

Modeling, Simulation and Control Issues for a Robot ARM; Education and Research (III).

Farhan A. Salem^{1,2}

¹Mechatronics sec., Dept. of Mechanical Engineering, College of Engineering, Taif University, 888, Taif, Saudi Arabia;

²Alpha Center for Engineering Studies and Technology Researches, Amman, Jordan

E-mail: salem_farh@yahoo.com

Abstract—This paper extends writer's previous work and proposes design, modeling and control issues of a simple robot arm design. Mathematical, Simulink models and MATLAB program are developed to return maximum numerical visual and graphical data to select, design, control and analyze arm system. Testing the proposed models and program for different input values, when different control strategies are applied, show the accuracy and applicability of derived models. The proposed are intended for education and research purposes.

Index Terms— Robot ARM, DC Machine, Modeling/ Simulation

I. Introduction

The accurate control of motion is a fundamental concern in Mechatronics applications, where placing an object in the exact desired location with the exact possible amount of force and torque at the correct exact time is essential for efficient system operation [1]. This paper extends writer's previous works [1,2], the ultimate goal of this work addresses design, Modeling, simulation and control issues for robotic arm, Particularly, to design a robot arm, select and apply different control strategies, so that an applied input voltage with a range of 0 - 12 volts corresponds linearly to an output arm angle of 0 to 180, The designed system should respond to the applied input with an overshoot less than 5%, a settling time less 2 second and a zero steady state error, as well as, to simplify and accelerate the process of robot and control design and analysis, two suggested MATLAB function block models with its function block parameters for robot arm design, control selection and analysis purposes to be proposed.

Having both electrical and mechanical parameters, a single joint robot arm is an application example of a Mechatronics electromechanical system used in industrial automation. Each degree of freedom (DOF) is a joint on Robot arm, where arm can rotate or translate, each DOF requires an actuator, A single joint Robot arm is a system with one DOF that is one actuator, when designing and building a robot arm it is required to have as few degrees of freedom allowed for given

application. As shown in Figure 1 , PMDC motor is, widely, used as an electric actuator to drive a robot arm horizontally. The motor system input signal used to provide the control voltage and current to the PMDC motor is a voltage signal. In part I [1] to simplify and accelerate the process of DC motors sizing, selection and dynamic analysis for different applications, using different approaches, different refined mathematical models in terms of output position, speed, current, acceleration and torque, as well as corresponding Simulink models, supporting MATLAB *m.file* and function block models are introduced for comparison and analysis purposes were introduced. In part II [2] addresses a comprehensive study and analysis of most used control strategies of a given DC motor for comparison and analysis purposes, to select the best and most suitable control strategy for specific motion application, e.g. to control the output angular position, θ and speed of a given DC motor, used in Mechatronics motion application. These works are intended for research purposes, as well as for the application in educational process. In [3] to simplify and accelerate Mechatronics motion control design process including; performance analysis and verification of a given electric DC system, proper controller selection and verification for desired output speed or angle, a new, simple and user-friendly MATLAB built-in function, mathematical and Simulink models are introduced

II. System Modeling, Simulation, Characteristics And Analysis

In modeling process, to simplify the analysis and design processes, linear approximations are used as long as the results yield a good approximation to reality [3]. As shown in Figure 1, Single joint robot arm system consists of three main parts; arm, connected to actuator through gear train with gear ratio, n [4]. Because of the ease with which they can be controlled, systems of DC machines have been frequently used in many applications requiring a wide range of motor speeds and a precise output motor control [5-6]. The robot arm system to be designed, has the following nominal values; arm mass, $M= 8$ Kg, arm length, $L=0.4$ m, and viscous damping constant, $b = 0.09$ N.sec/m. The following nominal values for the various

parameters of electric motor used: $V_{in}=12$ Volts;
 $J_m=0.02$ kg m²; $b_m=0.03$; $K_t=0.023$ N-m/A; $K_b=0.023$

V-s/rad; $R_a=1$ Ohm; $L_a=0.23$ Henry; T_{Load} , gear ratio, for simplicity, $n=1$.

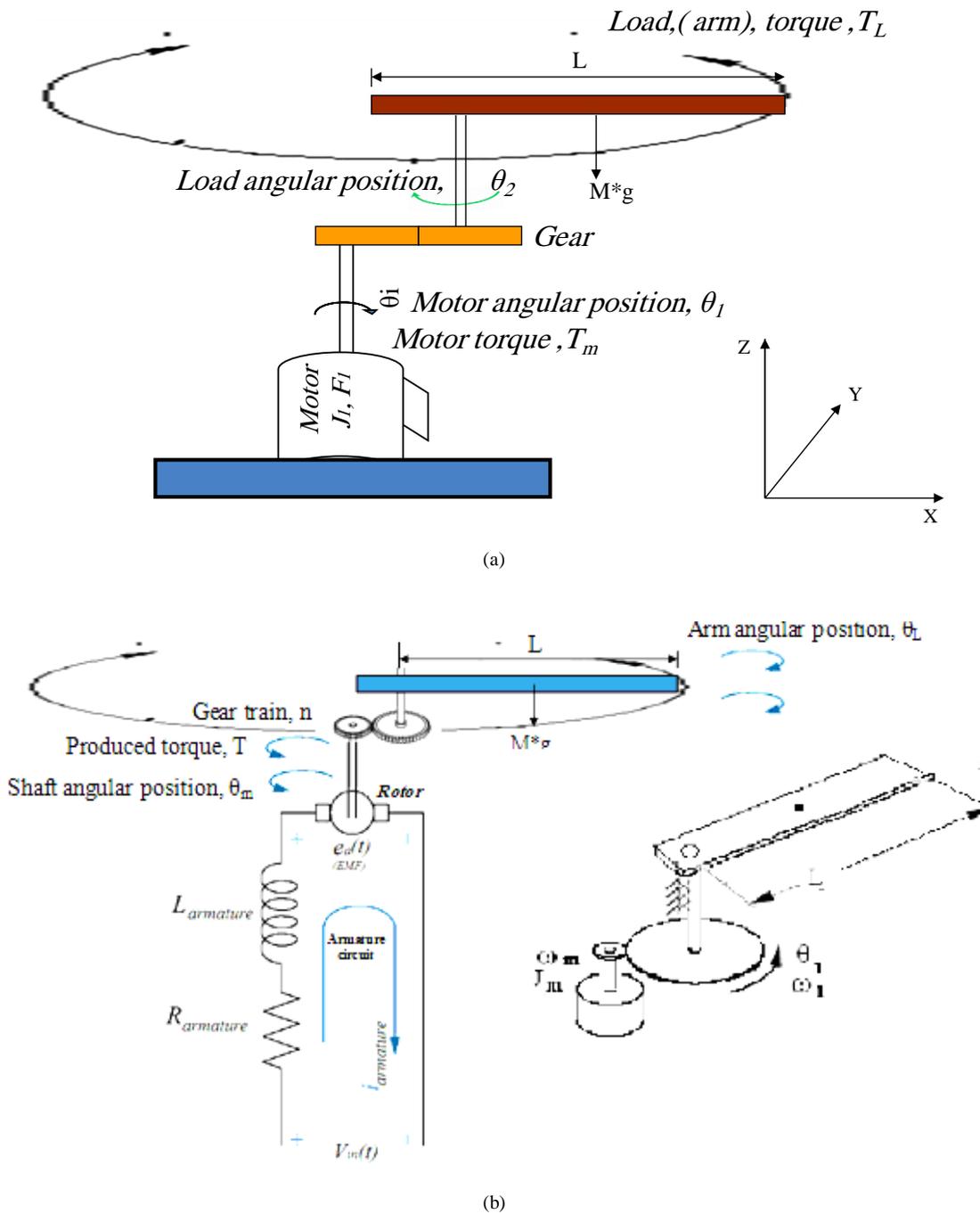


Fig. 1(a)(b): Simplified schematic model of a single joint (one DOF) robot arm and DC motor used to drive arm horizontally

Based on the Newton's law combined with the Kirchoff's law, the mathematical model of PMDC motor in the form of differential equations or state space equations, describing electric and mechanical characteristics of the motor can be derived. In [2-4], based on different approaches, detailed derivation of different refined mathematical models of PMDC motor (e.g. Eqs (1-2)) and corresponding Simulink models are introduced, as well as an function blocks with its function block parameters window for open loop DC

system motor selection, verification and performance analysis. Any of those proposed models (e.g. Figure 2(a)(b)) can be used to analyze the performance of open loop robot arm system and obtaining angular/Linear position/time, angular/Linear speed/time, angular/Linear Acceleration/time, Current/time and Torque/time response curves for step input voltage, defining desired robot arm system parameters and running this model will result in response curves shown in Figure 3.

The PMDC motor open loop transfer function without any load attached relating the input voltage,

$V_{in}(s)$, to the motor shaft output angle, $\theta_m(s)$, and given by:

$$G_{angle}(s) = \frac{\theta(s)}{V_{in}(s)} = \frac{K_t}{\left\{ \left[(L_a J_m s^3 + (R_a J_m + b_m L_a) s^2 + (R_a b_m + K_t K_b) s) \right] \right\}} \quad (1)$$

The PMDC motor open loop transfer function relating the input voltage, $V_{in}(s)$, to the motor shaft output angular velocity, $\omega_m(s)$, given by:

$$G_{speed}(s) = \frac{\omega(s)}{V_{in}(s)} = \frac{K_t}{\left\{ \left[L_a J_m s^2 + (R_a J_m + b_m L_a) s + (R_a b_m + K_t K_b) \right] \right\}} \quad (2)$$

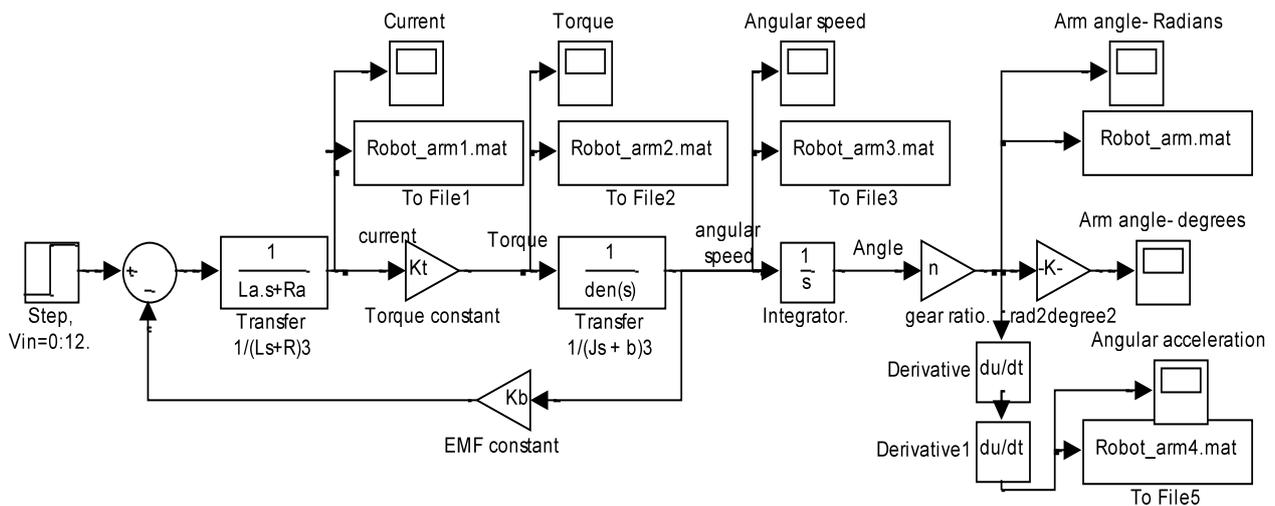


Fig. 2(a) PMDC motor Simulink model based on Eqs. (1) and (2)

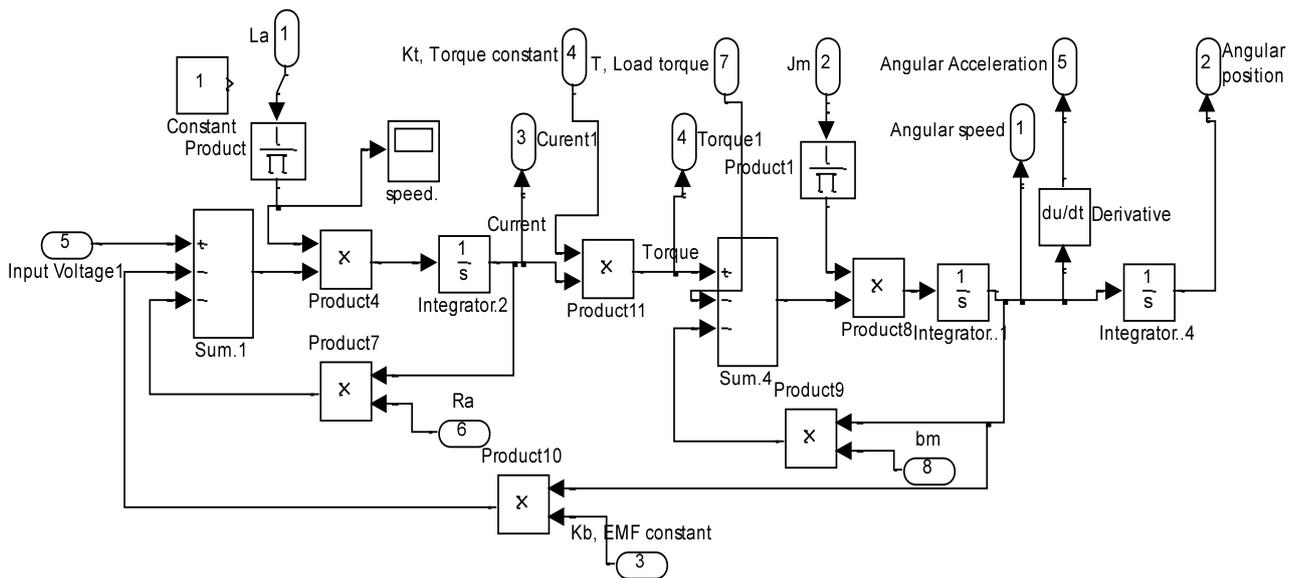


Fig. 2 (b)

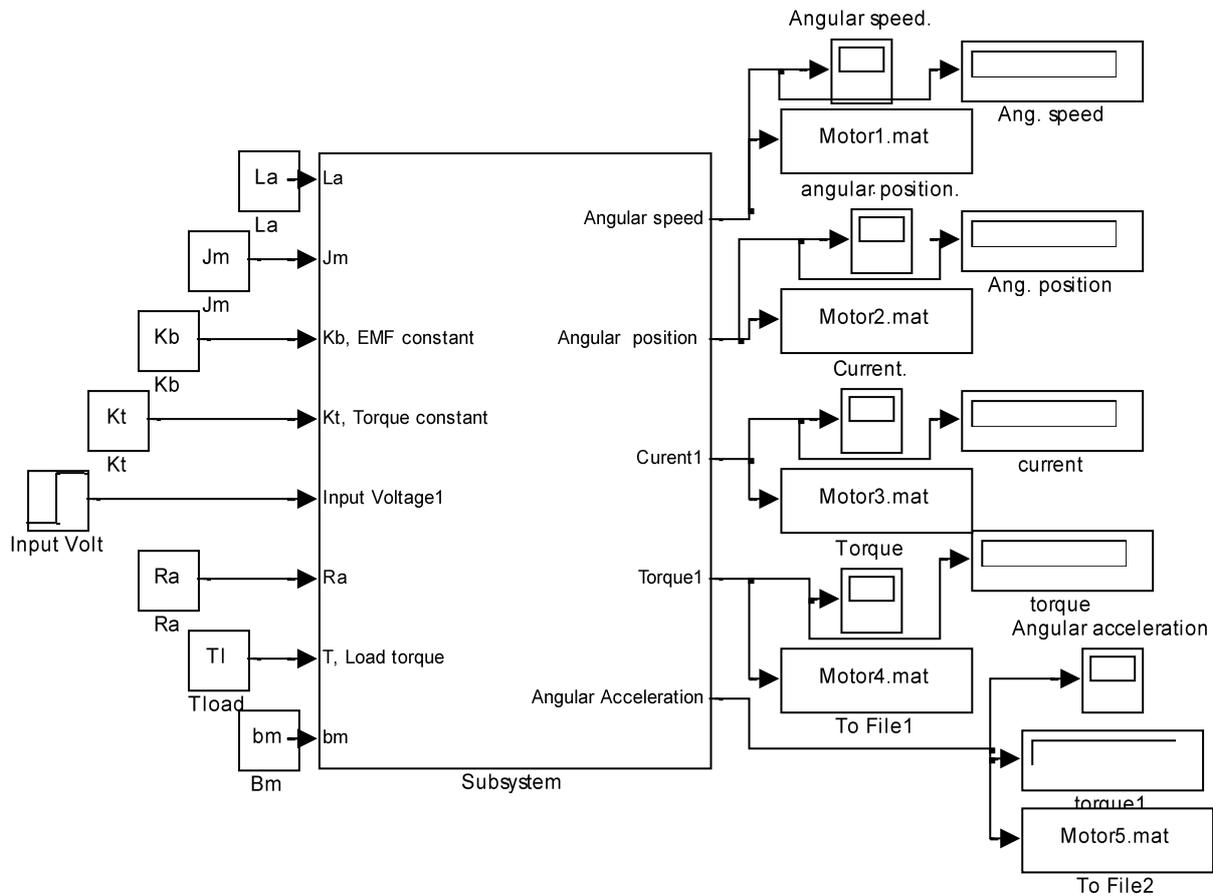


Fig. 2 (c)

Fig. 2(b)(c): Subsystem and function block model with its function block parameters window for DC motor performance verification and analysis [2]

To model, Simulate and analyze the open loop Robot arm system, considering that the end-effector is a part of robot arm, The total equivalent inertia, J_{equiv} , and total

equivalent damping, b_{equiv} at the armature of the motor are given by:

$$b_{equiv} = b_m + b_{Load} \left(\frac{N_1}{N_2} \right)^2 \quad J_{equiv} = J_m + J_{Load} \left(\frac{N_1}{N_2} \right)^2 \quad (3)$$

The geometry of the mechanical part determines the moment of inertia and damping, To compute the total inertia, J_{equiv} , we first consider robot arm as thin rod of mass m , length ℓ , (so that $m = \rho * \ell * s$), this rod is rotating around the axis which passes through its center and is perpendicular to the rod. The moment of inertia of the robot arm can be found by computing the following integral:

$$\int_{-l/2}^{l/2} \rho x^2 s dx = \rho s \frac{x^3}{3} \Big|_{-l/2}^{l/2}$$

$$= \frac{m}{sl} s 2 \frac{l^3}{8} = \frac{1}{12} ml^2$$

Calculating and substituting values in (14) gives:

$$J_{Load} = (8 * (0.4)^2) / 12 = 0.1066666666666667 \text{ kg.m}^2$$

Substituting, we obtain, J_{equiv} , to be :

$$J_{equiv} = J_m + J_{load} * (1/1)$$

$$J_{equiv} = 0.02 + 0.107 = 0.1267 \text{ kg m}^2$$

Obtaining the total damping, b_{total} , gives:

$$b_{equiv} = b_m + b_{load} (1/1)$$

$$b_{equiv} = 0.03 + 0.09 = 0.12 \text{ N.sec/m}$$

In order to obtain total system transfer function, relating input voltage V_{in} and Arm-Load output angular position θ_{Load} , We Substitute values into transfer function given by (12) with gear ratio, n , gives:

$$G_{am_angle_open}(s) = \frac{\theta_{Load}(s)}{V_{in}(s)} = \frac{K_t * n}{\left\{ \left[(L_a J_{equiv} s^3 + (R_a J_{equiv} + b_{equiv} L_a) s^2 + (R_a b_{equiv} + K_t K_b) s \right] \right\}}$$

$$G_{am_angle_open}(s) = \frac{\theta_{Load}(s)}{V_{in}(s)} = \frac{0.023}{0.02913s^3 + 0.1543s^2 + 0.1205s}$$

$$G_{am_angle_open}(s) = \frac{\theta_{Load}(s)}{V_{in}(s)} = \frac{0.023}{s(s+4.342)(s+0.9527)} \tag{4}$$

Defining the given DC motor's nominal values , gear ratio and arm parameters, and running any of both models ,We can analyze the performance of open loop Robot arm systemand obtaining the response curves for step input voltage (0-12 volts), in terms of output Position/time, speed/time, Acceleration / time, Current/time and Torque/time, shown in Figure 3.

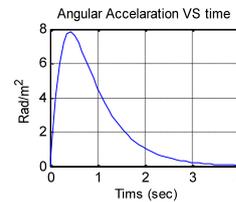
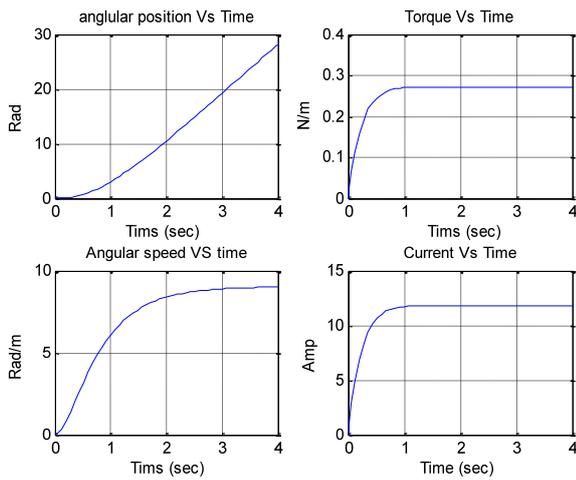


Fig. 3: Open loop DC motor system; Angular Position/time, Angular speed/time, Angular Acceleration / time, Current/time and Torque/time response curves for 12 V input



III. Control System Selection, Design and Analysis

A negative closed loop feedback control system with forward controller and corresponding Simulink model shown in Figure .4(a)(b) are to be used [3]. Our design goal is to design, model, simulate and analyze a control system so that a voltage input in the range of 0 to 12 volts corresponds linearly of an Robot arm output angle range of 0 to 180, that is to move the robot arm to the desired output angular position, θ_L , corresponding to the applied input voltage, V_{in} , with overshoot less than 5%, a settling time less than 2 second and zero steady state error. The error signal, e is the difference between the actual output robot arm position, θ_L , and desired output robot arm position.

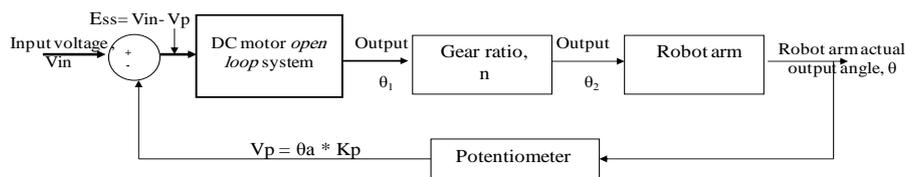


Fig. 4(a)

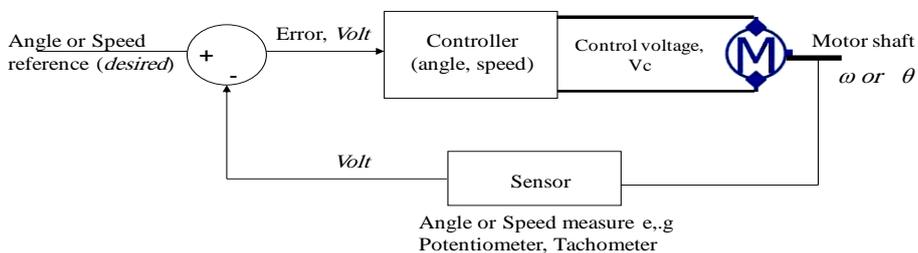


Fig. 4(b)

Fig. 4(a)(b): Two Block diagram representations of PMDC motor control [3]

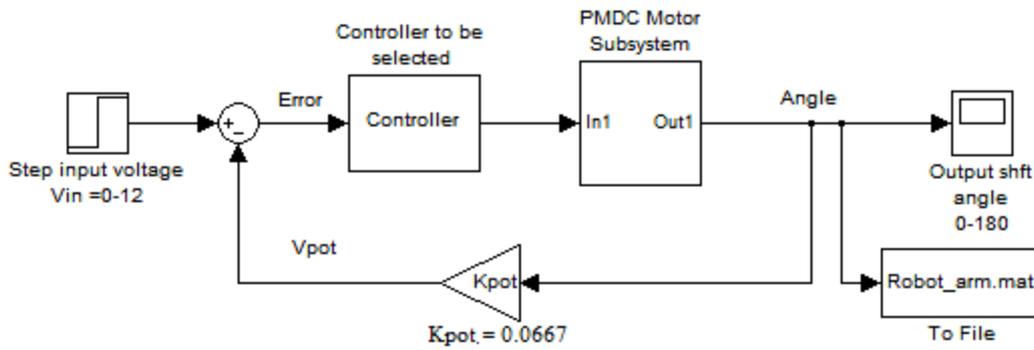


Fig. 4 (c): Preliminary Simulink models for negative feedback with forward compensation [3]

3.1 Sensor Modeling

To calculate the error, we need to convert the actual output robot arm position, into voltage, V , then compare this voltage with the input voltage V_{in} , the difference is the error signal in volts. Potentiometer is a popular sensor used to measure the actual output robot (arm) position, θ_L , convert into corresponding volt, V_p and then feeding back this value, the Potentiometer output is proportional to the actual robot arm position, θ_L , this can be accomplished as follows: The output voltage of potentiometer is given by (5), in this equation: θ_L : The actual robot arm position. K_{pot} the potentiometer constant; It is equal to the ratio of the voltage change to the corresponding angle change, and given by Eq(5). Depending on maximum desired output arm angle, the potentiometer can be chosen, for our case $V_{in}= 0:12$, and output angle = 0:180 degrees, substituting, we have:

$$\begin{aligned}
 V_p &= \theta_L * K_{pot} \\
 K_{pot} &= \frac{\text{(Voltage change)}}{\text{(Degree change)}} \\
 &= \frac{(12 - 0)}{(180 - 0)} \\
 &= 0.0667 \text{ V / degree}
 \end{aligned} \tag{5}$$

The value (0.0667), means that each one input volt corresponds to $180/12= 15$ output angle in radians, to obtain a desired output angular position of 180, we need to apply 12 volts, to obtain an angular position of 90 we need to apply ($90*0.0667= 6.0030$ Volts).

3.2 Control Strategy Selection and Design

Running the Simulink model of the robot arm closed loop system without controller attached ($K=1$) shown in Figure 5(a), as well as the below MATLAB mcode, will return step response shown in Figure 5(b), the response curve shows that the arm system responds with a very huge value of settling time, (approximately $T_s=500$ seconds) to reach steady state output angle of 180 degrees, that is not corresponding to the required

design specifications. A controller is to be designed to meet desired specifications. In [3] different control strategies were applied to PMDC and it is found that, there are many motor control strategies that may be more or less appropriate to a specific type of application each has its advantages and disadvantages. The designer must select the best one for specific application, where when applying lag-controller to PMDC system, the system stability is increased, the settling time, T_s is reduced and steady state error is reduced, so called PD controller with deadbeat response, enables designer to satisfy most required design specifications smoothly and within a desired period of time and can be applied to meet the desired specification., applying internal model controller result in improving system's both dynamic and steady state performances, reducing overshoot, rise time and settling time as well as disturbance rejection, when applying PID controller to PMDC system, the system stability is increased, the settling time T_s is decreased and steady state error is almost eliminated, and so on. By proper selection of the three PID gains, different characteristics of the motor response are controlled. To achieve a fast response to a step command with minimal overshoot and zero steady state error. We will, separately, apply these control strategies to our Robot arm system, and later to simplify and accelerate the whole process of robot arm system design in terms of designing the mechanical arm and gear parts as well as selection and designing the most suitable control strategy, a general model in the form of function block with its function block parameters will be built.

```

>> Vin=12; Jm=0.02 ; Jm=0.1267;bm=0.12 ;Kt
=0.023 ; Kb =0.023 ; Ra =1 ; La=0.23 ; TL = 0 ;Tl =
TL ;n=1; M=8;desired_max_angle=180; Arm_Length
=0.4 ;b_arm= 0.09;
Kpot = Vin/desired_max_angle; ;
J_equiv= Jm+((M* Arm_Length ^2)/12)*(1/n)^2;
b_equiv= bm + b_arm*(1/n)^2;
num_arm_open = [Kt*n];%[Kt*n*(180/pi)]
den_arm_open=[La*J_equiv (Ra*J_equiv + La*
b_equiv) (Ra*b_equiv + Kt*Kb) 0];
disp(' Robot arm open loop transfer function , (Arm
angle)/Volt')

```

$G_{a_arm_open}=tf(num_arm_open,den_arm_open)$ $step(12*G_{a_arm_closed})$
 $G_{a_arm_closed}=feedback(G_{a_arm_open},Kpot/n),$

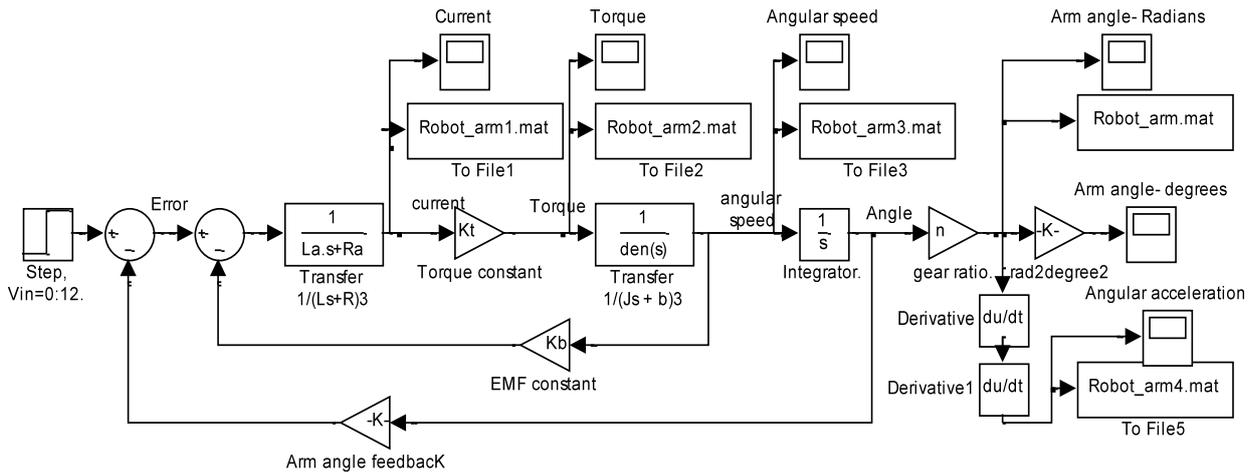


Fig. 5(a): Simulink model of closed loop robot arm, with K=1

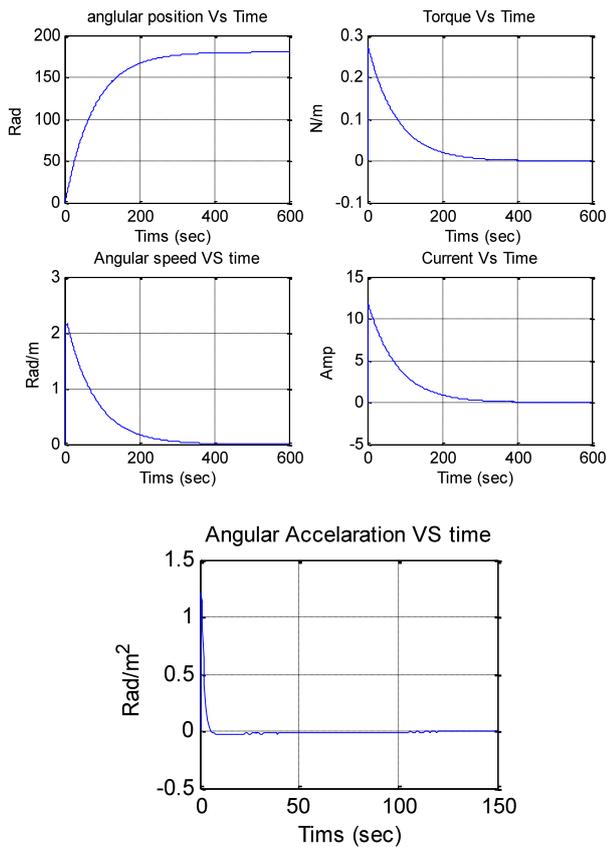


Fig. 5(b): Closed loop robot arm system step response Angular Position/time, Angular speed/time, Angular Acceleration / time, Current/time and Torque/time curves for 12 V input

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s$$

$$= \frac{K_D s^2 + K_p s + K_I}{s} \tag{6}$$

$$= \frac{K_D \left[s^2 + \frac{K_p}{K_D} s + \frac{K_I}{K_D} \right]}{s}$$

The Simulink model of closed loop robot arm, with DC motor, as subsystem and PID controller applied, is shown in Figure 6 (a). The response of this model is shown in Figure .6 (b). With optimal set of PID controller gains; K_p , K_I , K_D , response curves still show the existence of steady state error ($e_{ss}=0.0062$) and overshoot of 0.024%. It is noted that the current can be controlled by adding and adjusting saturation block. The following MATLAB code can also be used to obtain step response

```
>> Kp= 506.712827552343; Ki =
46.1017257705634; Kd= 562.448810124181
num_PID= [ Kd Kp Ki ];
den_PID= [ 1 0];
G_PID= tf(num_PID, den_PID)
G_forward= series(G_PID, Ga_arm_open);
G_close_arm_PID = feedback(G_forward, Kpot/n)
step(V*G_close_arm_PID)
axis([0 8 0 200])
```

3.2.1 Controller Design Using PID Strategy:

PID controllers are commonly used to regulate the time-domain behavior of many different types of dynamic plants [6]. The transfer function of PID control is given by:

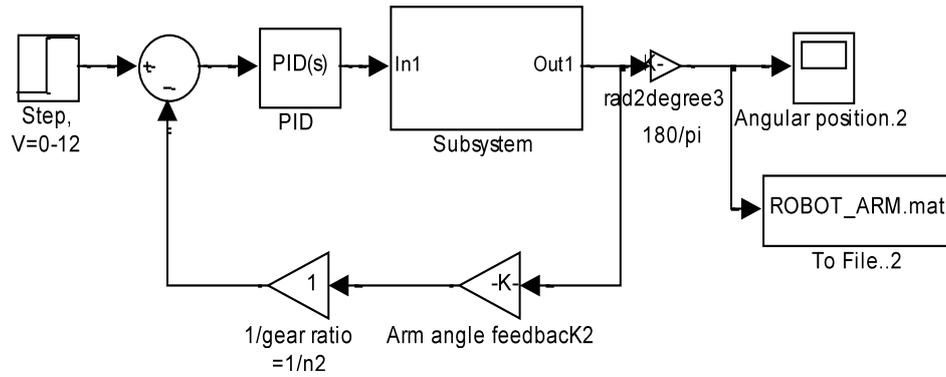


Fig. 6 (a) Simulink model with PMDC motor subsystem and PID controller

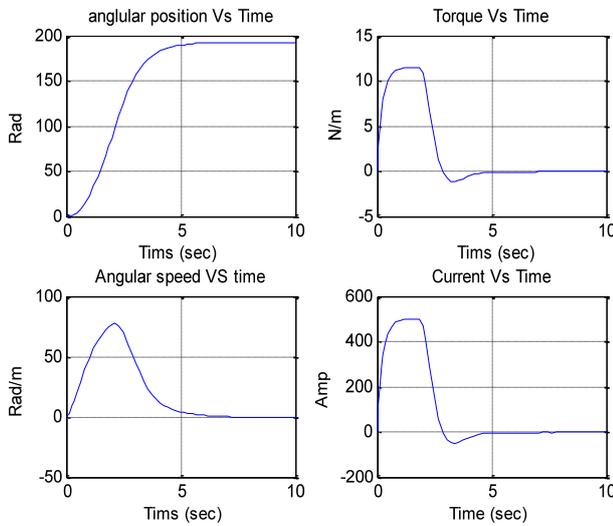


Fig. 6(a): Step response of closed loop robot arm, with DC motor, as subsystem and PID controller

3.3 Controller Design Using Lead and Lag Compensators

Lead compensator is a soft approximation of PD-controller, The PD controller, given by $G_{PD}(s) = K_P + K_D s$, is not physically implementable, since it is not proper, and it would differentiate high frequency noise, thereby producing large swings in output. to avoid this, PD-controller is approximated to lead controller of the following form [13]:

$$G_{PD}(s) \approx G_{Lead}(s) = K_P + K_D \frac{Ps}{s + P}$$

The larger the value of P , the better the lead controller approximates PD control, rearranging gives:

$$\begin{aligned} G_{Lead}(s) &= K_P + K_D \frac{Ps}{s + P} \\ &= \frac{K_P(s + P) + K_D Ps}{s + P} \end{aligned}$$

$$\begin{aligned} G_{Lead}(s) &= \frac{(K_P + K_D P)s + K_P P}{s + P} \\ &= (K_P + K_D P) \frac{s + \left[\frac{K_P P}{K_P + K_D P} \right]}{s + P} \end{aligned}$$

$$\text{Now, let } K_C = K_P + K_D P \text{ and } Z = \left[\frac{K_P P}{K_P + K_D P} \right],$$

we obtain the following approximated controller transfer function of PD controller, and called lead compensator:

$$G(s) = K_C \frac{s + Z}{s + P}$$

If $Z < P$ this controller is called a lead controller (or lead compensator). If $Z > P$: this controller is called a lag controller (or lag compensator). The transfer function of lead compensator is given by:

$$G_{lead}(s) = K_c \frac{(s + Z_o)}{(s + P_o)} \quad (7)$$

$$\text{Where: } P > Z,$$

The Lag compensator is a soft approximation of PI controller, it is used to improve the steady state response, particularly, to reduce steady state error of the system, the reduction in the steady state error accomplished by adding equal numbers of poles and zeroes to a systems.

$$\begin{aligned} G_{PI}(s) \approx G_{lag}(s) &= K_P + \frac{K_I}{s} \\ &= \frac{K_P s + K_I}{s} \\ &= K_P \frac{\left(s + \frac{K_I}{K_P} \right)}{s} \end{aligned} \quad (8)$$

Since PI controller by it self is unstable, we approximate the PI controller by introducing value of P_o that is not zero but near zero; the smaller we make P_o , the better this controller approximates the PI controller, and the approximation of PI controller will have the form:

$$G_{lag}(s) = K_c \frac{(s + Z_o)}{(s + P_o)} \tag{9}$$

Where: $Z_o > P_o$, and Z_o small numbers near zero and $Z_o = K_I/K_P$, the lag compensator zero. P_o : small number, The smaller we make P_o , the better this controller approximates the PI controller. The Simulink model of both Lead and Lag compensators is shown in Figure .7 (a). The step responses are shown in Figure .7(b).

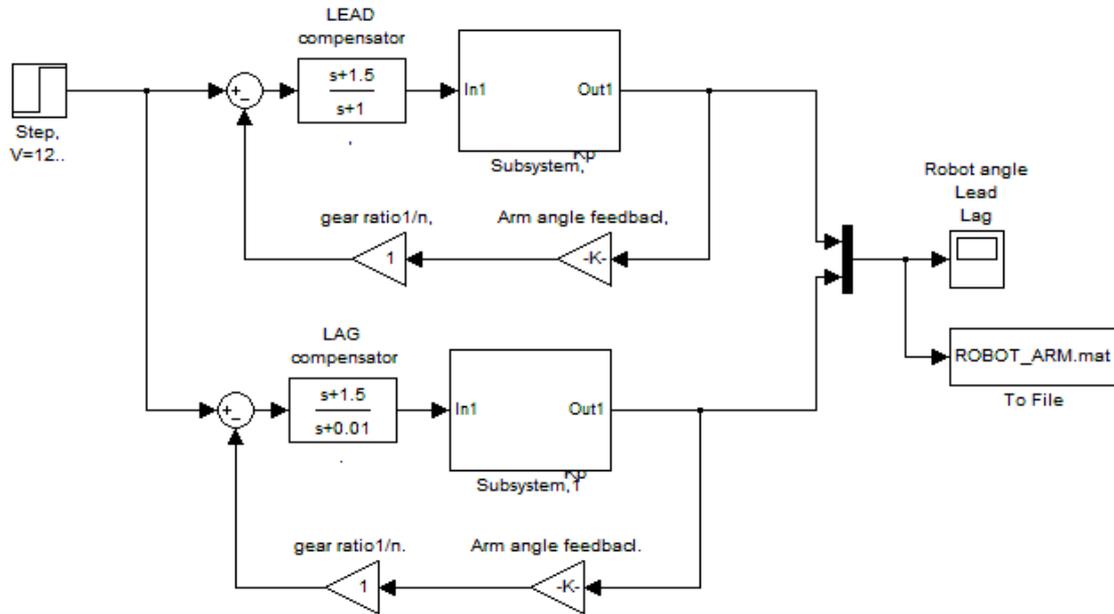


Fig. 7 (a): Simulink model of both Lead and Lag compensators

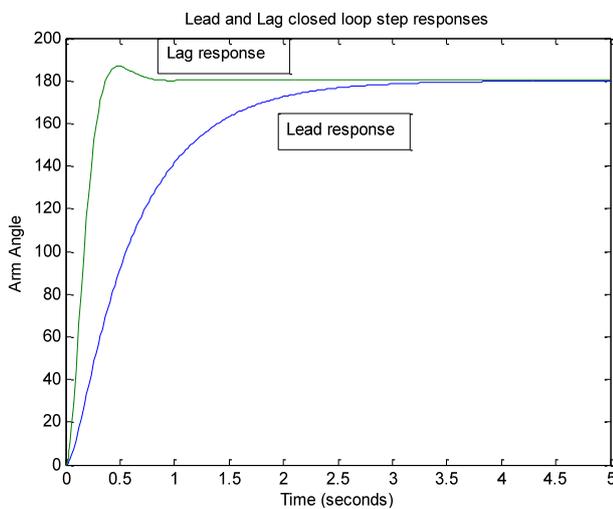


Fig. 7 (b): Position step response applying separately lead and lag compensators, step 12 volts, output 180 degree

3.4 Function Block for System Design and Analysis

To simplify and accelerate the whole process of Mechatronics robot arm system design in terms of designing mechanical parts; arm and gear , as well as selection and designing the most suitable control

strategy, Two models in the form of function block with its function block parameters will be built, first simple using only PID controller and a second general model with most control strategies applied to Robot arm design and control.

3.4.1 PID Controller Function Block for Robot ARM Design and Analysis

A suggested Function Block model for robot arm design, controller selection, design and robot system analysis is shown in Figure .8 (b) , the function block subsystem is shown in Figure .8 (a). this model allows designer to apply Simulink PID controller block and its subsystem (P, PI, PD and PID) tuning capabilities and feature to select controller and analyze resulting robot arm response, by defining parameters of Robot arm subsystems (Arm, gears, and DC motor) selecting controller (PID), and running model, will result in speed/time and position/time curves shown in Figure .8(c) .

By changing the potentiometer constant, K_{pot} , the output angle range will change, Running this function block for the given robot arm, for an applied voltage of 0 to 12 volts and with $K_{pot} = 0.1333$ will correspond

linearly to an output arm angle of 0 to 90 , speed/time and position/time curves are shown in Figure .9(a), where Running this function block, for an applied voltage of 0 to 12 volts and with $K_{pot} = 0.2667$ will

correspond linearly to an output arm angle of 0 to 45 degrees, a 45 degrees position step response is shown in Figure .9(b)

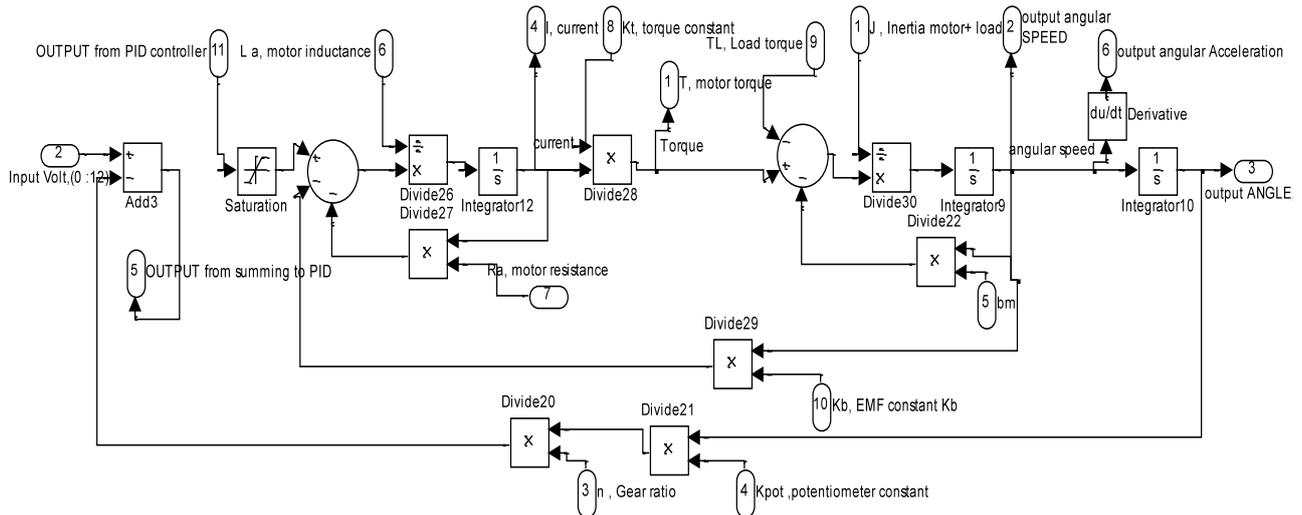


Fig. 8 (a) Robot arm function block subsystem, using only PID Controller

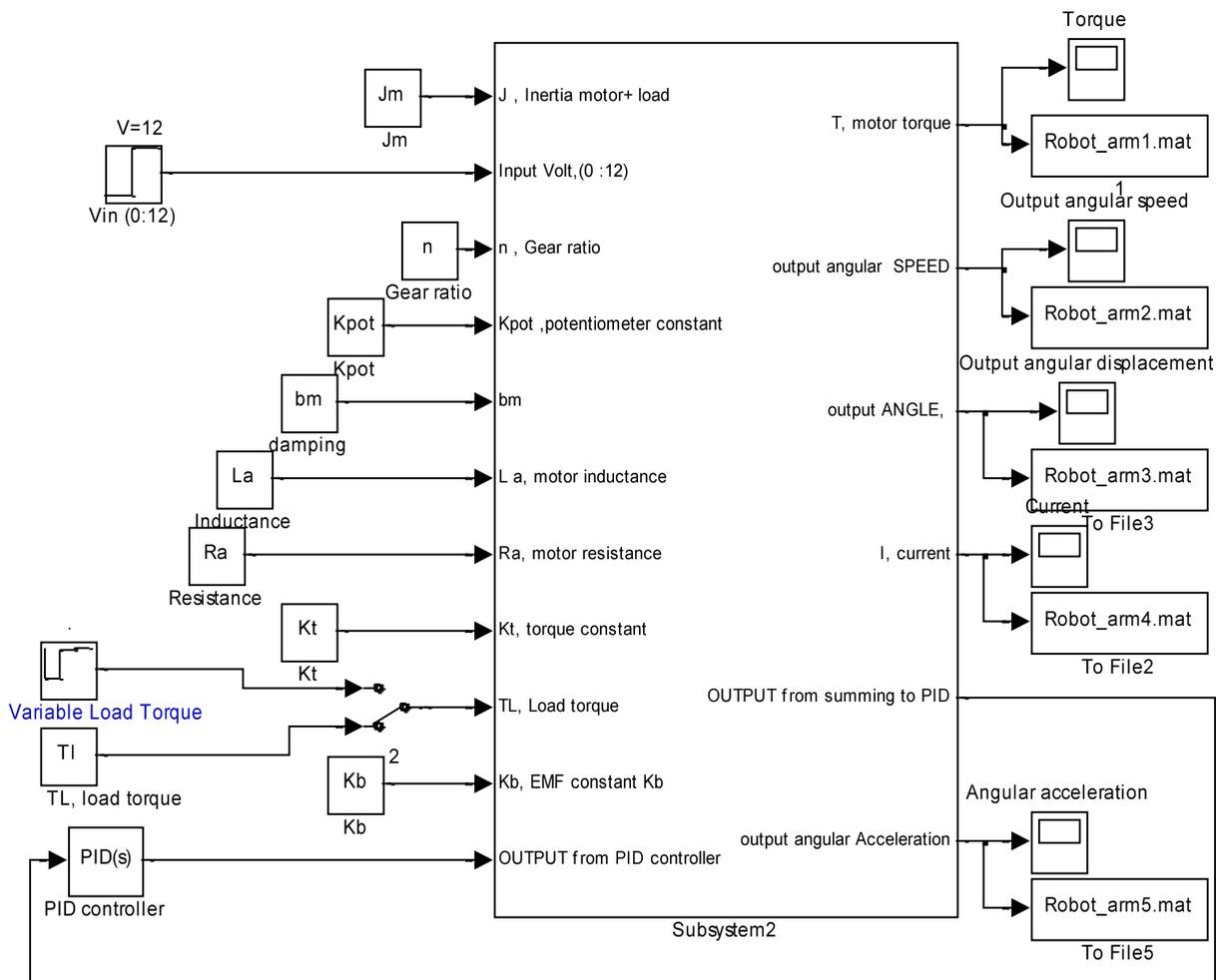


Fig. 8 (b) Function Block using PID for robot arm analysis and design

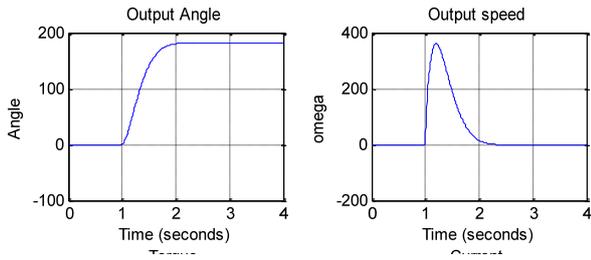


Fig. 8 (c) Speed/time and position/time response of function block using PID controller, $K_{p\theta} = 0.066$, and Arm angular position $\theta = 180$

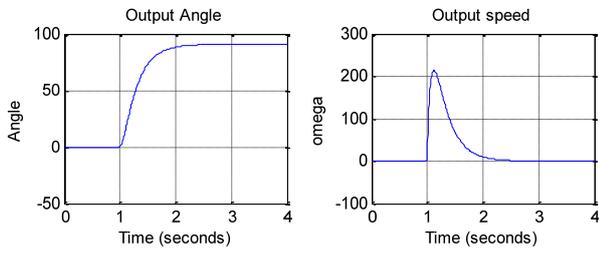


Fig. 9(a) Speed/time and position/time response of function block using PID controller, $K_{p\theta} 0.1333$, output angular position of $\theta = 180$.

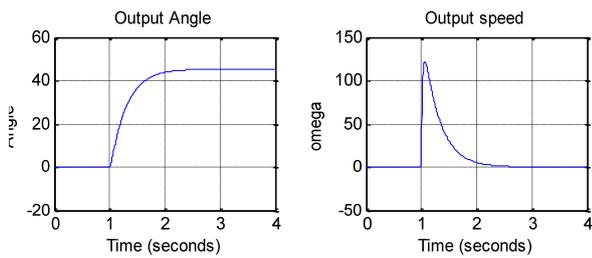


Fig. 9(b) Output Speed/time and position/time, ($\theta = 45$), $K_{p\theta} = 0.2667$

3.4.2 A Proposed Function Block with Its Function Window Parameters, For Robot Arm Design, Controller Selection, Design and Verification

The suggested general function block is shown in Figure 10(b) based and subsystem in Figure 10(a), this model includes most control strategies applied to Robot arm design and control selection ,this model allow designer to define robot arm subsystems parameters, e.g. arm length, weight, gear ratio, damping factor, as well as to define electric motor parameter, e.g. $V_{in}=12;J_m=0.02$. after defining all parameters, model allows using manual switch, select any control strategy , finally run the model and analyze response curves of a given robot arm system. the control strategies included in this model are, P, PI, PD, PID included in PID Simulink block, lead, lag compensator, PD with deadbeat, lead-integral controllers, model will return all the following response curves for any input from zero to 12 volts, Angular Position/time, Angular speed/time, Angular Acceleration / time, Current/time and Torque/time curves for 12 V input, this model, also, allows designer to use Simulink tuning capabilities of PID controller and its features. Switching model to PID controller, running simulation, will result in Arm position/time response curve shown in Figure .10 (c). This model can be modified to include other control strategies.

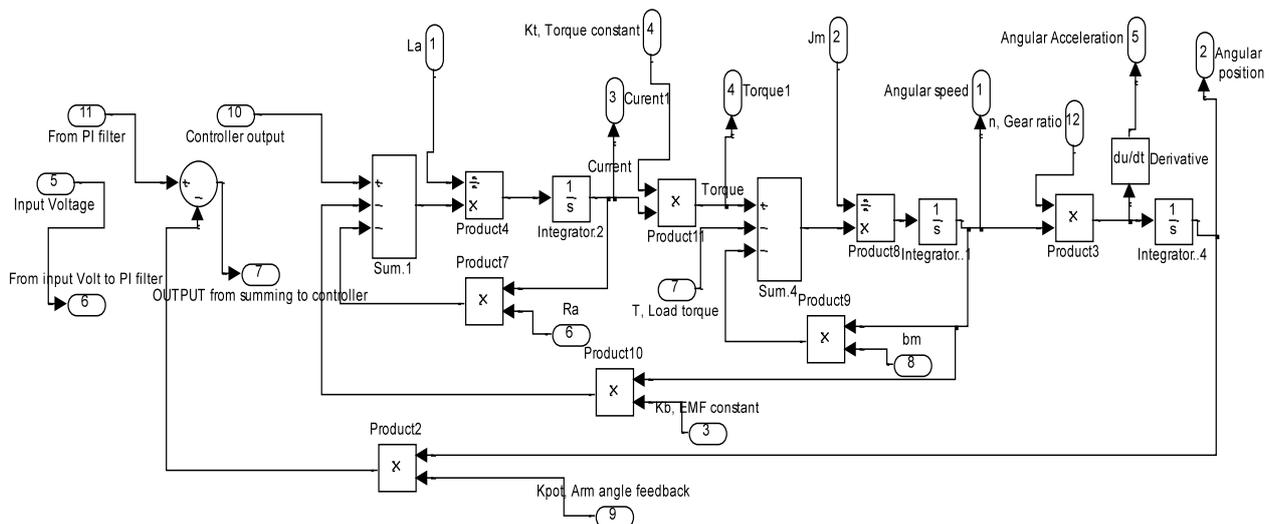


Fig. 10 (a) Robot arm function block subsystem

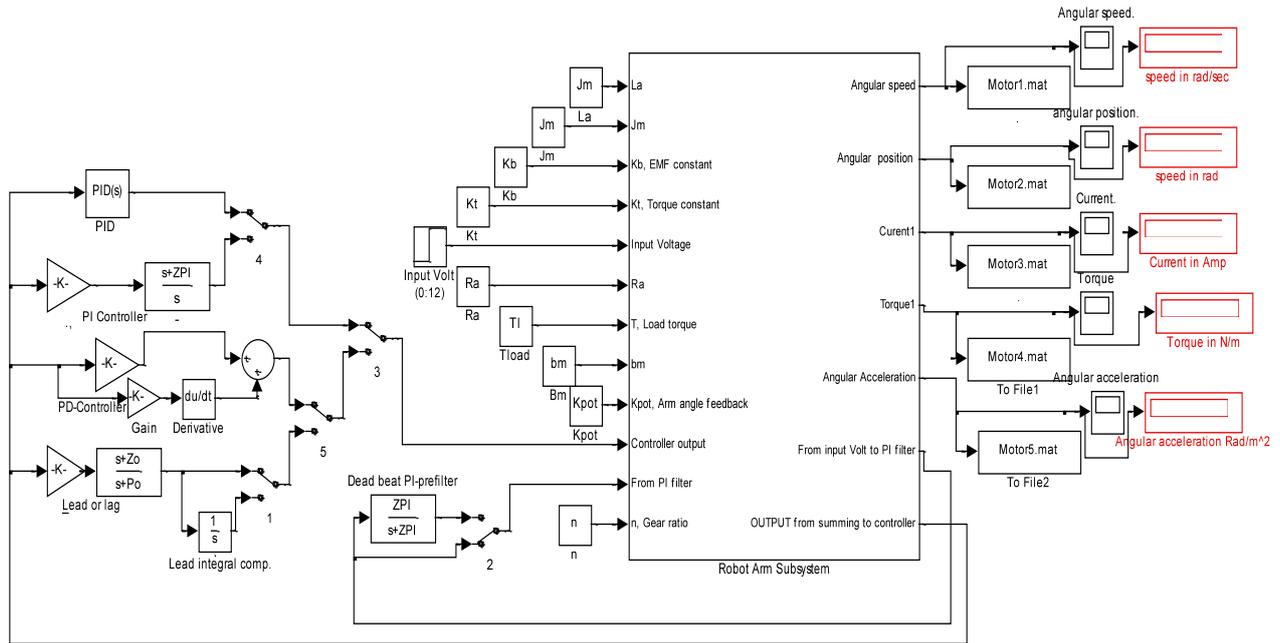


Fig. 10 (b) Function block with its function block parameters window for robot arm design, controller selection, design and verification

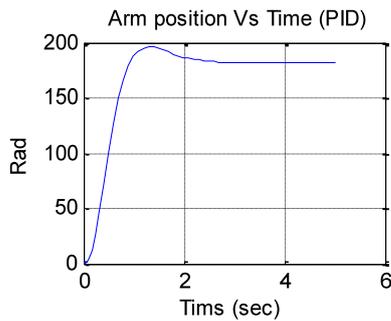


Fig. 10 (c) Arm position/time response of function block

3.5 A Proposed m.file for Robot ARM Research and Education

The following m.file can be used to return transfer functions, open loop and closed loop, with and without applying controller as well as corresponding responses and performance specifications.

```

clc, clear all, close all
Jm = input(' Enter Motor armature moment of inertia (Jm) :');
bm = input(' Enter damping constant of the the motor system (bm):');
Kb = input(' Enter ElectroMotive Force constant (Kb):');
Kt = input(' Enter Torque constant (Kt):');
Ra = input(' Enter electric resistance of the motor armature (ohms), (Ra):');
La = input(' Enter electric inductance of the motor armature ,Henry,(La) :');
V = input(' Enter applied input voltage Vin :');
    
```

```

V_max = input(' Enter maximum allowed voltage, V max :');
Angle_max = input(' Enter maximum allowed angle for robot arm : ');
    
```

```

M = input(' Enter robot arm mass , M=');
Length = input(' Enter robot arm Length, L=');
b_arm = input(' Enter damping factor of robot arm, b_arm =');
n = input(' Enter gear ratio n = ');
%obtaining open loop transferfunctions of DC motor system and step response
    
```

```

num1 = [1];
den1 = [La ,Ra];
num2 = [1];
den2 = [Jm ,bm];
A = conv( [La ,Ra], [Jm ,bm]);
TF1 = tf( Kt*n, A);
disp('DC motor open loop transfer function, Speed/Volt: ')
Gv = feedback(TF1,Kt)
disp(' DC motor open loop transfer function, Angle/Volt: ')
Ga = tf(1,[1,0] ) *Gv
subplot(4,2,1),
step (V*Ga);
title(' Position step response of open loop DC motor system')
subplot(4,2,2)
step (V*Gv)
title(' velocity step response of open loop DC motor system')
%{
or based on open loop transfer function ,the following code can be used
num = [Kt*n];
    
```

```

den=[La*Ja (Ra*Ja+Lm*bm) (Ra*bm+Kt*Kb) 0]
disp(' Robot arm open loop transfer function,
Angle/Volt: ')
Ga =tf(num,den)
step (V*Ga)
%}
% obtaining open loop transfer functions of robot
arm system and step response
J=((M*Length^2)/12)+Jm;
b= bm + b_arm;
num_arm_open = [Kt*n*(180/pi)];
den_arm_open=[La*J (Ra*J+La*b)
(Ra*bm+Kt*Kb) 0];
disp(' Robot arm open loop transfer function , (Arm
angle)/Volt')
Ga_arm_open=tf(num_arm_open,den_arm_open)
subplot(4,2,3)
step(V*Ga_arm_open);
title(' Position step response ,open loop arm system')
subplot(4,2,4)
comet(step(V,Ga_arm_open))
title(' Position step response')
% Obtaining closed loop transfer functions of robot
arm system and step response
Kpot=V_max/Angle_max; % potentiometer
G_close_arm= feedback(Ga_arm_open,(Kpot/n))
subplot(4,2,5),
step(V*G_close_arm);
title(' closed loop position step response')
subplot(4,2,6)
comet(step(V*G_close_arm))
title(' Robot arm Position step response')
xlabel('')
% Controller Design : PID Controller
Kpro=input(' Enter Proportional gain Kp :');
Ki=input(' Enter Integral gain Ki :');
Kd=input(' Enter derivative gain Kd :');
num_PID=[Kd Kpro Ki] ;
den_PID=[ 0 1 0];
disp(' PID transfer function')
G_PID=tf(num_PID,den_PID);
G_forward= series(G_PID, Ga_arm_open);
disp(' TF of Robot arm with PID controller, (Arm
angle)/Volt')
G_close_arm_PID = feedback(G_forward, Kpot/n)
% step(V*G_close_arm_PID)
subplot(4,2,7)
step(G_close_arm_PID)

```

IV. Conclusion

To simplify and accelerate the process of robot arm system design in terms of motor subsystem sizing, designing the mechanical subsystem, as well as selection and design of the most appropriate control strategy to achieve desired performance, two Simulink models first for open loop DC system motor selection and performance analysis and second for overall arm system performance analysis are proposed and tested.

Introduced models can be used to select and analyze the performance of open and closed loop systems and obtaining the Angular Position/time, Angular speed/time, Angular Acceleration / time, Current/time and Torque/time response curves for given input voltage; finally for research and education purposes, a MATLAB m.file is written, to return transfer functions, open loop and closed loop, with and without applying controller, as well as corresponding responses and performance specifications. Analyzing obtained response curves for different input voltage values, when different control strategies is applied, show the accuracy and applicability of derived models for research purposes in selection, performance analysis and control of mechatronics system requiring output position control.

References

- [1] Chun Htoo Aung, Khin Thandar Lwin, and Yin Mon Myint, Modeling Motion Control System for Motorized Robot arm using MATLAB, World Academy of Science, Engineering and Technology 42 2008.
- [2] Ahmad A. Mahfouz, Mohammed M. K., Farhan A. Salem, Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink (I). IJISA Vol. 5, No. 5, 39-57 April 2013 .
- [3] Norman S. Nise, Control system engineering, sixth edition, John Wiley & Sons, Inc, 2011.
- [4] Farhan A. Salem, Modeling controller selection and design of electric DC motor for Mechatronics applications, using different control strategies and verification using MATLAB/Simulink, Submitted to European Scientific Journal, 2013
- [5] M.P.Kazmierkowski, H.Tunia "Automatic Control of Converter-Fed Drives", Warszawa 1994.
- [6] R.D. Doncker, D.W.J. Pule, and A. Veltman. Advanced Electrical Drives: Analysis, Modeling, Control. Springer, 2011.
- [7] Grzegorz SIEKLUCKI, Analysis of the Transfer-Function Models of Electric Drives with Controlled Voltage Source PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review), ISSN 0033-2097, R.88NR7a/2012.
- [8] An educational MATLAB m.file applied to PMDC motor controllers design comparison, selection and analysis Ahmad A. Mahfouz' Farhan A. Salem
- [9] B. Shah, Field Oriented Control of Step Motors, MSc. Thesis, SVMITB haruch, India, Dec. 2004.

- [10] Jamal A. Mohammed, Modeling, Analysis and Speed Control Design Methods of a DC Motor Eng. & Tech. Journal, Vol. 29, No. 1, 2011
- [11] R.C. Dorf and R.H. Bishop, Modern Control Systems, 10th Edition, Prentice Hall, 2008

How to cite this paper: Farhan A. Salem, "Modeling, Simulation and Control Issues for a Robot ARM; Education and Research (III)", International Journal of Intelligent Systems and Applications (IJISA), vol.6, no.4, pp.26-39, 2014. DOI: 10.5815/ijisa.2014.04.03

Appendix:

Table 1 Nomenclature

Symbol	Quantity	Unit
V , or V_m	The applied input voltage	Volte, V
R_a	Armature resistance, (terminal resistance)	Ohm, Ω
R_f	Stator resistance	Ohm, Ω
i_a	Armature current	Ampere, A
K_t	Motor torque constant	N.m/A
K_e	Motor back-electromotive force const.	V/(rad/s)
ω_m	Motor <i>shaft</i> angular velocity	rad/s
T_m	Torque produced by the motor	N.m
J_m	Motor armature moment of inertia	kg.m ²
J_{total}	Total inertia= J_m+J_{load}	kg.m ²
L_a	Armature inductance	Henry, H
b_m	Viscous damping, friction coefficient	N.m/rad.s
e_a , EMF:	The back electromotive force, $EMF = K_b d\theta/dt$	e_a , EMF:
θ_m	Motor shaft output angular position	radians
ω_m	Motor shaft output angular speed	rad/sec
$V_R = R*i$	The voltage across the resistor	Voltage
$V_L = L di/dt$	The voltage across the inductor	Voltage
T_{load}	Torque of the mechanical load	T_{load}
T_α	Torque du to rotational acceleration	
T_ω	Torque du to rotational v velocity	
T_{EMF}	The electromagnetic torque.	

Author's Profile

Farhan A. Salem: B.Sc., M.Sc and Ph.D., in Mechatronics of production systems, 2013. Now he is with Taif University, Mechatronics program, Dept. of Mechanical Engineering and gen. director of alpha center for engineering studies and technology researches.