error = zeros(size(t));
x = zeros(size(t));
duty = zeros(size(n));
step_p = zeros(size(g));

for k=0:1:n-1
    if (j==1)
        % Generating pulse
        for i=1:g
            if(pulse(i)>= duty(j))
                step_p(i) = min(pulse);
            else
                step_p(i) = max(pulse);
            end
        end
    else
        for i=1:g
            if(pulse(i)>= duty(j-1))
                step_p(i) = min(pulse);
            else
                step_p(i) = max(pulse);
            end
        end
    end

    % Calculating current and voltages
    for u = 1:Ts/dt
        if(step_p(u) == max(step_p))
            % phase 1
            IL(u+1+v) = sysa(1,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(1,:)*[Vs; Va(u+v)];
            Ia(u+1+v) = sysa(2,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(2,:)*[Vs; Va(u+v)];
            Ic(u+v) = IL(u+v) - Ia(u+v);
            Vc(u+1+v) = sysa(3,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(3,:)*[Vs; Va(u+v)];
            VL(u+v) = Vs - Vc(u+v);

        else
            % phase 2
            IL(u+1+v) = sys2a(1,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(1,:)*[Vs; Va(u+v)];
            Ia(u+1+v) = sys2a(2,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(2,:)*[Vs; Va(u+v)];
            Ic(u+v) = IL(u+v) - Ia(u+v);
            Vc(u+1+v) = sys2a(3,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(3,:)*[Vs; Va(u+v)];
            VL(u+v) = - Vc(u+v);

        end
    end
end

```



```

    %omega computing
    Te(u+v) = Kt*Ia(u+v);
    wm(u+1+v) = motora*Ia(u+v) + motorb*TL;
    Va(u+1+v) = motorc*wm(u+1+v);

    % error computing
    error(u+v) = wm(u+v) - omega;
    x(u+1+v) = cona*x(u+v) + conb*error(u+v);

    if(u == g)
        x1 = x(u+v);
        x2 = error(u+v);
    end

end

duty(j) = abs(conc*x1 + cond*x2);

%incrementing constants
v = v+g;
j = j+1;
end

% Plots
figure;
plot(t,IL(1:end-1));
grid on;
xlabel('Sec');
ylabel('A');
title('Inductor current T_L = 15 Nm');

figure;
plot(t,VL);
grid on;
xlabel('Sec');
ylabel('V');
title('Inductor voltage T_L = 15 Nm');

figure;
plot(t,Ia(1:end-1));
grid on;
xlabel('Time(sec)');
ylabel('I_{Load} (A)');
title('Load current T_L = 15 Nm');

figure;
plot(t,Vc(1:end-1));
grid on;
xlabel('Sec');
ylabel('V');
title('Capacitor voltage T_L = 15 Nm');

figure;
plot(t,Ic);
grid on;
xlabel('Sec');
ylabel('A');
title('Capacitor current T_L = 15 Nm');

figure;
plot(t,wm(1:end-1));
grid on;
xlabel('Sec');
ylabel('rad/s');
title('Angular frequency T_L = 15 Nm');

```

```

figure;
plot(t,Te);
grid on;
xlabel('Sec');
ylabel('Nm');
title('Electric torque T_L = 15 Nm');

```

Υποερώτημα 4.2

```

%% 4.2
clear all;
close all;
clc;

Vs = 300; % Source voltage
C = 0.00001; % Capacitance
R = 3; % Resistance
L = 0.001; % Induction
fs = 10000; % Frequency
Ts = 1/fs; % Period

dt = 0.000001; % Step
n = 1000; % Amount of periods
t = 0:dt:n*Ts-dt; % Time

Kp = 0.001;
Ki = 0.5;

%Motor parameters
Ra = 3; % Resistance
La = 0.0005; % Induction
J = 0.005; % kgm^2
Kt = 0.3; % Nm/A
Ke = 0.3; % v/rad/s
TL = 20; % Torque
omega = 80; % Reference value

pulse = (sawtooth(2*pi*fs*t, (1/2))+1)/2;

% Phase 1 state space
A1 = [0 0 -1/L; 0 -Ra/La 1/La; 1/C -1/C 0];
B1 = [1/L 0; 0 -1/La; 0 0];
C1 = eye(3);
D1 = zeros(3,2);

sys1 = ss(A1,B1,C1,D1);
sys_dis1 = c2d(sys1,dt);

sysa = sys_dis1.A;
sysb = sys_dis1.B;
sysc = sys_dis1.C;
sysd = sys_dis1.D;

% Phase 2 state space
A2 = [0 0 -1/L; 0 -Ra/La 1/La; 1/C -1/C 0];
B2 = [0 0; 0 -1/La; 0 0];
C2 = eye(3);
D2 = zeros(3,2);

sys2 = ss(A2,B2,C2,D2);
sys_dis2 = c2d(sys2,dt);

sys2a = sys_dis2.A;
sys2b = sys_dis2.B;
sys2c = sys_dis2.C;

```

```

sys2d = sys_dis2.D;

% Motor state space
A_motor = Kt/J;
B_motor = -1/J;
C_motor = Ke;
D_motor = 0;

sysmotor = ss(A_motor,B_motor,C_motor,D_motor);
sys_dismotor = c2d(sysmotor,dt);

motora = sys_dismotor.A;
motorb = sys_dismotor.B;
motorc = sys_dismotor.C;
motord = sys_dismotor.D;

% Controller state space
Ac = 0;
Bc = 1;
Cc = Ki;
Dc = Kp;

sysc = ss(Ac,Bc,Cc,Dc);
sysc_dis = c2d(sysc,dt);

cona=sysc_dis.A;
conb=sysc_dis.B;
conc=sysc_dis.C;
cond=sysc_dis.D;

x1 = 0;
x2 = 0;
g = 100;
v = 0;
j = 1;

% Helping vector
IL = zeros(size(t));
VL = zeros(size(t));
Va = zeros(size(t));
Ia = zeros(size(t));
Vc = zeros(size(t));
Ic = zeros(size(t));
Te = zeros(size(t));
wm = zeros(size(t));
error = zeros(size(t));
x = zeros(size(t));
duty = zeros(size(n));
step_p = zeros(size(g));

for k=0:1:n-1
    if (j==1)
        % Generating pulse
        for i=1:g
            if(pulse(i)>= duty(j))
                step_p(i) = min(pulse);
            else
                step_p(i) = max(pulse);
            end
        end
    else
        for i=1:g
            if(pulse(i)>= duty(j-1))
                step_p(i) = min(pulse);
            else
                step_p(i) = max(pulse);
            end
        end
    end
end

```

```

        end
    end

    % Calculating current and voltages
    for u = 1:Ts/dt
        if(step_p(u) == max(step_p))
            % phase 1
            IL(u+1+v) = sysa(1,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(1,:)*[Vs; Va(u+v)];
            Ia(u+1+v) = sysa(2,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(2,:)*[Vs; Va(u+v)];
            Ic(u+v) = IL(u+v) - Ia(u+v);
            Vc(u+1+v) = sysa(3,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sysb(3,:)*[Vs; Va(u+v)];
            VL(u+v) = Vs - Vc(u+v);

        else
            % phase 2
            IL(u+1+v) = sys2a(1,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(1,:)*[Vs; Va(u+v)];
            Ia(u+1+v) = sys2a(2,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(2,:)*[Vs; Va(u+v)];
            Ic(u+v) = IL(u+v) - Ia(u+v);
            Vc(u+1+v) = sys2a(3,:)*[IL(u+v); Ia(u+v); Vc(u+v)] +
sys2b(3,:)*[Vs; Va(u+v)];
            VL(u+v) = - Vc(u+v);

        end

        %omega computing
        Te(u+v) = Kt*Ia(u+v);
        wm(u+1+v) = motora*Ia(u+v) + motorb*TL;
        Va(u+1+v) = motorc*wm(u+1+v);

        % error computing
        error(u+v) = wm(u+v) - omega;
        x(u+1+v) = cona*x(u+v) + conb*error(u+v);

        if(u == g)
            x1 = x(u+v);
            x2 = error(u+v);
        end

    end

    duty(j) = abs(conc*x1 + cond*x2);

    %incrementing constants
    v = v+g;
    j = j+1;
end

% Plots
figure;
plot(t,IL(1:end-1));
grid on;
xlabel('Sec');
ylabel('A');
title('Inductor current T_L = 20 Nm');

figure;
plot(t,VL);
grid on;
xlabel('Sec');
ylabel('V');
title('Inductor voltage T_L = 20 Nm');

```

```

figure;
plot(t,Ia(1:end-1));
grid on;
xlabel('Time(sec)');
ylabel('I_{Load}(A)');
title('Load current T_L = 20 Nm');

figure;
plot(t,Vc(1:end-1));
grid on;
xlabel('Sec');
ylabel('V');
title('Capacitor voltage T_L = 20 Nm');

figure;
plot(t, Ic);
grid on;
xlabel('Sec');
ylabel('A');
title('Capacitor current T_L = 20 Nm');

figure;
plot(t,wm(1:end-1));
grid on;
xlabel('Sec');
ylabel('rad/s');
title('Angular frequency T_L = 20 Nm');

figure;
plot(t,Te);
grid on;
xlabel('Sec');
ylabel('Nm');
title('Electric torque T_L = 20 Nm');

```