ROBOT DYNAMICS

SEEKERS

Systems & Biomedical Engineering Department
Cairo University

Faculty of Engineering
Cairo University

<u>Project # 22</u>

Robotic Dynamics

<u>Seekers Team</u>:

Marina Maher
Esraa Khaled
Alzahraa Eid
Irini Adel

<u>Under supervision of :</u>

Dr Ahmed Gomaa

Table of Contents

Table of figures:

1)Abstract:
In the past, human beings were forced to do jobs that subjected them to great danger, and sometimes lead to death. But now everything has changed, life has become better and greatly facilitated due to modern technology and manufacture and development of robots. We cannot deny that none of these would have happened without mathematics.

In this research several aspects of the robotic manipulator will covered including the analysis of their motion using different laws, how they are controlled to move to a desired position, finally some of the very useful applications in several fields will be presented.

2)Introduction:
Due to hard life, we are in urgent need of robotics. They are the bridge between untouchable technology and the physical world. The main components of a robot include sensors, manipulators (moving parts), power supplies that cause motion, and software necessary for assigning tasks. All of these contribute to its great capability for knowing information that exceeds the abilities of the common five senses like: detecting small amounts of non-seen radiation .They are essential in many fields including: designing of biomedical equipment, helping the physically challenged people, robotic assistant for microsurgery, doing dangerous work, education, hospital, industry and even going far down into the unknown waters and mines where human would be crushed.

There are many types of robots including: legged robot, wheeled robot, autonomous underwater vehicle, unmanned aerial vehicle, mobile, robotic eye. They also can be used in many fields like: medical, military, agriculture, mobile, educational, domestic, air crafts, ships and even toys. They can perform repetitive jobs that are boring and stressful, also they do menial and in humane tasks that people don't want to do[1].

The robotic manipulator includes a robot arm, wrist and gripper. Its mission is to put an object taken by the gripper into an arbitrary position, the robot arm is the part that enables robots to move a certain location , while that of the robot wrist is responsible for the rotation of the object grasped by robot manipulator. It is formed from a group of links and joints. The links are the rigid members connecting the joints and axes. The axes are the portions of the robot responsible for the movement with respect to the links. The mechanical joints which are used to build the robotic arm manipulator are composed of five main parts: two of the joint are transitional and three are rotational

Manipulators can be divided according to different perspectives one of which is the type of motion : SCARA (Selective Compliant Articulated Robot) which used in industrial processes, anthropomorphic robot arm which looks like the human arm to a large extent, Cartesian robot which are known for high accuracy, cylindrical and spherical robot arm[2].

3)History:
In 1478 when Leonardo Davinci was 26 years old, he started drawing pictures about robotic arm which scientists create it 500 years later. It was in the shape of a front wheel and to the surprise it was fully automated. In 1495, he converted his pictures into reality by inventing the first humanoid robot with many degrees of freedom: legs, ankles, knees and hips.and here we introduce important decades of development[3].

1963

Rancho robotic arm

which greatly helped the disabled

1969

stanford arm

- the first computer controlled arm

1974

silver arm

- **Had touch and pressure sensor micro computer**

1979

SCARA Robot

Used in the electronics industry

4) Some useful definition [4].

- Co-ordinates(frame) : are described by a referring to a reference coordinates system that can be replaced anywhere we want and can be transferred by transformation matrix
- End-effectors: the terminal part of robotic arm which is responsible for the task of the robot.
- Work space: the allowable region for the robotic
- Degree of freedom: is the place that we identify in order to know the places of other particles of robotic arm.
- Joints: it connects the links to each other and their arm to types according to their motion, like: revolute joints (rotary) and prismatic which move horizontally.
- Inertia: a property of the object used in describing the rotational movement just like the mass which is used in transitional motion.

  EX: Newton second law can be expressed in two ways according to the type of motion

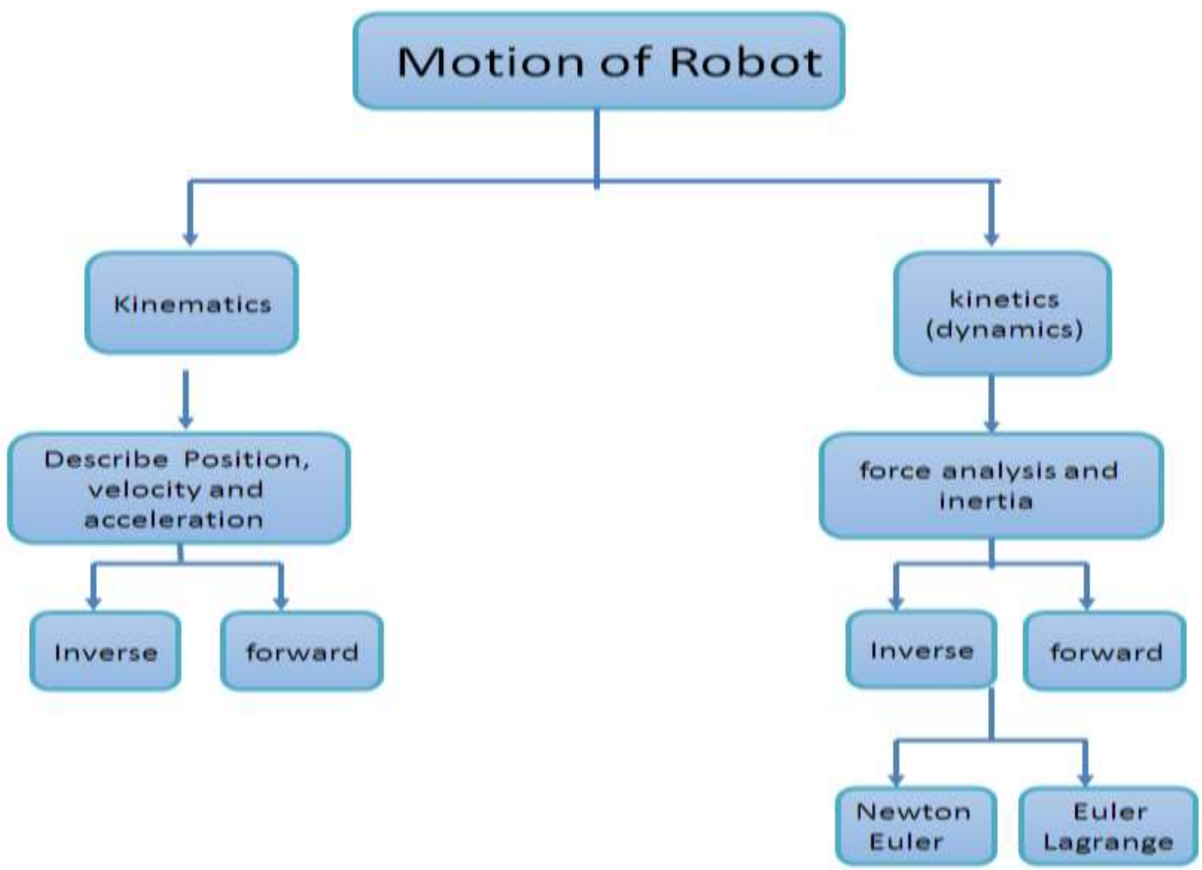
$$\sum \tau = I\,\ddot{\theta} \quad , \quad (1a)$$

$$\sum F = ma \quad . \quad (1b)$$

  EX: rotational kinematic energy = $\frac{1}{2}\, I\,\dot{\theta}$, $(2a)$

  While the transitional kinetic energy= $\frac{1}{2}mv^2$. $(2b)$

- inertia tensor: defines the resistance to changes to the rotation of the arm
- Denavit-Hartenburg convention: Used for representing a kinematic model of the link where it helps in choosing the reference frames in robot system. [3]
- inertial frame : is a kinematical device for the geometrical description of motion without considering the masses or forces involved
- Rotational and transitional matrices: They are necessary for expressing the position of one link with respect to another .

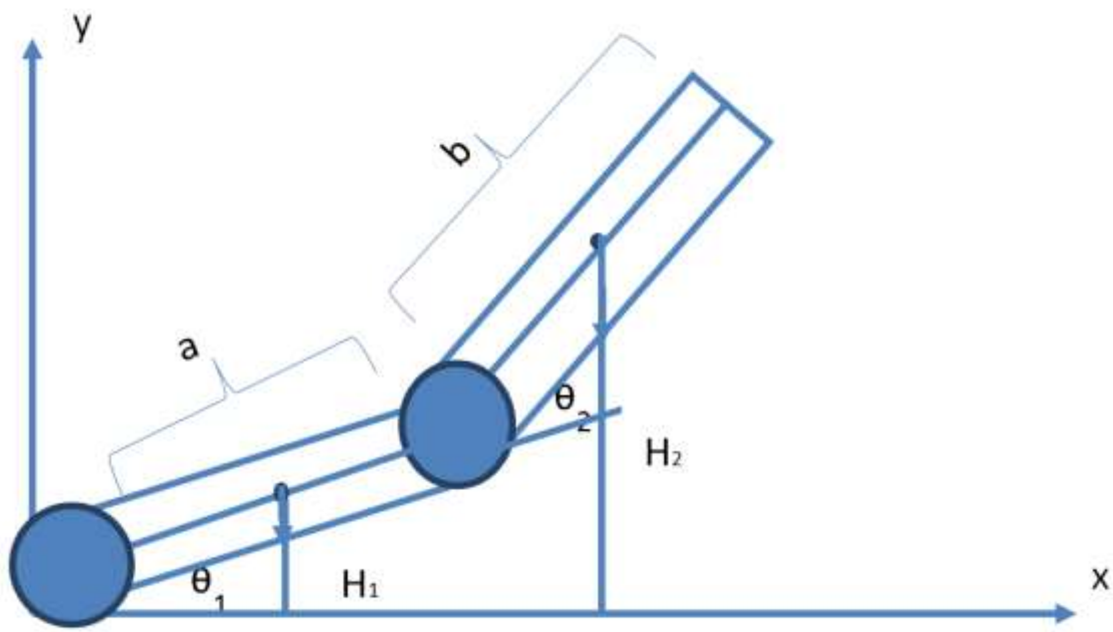


5)The Dynamics and kinematics of a robot manipulator:

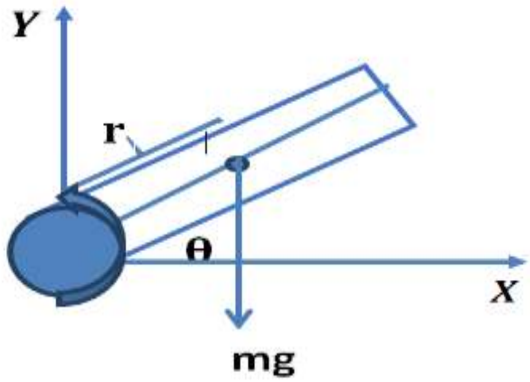
To have a complete knowledge on how the robot moves in a specific path, and how it grabs different objects, the system should be analyzed and represented by a dynamic model. This could be achieved with two different procedures: dynamics and Kinematics.

The aim of robot dynamics is studying its motion by considering the forces affecting it, and it includes two main methods: Newton-Euler equation and Euler-Lagrange equation both of which will be introduced with a bit of more details in the coming sections. While the Kinematics ignores the circumstances and the forces causing this motion; instead the system should be analyzed geometrically with the knowledge of the initial conditions of location, acceleration and velocity. Moreover, there are two different approaches for the Kinematics: inverse kinematics, where the final location reached by the gripper, which is in contact with the surrounding environment, is known so it is required to know the velocity and joint angles necessary to reach this position, and the forward is the opposite ; as the final destination is obscure

This two-example shown the difference between kinematics and dynamics



$$p_a = a\cos\theta_1 + a\sin\theta_1 \quad (3)$$
$$p_b = p_a + p_{ba} \quad (4a)$$
$$= (a\cos\theta_1 + a\sin\theta_1) + (b\cos\theta_2 + b\sin\theta_2) \quad (4b)$$
$$v_a = \frac{dp_a}{dt} \quad (5a)$$
$$v_b = \frac{dp_b}{dt} \quad (5b)$$
$$K.E = \frac{1}{2}I_1\dot{\theta}_1{}^2 + \frac{1}{2}I_2\dot{\theta}_2{}^2 \quad (6)$$
$$P.E. = m_1gh_1 + m_2gh_2 \quad (7)$$



b)

$$T = I\,\ddot{\theta} + mgr\ \cos\theta \qquad (8)$$

Where(I) is moment of inertia

5.1)Dynamics of motion

The dynamic motion is described by considering the rate with which the robot disposition changes with time relative to the joint torque exerted by actuators. This relationship can be formulated using a group of differential equations representing motion that governs the response of robot linkage to input joint torque. These differential equations are called equation of motion. These equations are obtained by two methods: Euler Lagrange equation and Newton Euler equations. [4]

Newton-Euler Vs Euler-Lagrange:

The principle aim is the deduction of a model representing the dynamics of the robot in a short time and with full representation. There are two approaches that use different techniques to reach these equations and choosing one of them is a personal preference. Newton -Euler: each link of manipulator is treated separately via two methods: first, forward method describes the linear and angular motion by using initial conditions. Second, backward method: we get forces and terminal conditions torques. While in Euler-Lagrange the manipulator is treated as a whole where the kinetic and the potential energies are the main concern.

5.1.1)Newton-Euler Equation:

The basis of Newton-Euler derivation depends on three important mechanics laws[5].

- there is a reflex reaction to every action and it has the same magnitude , but its effect is in a reversed direction.
- Newton's second laws: the rate with which the momentum changes is equivalent to the summation of the forces applied to the link. [5]

$$\frac{d(mv)}{dT} = F \qquad , \qquad (8.a)$$

$$\therefore \vec{F} = m\vec{a}\ . \qquad , \qquad (8.b)$$

- The rate with which the angular momentum changes is equivalent to the total torque affecting the link

$$\frac{d(I_o\omega)}{dt} = \tau \qquad , \qquad (9)$$

Where, I_O is moment of inertia

ω is angular velocity of the link.

the previous three variables are represented in an inertial frame with an origin positioned at the center of mass.so, I_o should be transformed and this is achieved by the rotational matrix R .

$$I_o = RTR^T \quad , \quad (10)$$
$$\omega_o = R\omega \quad , \quad (11)$$
$$\tau_o = R\tau \quad , \quad (12)$$

Therefore,

$$\frac{d(I_o\omega_o)}{dt} = \frac{d(RIR^T R\omega)}{dt} \quad , \quad (13a)$$
$$= \frac{d(R\ I\omega)}{dt} \quad , \quad (13b)$$
$$= \dot{R}\omega I + RI\dot{\omega} \quad , \quad (13c)$$

And the equation describing the rate with which the angular momentum of the link changes relative to the frame

$$\tau = R^{\mathrm{T}}\tau_o \quad , \quad (14a)$$
$$= R^{\mathrm{T}}\left(\dot{R}\mathrm{I}\omega + R\mathrm{I}\dot{\omega}\right) \quad , \quad (14b)$$
$$= R^{\mathrm{T}}\dot{R}\mathrm{I}\omega + \mathrm{I}\dot{\omega} \quad , \quad (14c)$$

After simplification the final torque expression becomes:

$$\tau = R^{\mathrm{T}}S(\omega_O)R\ \mathrm{I}\omega + \mathrm{I}\dot{\omega} \quad , \quad (14d)$$
$$= S(R^{\mathrm{T}}\omega_O)\ \mathrm{I}\omega + \mathrm{I}\dot{\omega} \quad , \quad (14e)$$
$$= S(\omega)\mathrm{I}\omega + \mathrm{I}\dot{\omega} \quad , \quad (14f)$$
$$=\omega \times (\mathrm{I}\omega) + \mathrm{I}\dot{\omega} \quad . \quad (14g)$$

So, the general forms representing the mass balance and moment balance can be written as follows:

$$\boxed{F = ma} \quad , \quad (15)$$
$$\boxed{\tau = \omega\ \times (\mathrm{I}\omega) + \mathrm{I}\dot{\omega}} \quad . \quad (16)$$

5.1.2)Euler-Lagrange equation

Because of the difficulties that are faced during the dynamic analysis of a model, somehow there had to be another facilitated way to reach the same results, and this was what Lagrange did.he developed a system that purely depends on mathematics without the need to draw free body diagrams, or vectors and gave the results with high efficiency and resembled the results reached by Newtonian mechanics. The idea he used to formulate the law was (The principle of least actions) : an object tends to reach the maximum possible stability without consuming much energy and we want to know this path that confirms the laws of universe such as conservation of energy , conservation of momentum and laws of gravity[6].

The equation he derived could be applied to any system is given by the following:

$$\boxed{\frac{d}{dt}\left(\frac{\partial F}{\partial \dot{x}}\right) - \frac{\partial F}{\partial x} = 0} \quad . \quad (17)$$

And to express a mechanical system, F is substituted with (L) which is the Lagrangian term:

$$\boxed{L = T - V} \quad , \quad (18)$$

Where: 1)(T) is the kinetic energy, and in case of the robot manipulator which consists of several links, it can be expressed by the summation of two components:

- -Energy due to the translational velocity of the center of mass

$$\tfrac{1}{2}mv^T v \quad , \quad (19a)$$

- -Energy due to rotation about the center of mass:

$$\frac{1}{2}\omega^T I\omega \quad , \quad (19b)$$

(I): the inertia tensor, and it defines the resistance to changes to the rotation of the arm

$$I = \begin{vmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{vmatrix} \quad . \quad (20)$$

2)(V) is the total potential energy of the links

- The potential energy of the i^{th} link is given by:

$$P_i = m_i g^T r_i \quad , \quad (21a)$$

$$\therefore V = \sum_{i=1}^{n} P_i \quad . \quad (21b)$$

5.2)Kinematics

Kinematics is another branch of science that is concerned with studying the motion by neglecting the forces causing it and concentrating on the velocities, trajectories and acceleration.. Joint sensors can calculate velocities and trajectories. We can set apart direct and inverse kinematics. In Direct kinematics , the joint angle is known, therefore it can be used to estimate the final location of the gripper but in inverse kinematics ,it is completely the inverse process where the joint angles are the unknowns, and could be determined by using the final location reached by the gripper .If the two previous methods are put into a comparison, we recognize that the inverse kinematics is much more sophisticated than forward kinematics due to the several solutions for a single position of end-effector.

Kinematics equations:

The way for getting the values of the joint velocities by using the final location of the gripper can be illustrated as follows[7].

By considering the x-y plane and a manipulator with length l moving with an angleθ:

$$\therefore \quad x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad , \quad (22a)$$

$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad , \quad (22b)$$

$$dx_e = \frac{\partial x_e(\theta_1, \theta_2)}{\partial\theta_1} d\theta_1 + \frac{\partial x_e(\theta_1, \theta_2)}{\partial\theta_2} d\theta_1 \quad , \quad (22c)$$

$$dy_e = \frac{\partial y_e(\theta_1, \theta_2)}{\partial\theta_1} d\theta_1 + \frac{\partial y_e(\theta_1, \theta_2)}{\partial\theta_2} d\theta_2 \quad . \quad (22d)$$

By deriving an equation expressing the Jacobian in terms of dx and dq we find that:

$$J = \frac{dx}{dq} \quad , \quad (23a)$$

Where, $dx = \frac{dx}{dy}$,$dq = \frac{d\theta_1}{d\theta_2}$

$$\therefore \quad J = \begin{vmatrix} \frac{\partial x_e(\theta_1, \theta_2)}{\partial\theta_1} & \frac{\partial x_e(\theta_1, \theta_2)}{\partial\theta_2} \\ \frac{\partial y_e(\theta_1, \theta_2)}{\partial\theta_1} & \frac{\partial y_e(\theta_1, \theta_2)}{\partial\theta_2} \end{vmatrix} \quad , \quad (23b)$$

Which is The partial derivatives of x,y with joint displacement θ_1, θ_2

$$\therefore \quad J = \begin{vmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{vmatrix} \quad , \quad (23c)$$

Where J is the relation between the end point and the joint movement

Since, $$dx = J.dq \quad , \quad (24a)$$

And by differentiating with respect to time:

$$\frac{dx}{dt} = J\frac{dq}{dt} \quad , \quad (24b)$$

$$\therefore V = J\dot{q} \quad . \quad (24c)$$

Relation between velocity of joint and end effector

$$v_e = \dot{\theta}J_1 + J_2\dot{\theta}_2 \quad , \quad (25)$$

Where:

- v_e is the resultant end effector velocity
- $\dot{\theta}J_1$velocity of end effector induced by the first joint only
- $J_2\dot{\theta}_2$velocity of end effector induced by the second joint only

The general equation will be:

$$\dot{p} = J_1\dot{q}_1 \pm ------ \; J_n\dot{q}_n \quad , \quad (26)$$

Where , $\dot{p}$ is the end effector velocity

$\dot{q}$ is the end effector velocity of one link

From equation (25):

$$\therefore \dot{q} = J^{-1}\,v \quad , \quad (27)$$

$$J = \begin{vmatrix} \frac{dx}{d\theta_1} & \frac{dx}{d\theta_2} \\ \frac{dy}{d\theta_1} & \frac{dy}{d\theta_2} \end{vmatrix} \quad , \quad (28)$$

$$J^{-1} = \begin{vmatrix} \frac{d\theta_1}{dx} & \frac{d\theta_2}{dx} \\ \frac{d\theta_1}{dy} & \frac{d\theta_2}{dy} \end{vmatrix} \quad , \quad (29)$$

$$\therefore \boxed{\dot{\theta} = \frac{v_x\cos(\theta_1 + \theta_2) + v_y\sin(\theta_1 + \theta_2)}{\sin(\theta_2)}} \quad , \quad (30)$$

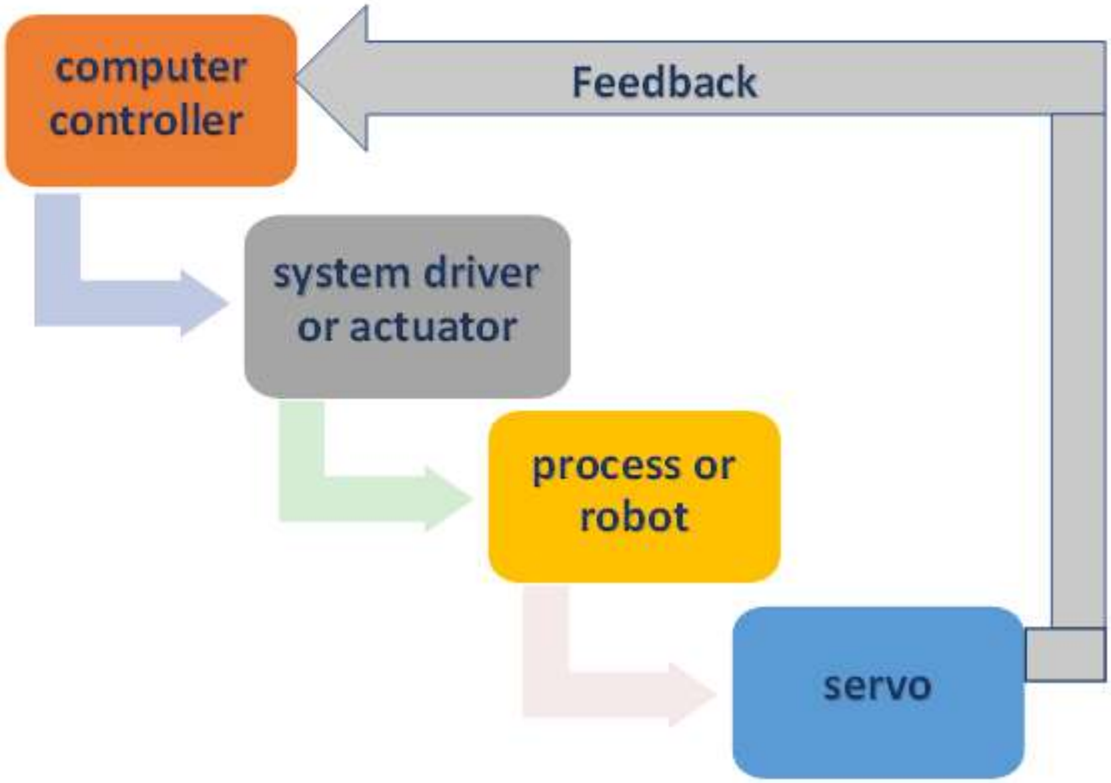
$$\boxed{\ddot{\theta}_2 = \frac{v_x\,(\cos(\theta_1) + \cos(\theta_1 + \theta_2)\,) + v_y(\sin(\theta_1) + \sin(\theta_1 + \theta_2))}{\sin(\theta_2\,)}} \quad , \quad (31)$$

So, our problem is solved.

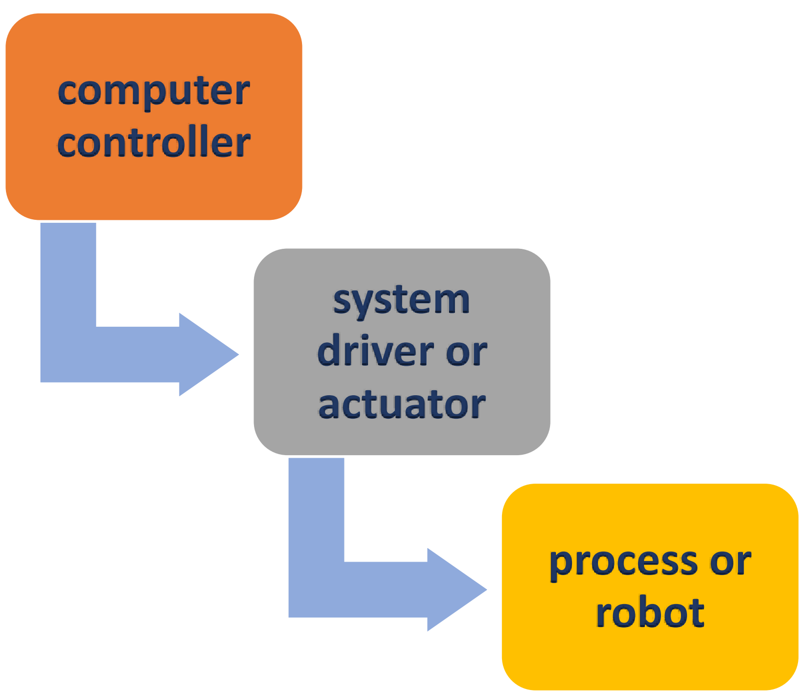
6) Control System:

The thing that is responsible for the motion of robot is the control system. Control system is essential to guide the robot to the desired position. Control systems vary; there is an open loop system as well as a closed loop system. The open loop system is not based on the feedback unlike the closed loop system and this is obviously illustrated in figures(8),(9). The feedback is essential for reducing the errors which are likely to happen. So we can deduce that the closed loop system is better than the open loop system from the figure (10) [8].

closed loop system



Open loop system

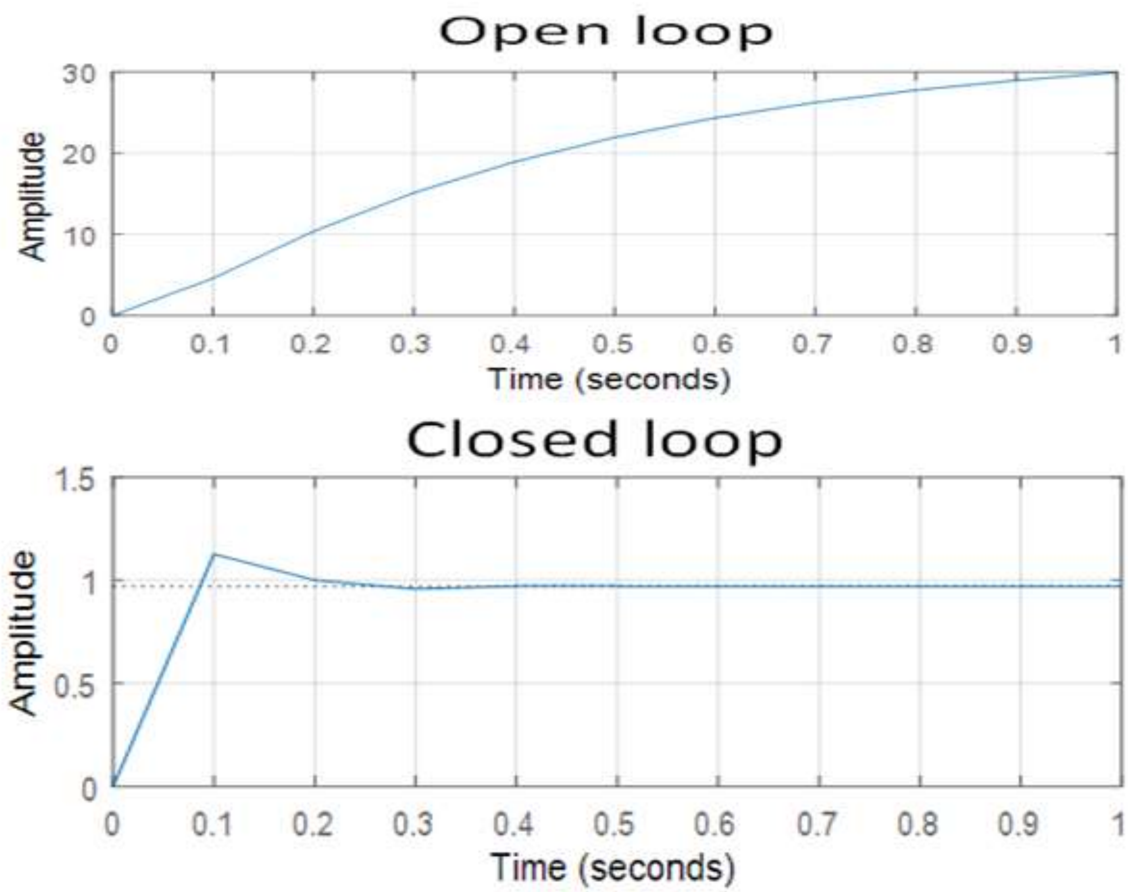


- ✓ The response of both closed loop and open loop using mat lab toolbox :

Mat lab helps us to see the response of both closed loop and open loop

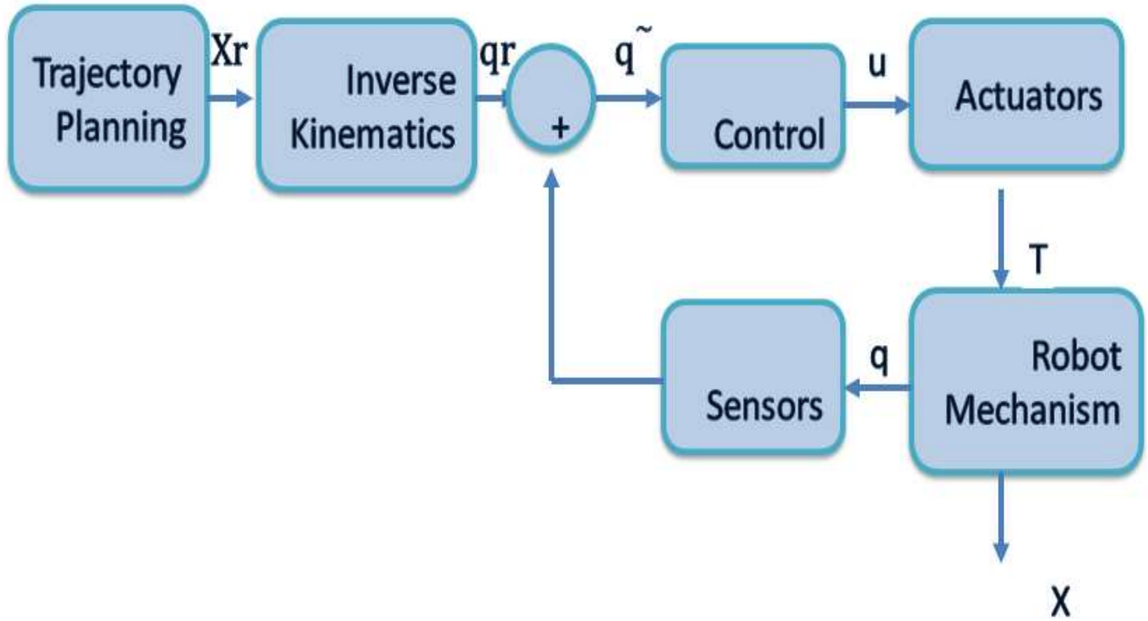
the figure show that in open loop system the overshoot that happens is greater than the one in the close loop system. The figures also show the rising time in the open loop is longer than one in closed loop so the closed loop system reach to steady state faster than open loop system

that doesn't mean that open loop is bad and doesn’t use. But the choice of the control system that we use is optional as it is related to the task of the robot.



Figure(11) shows the general control system. the input to control system is the desiered end point .the variable X is the desired position ,$q^{\sim}$ is the positional error,u the control system output ,$q_r$is the desired position calculated by the inverse kinematics.

Simply, we can express any control system by this diagram



6.1) Position control and trajectory tracking

Now, we are concerned with a different issue; that is if we are given a desired trajectory, how the joint torque should be selected to make the manipulator follow this trajectory. That is what will be illustrated below[9].

The main followed equation is called "Dynamic motion of manipulator robot "which states that:

$$\boxed{M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau} \quad , \quad (32a)$$

Where,

- $M(\theta)$ *is* moment
- τ is the vector of the motor torques.

By considering θ=θ_d for simplicity the equation will be:

$$M(\theta_d)\ddot{\theta_d} + C(\theta_d,\dot{\theta}_d)\dot{\theta}_d + N(\theta_d,\dot{\theta_d}) = \tau \quad , \quad (32b)$$

But these equations represent the open loop control law which is not real, because we never know the current position of the robot. Also, the robot can go away from its path even it begins in a right way .so, we need a feedback which is done by the closed loop control system discussed before.

By equating (32a) & (32b) we find

$$\mathrm{M}(\theta)\ddot{\theta} = \mathrm{M}(\theta_d)\ddot{\theta_d} \quad , \quad (32c)$$

And since $\mathrm{M}(\theta)$ is always positive for all possibilities of θ :

$$\ddot{\theta} = \ddot{\theta}_d \quad . \quad (33)$$

By summing up the equation of the feedback we get the equation of the computed torque control law which states that:

$$\boxed{\tau = M(\theta)\left(\ddot{\theta}_{\mathrm{d}} - \mathrm{K_v}\dot{e} - \mathrm{K_P}\mathrm{e}\right) + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta})} \quad , \quad (34)$$

where
$$e = \theta - \dot{\theta} \quad , \quad (35)$$

$\mathrm{K_v}$, $\mathrm{K_P}$ .are constant gain matrices

By substituting into (32a) into (34) :

$$\mathrm{M}(\theta)(\ddot{e} + \mathrm{K_v}\dot{e} + \mathrm{K_P}\mathrm{e}) = 0 \quad , \quad (36)$$

$$\boxed{\therefore (\ddot{e} + \mathrm{K_v}\dot{e} + \mathrm{K_P}\mathrm{e}) = 0} \quad , \quad (37)$$

We will use this equation to govern the error between the actual and desired trajectories

We can divide equation (6-2-5) into two parts as follows:

$$\tau = \underbrace{M(\theta)\ddot{\theta}_{d} + C\dot{\theta} + \mathrm{N}}_{\tau_{ff}} + \underbrace{\mathrm{M}(\theta)(-\mathrm{K}_{v}\dot{e} - \mathrm{K}_{P}e)}_{\tau_{fb}} \quad , \qquad (38)$$

Where,

- τ_{ff} is the feed forward component ,it provides the torque necessary to drive the system along its nominal path .
- τ_{fb} is the feed backward component, it provides the correction torque to reduce any errors in the trajectory of the manipulator.

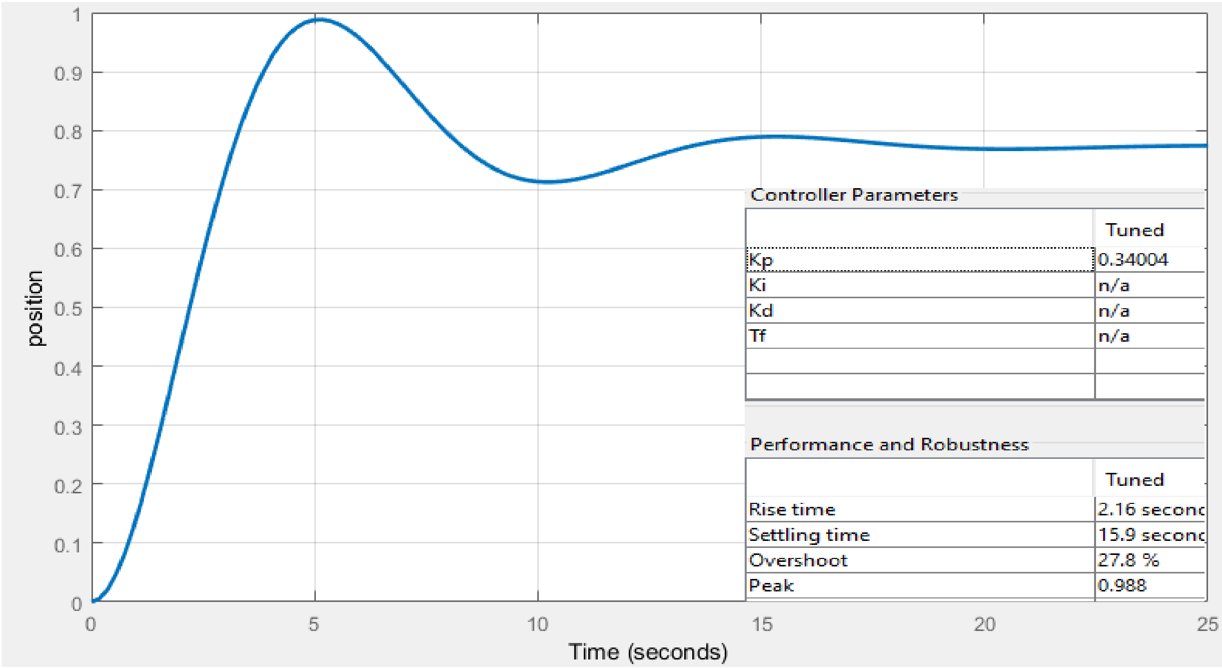
6-2) PID Control:

After studying and predicting the motion of the robot theoretically, what is missing now is knowing the suitable voltage with which the motor operates and the duration necessary for applying this voltage in order to do its task with highest possible efficiency , in other words: moving with high speed and minimum possible oscillations and reaching the exact position without great deviation.

Here comes the role of PID control-which stands for (proportional integral derivatives).The destination is reached with high performance by doing successive processes :to begin with, the difference between the initial location (measured value) and the final desired position(Reference value) is calculated, then this difference undergoes several operations where it is multiplied by a proportional constant (kp),gets differentiated ,integrated and results are multiplied by other constants and the outputs of the previous operations become the inputs needed for setting the voltage which consequently changes the power of the motor and the movement of the robotic arm then the final position after these modifications is read by a position sensor and it is compared with the reference value . It is noticed that the arm won’t reach the desired position until after several loops because of the inertia so the arm moves about the desired location. It is worth saying that the operations performed on the error measured are not available in every control system; choosing between them is optional, but we should take into consideration that the accuracy will be greatly affected as each one is responsible for a specific property of the motion .For example, if it is (p control)only, there will be overshoot, while (D) control causes the slow movement [10].

✓ Disscuss the effect of parameters using matlab tool box :

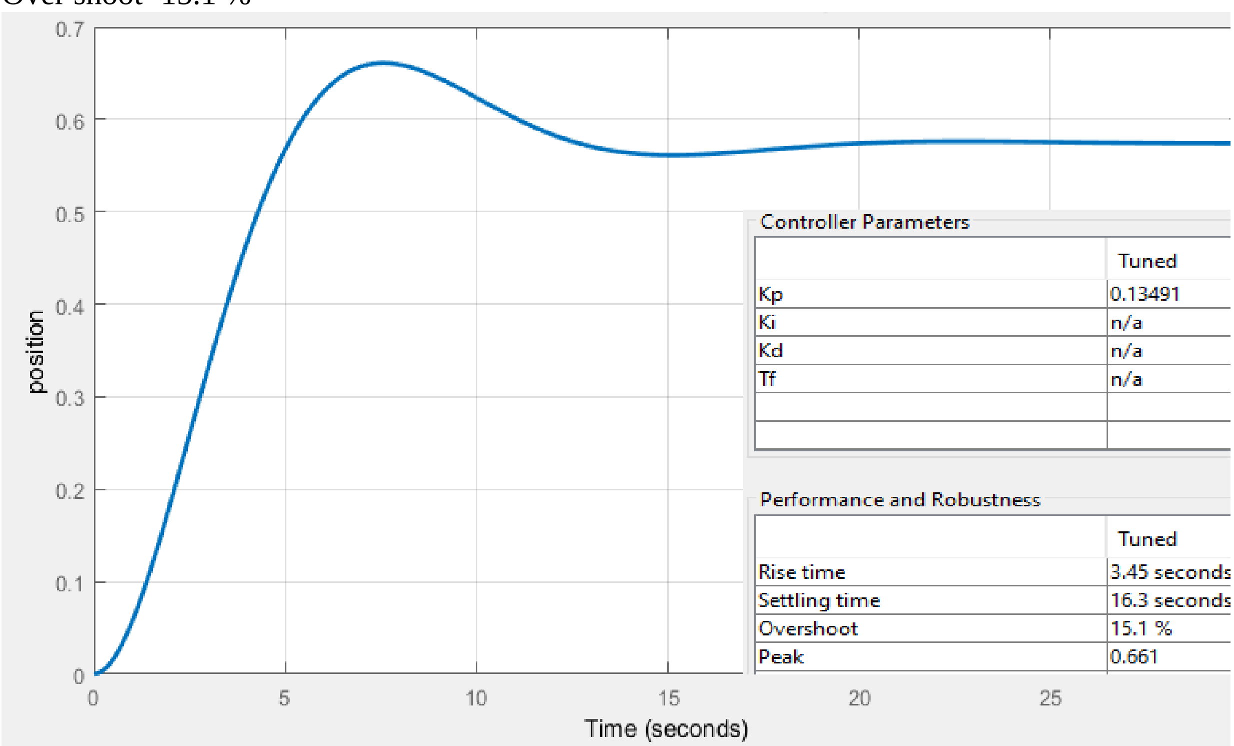
P controller



Results:

Rising time=3.45 seconds

Over shoot=15.1 %



Results:

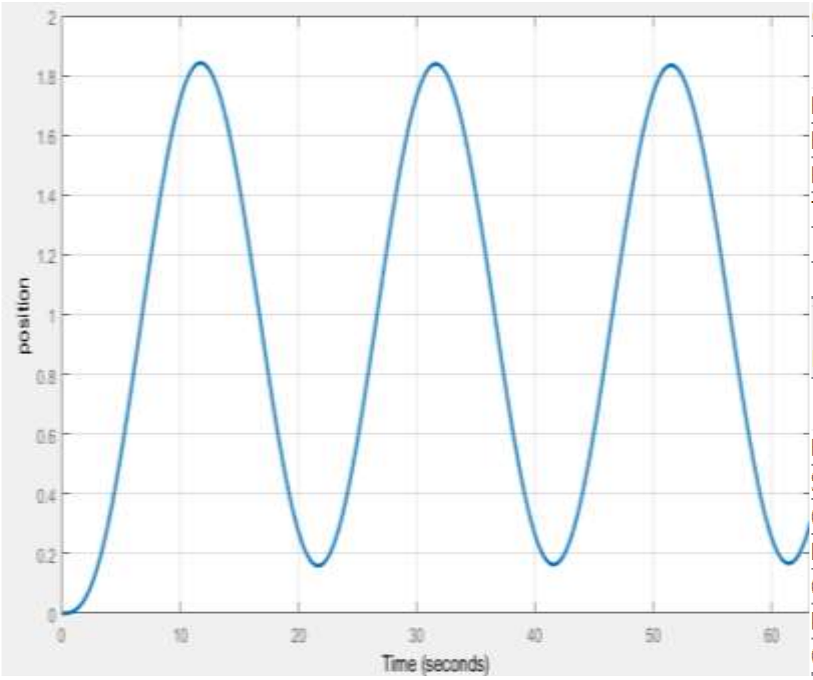
Rising time=2.16seconds

Overshoot= 27.8%

Conclusion:

- Increase kp lead to over shoot at a certain limit
- Increase kp reduce the rising time

I Controller



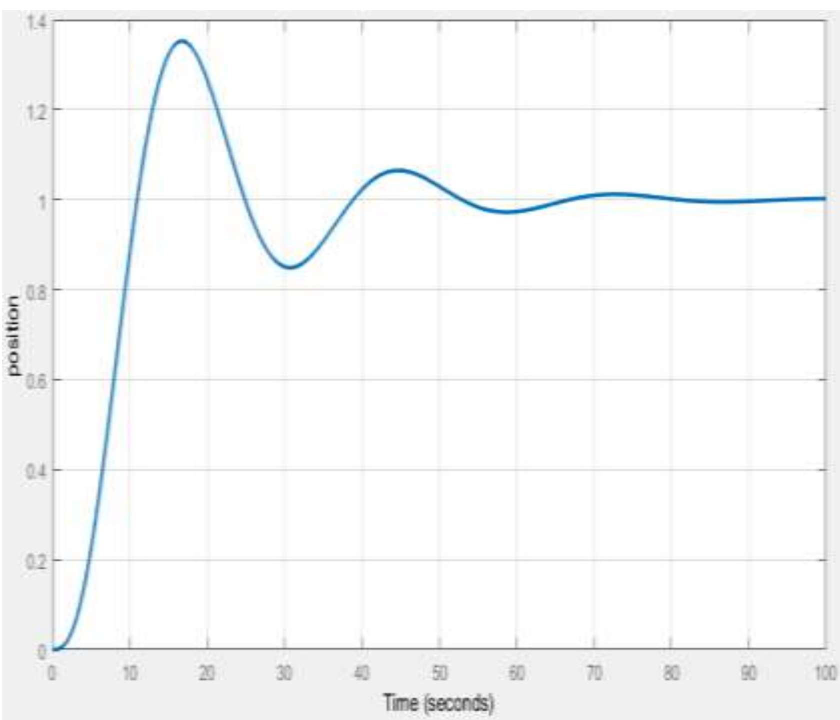
Controller Parameters

	Tuned
Kp	n/a
Ki	0.04983
Kd	n/a
Tf	n/a

Performance and Robustness

	Tuned
Rise time	3.91 seconds
Settling time	1.54e+04 seconds
Overshoot	83.8 %
Peak	1.84
Gain margin	0.0296 dB @ 0.316 rad/s
Phase margin	0.124 deg @ 0.316 rad/s
Closed-loop stability	Stable

Results : Rising time = 3.91 seconds, overshoot=83.8%



Controller Parameters

	Tuned
Kp	n/a
Ki	0.020504
Kd	n/a
Tf	n/a

Performance and Robustness

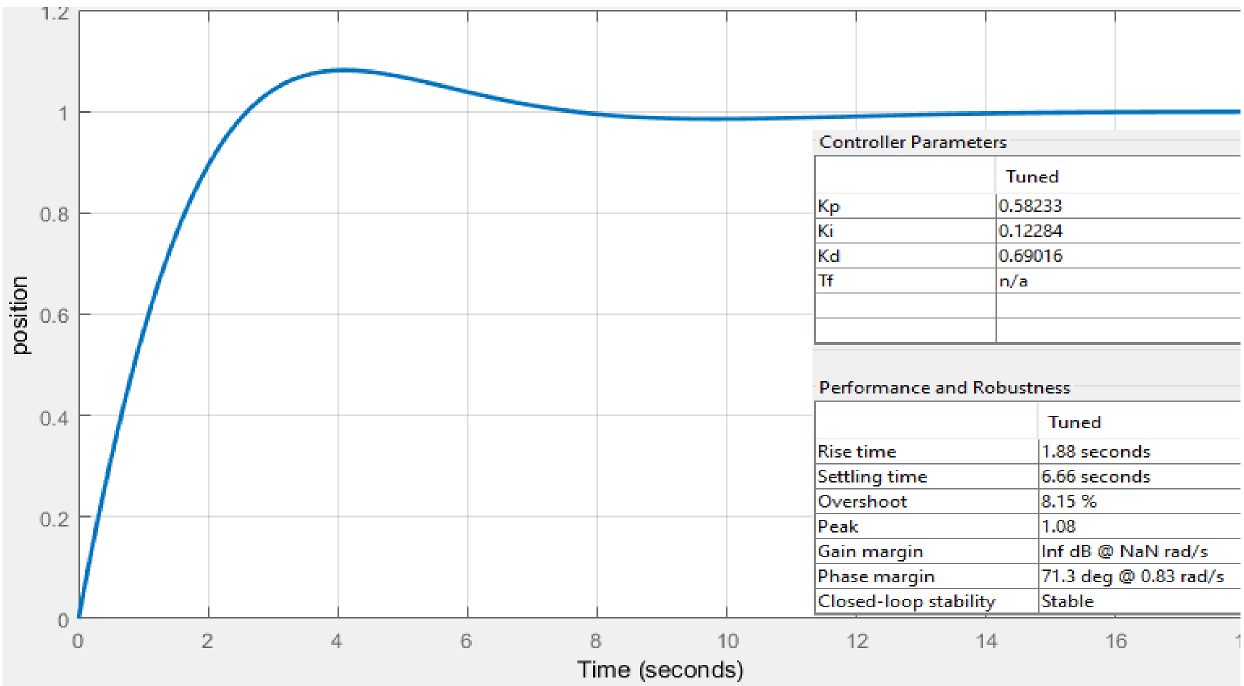
	Tuned
Rise time	6.56 seconds
Settling time	62.2 seconds
Overshoot	35.3 %
Peak	1.35
Gain margin	7.74 dB @ 0.316 rad/s
Phase margin	36.4 deg @ 0.182 rad/s
Closed-loop stability	Stable

Results : Rising time = 3.91 seconds, overshoot=83.8%

Conclusion:

- Increase ki increase the overshoot
- After a certain value, Increase ki leading to oscillation
- Increase ki making the system unstable
- Reducing the rising time

PID Controller



Results:

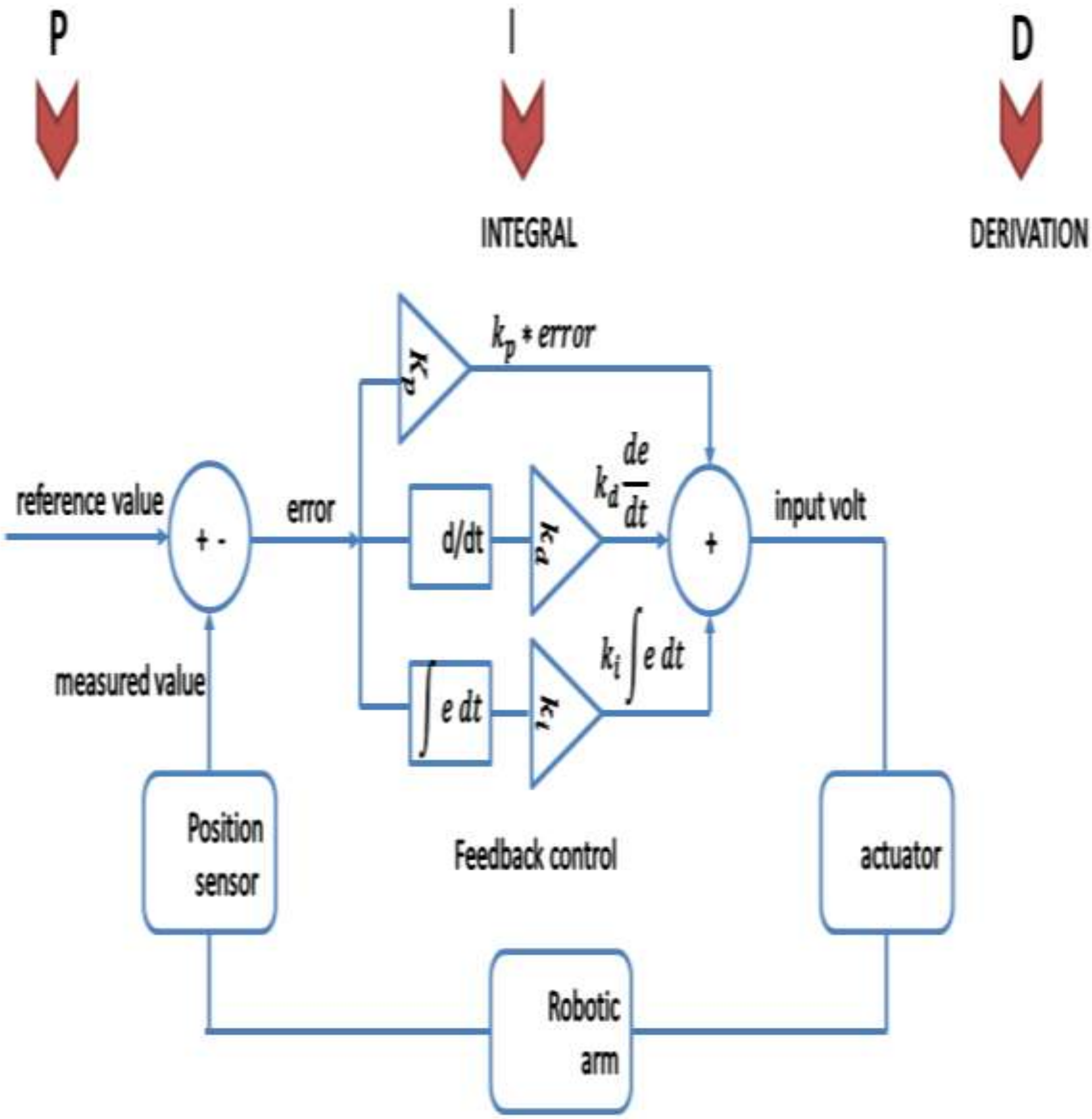
Rising time=1.88 seconds
Overshoot=8.15%

Conclusion

- PID decrease the overshoot by adjust gains
- PID decrease the rising time

The previous graphs show that each parameter has effect to the system and the system will be high efficiency in the existence of the tree parameter .Although there is some system doesn't need all these parameters

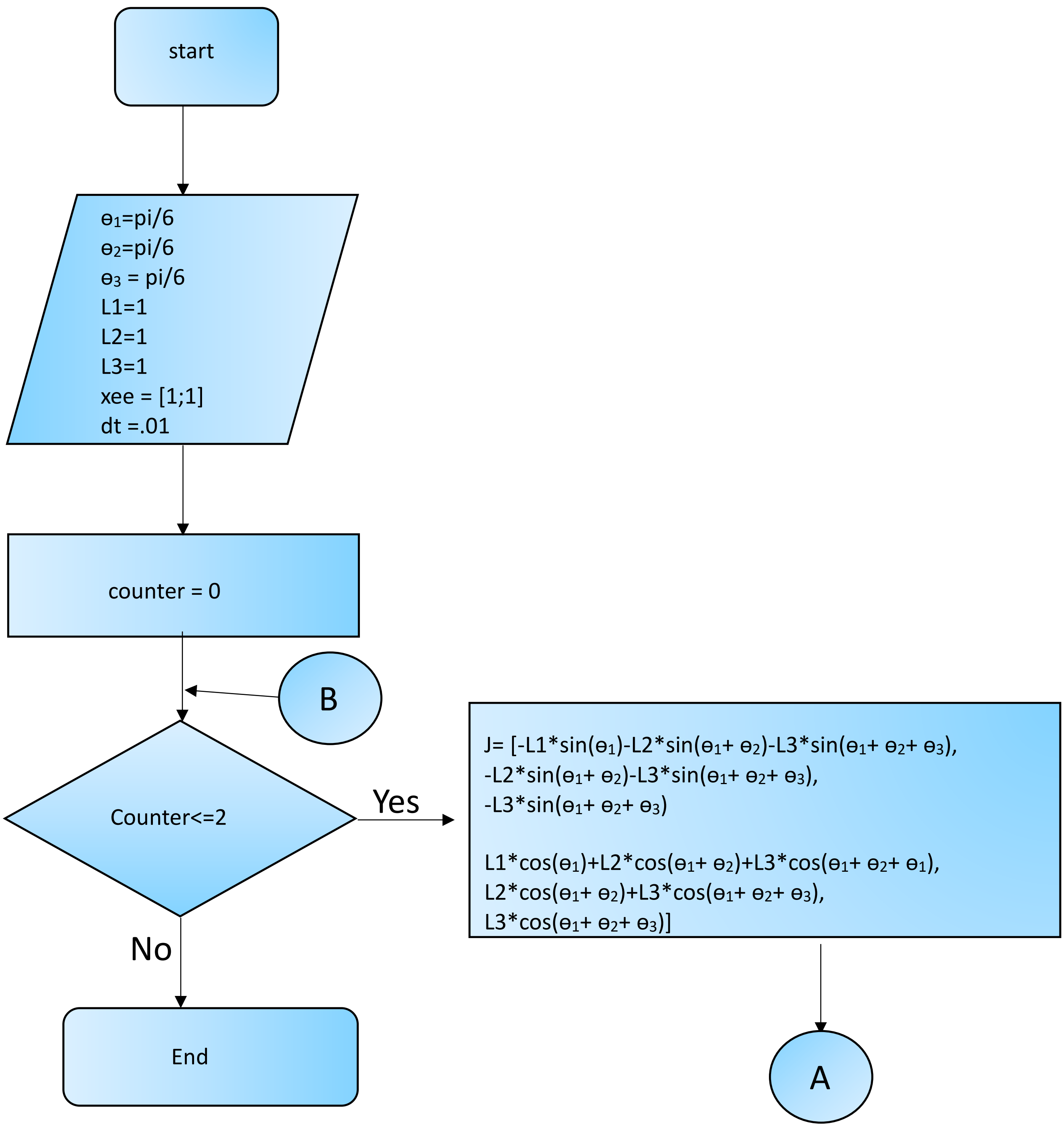
PID CONTROL

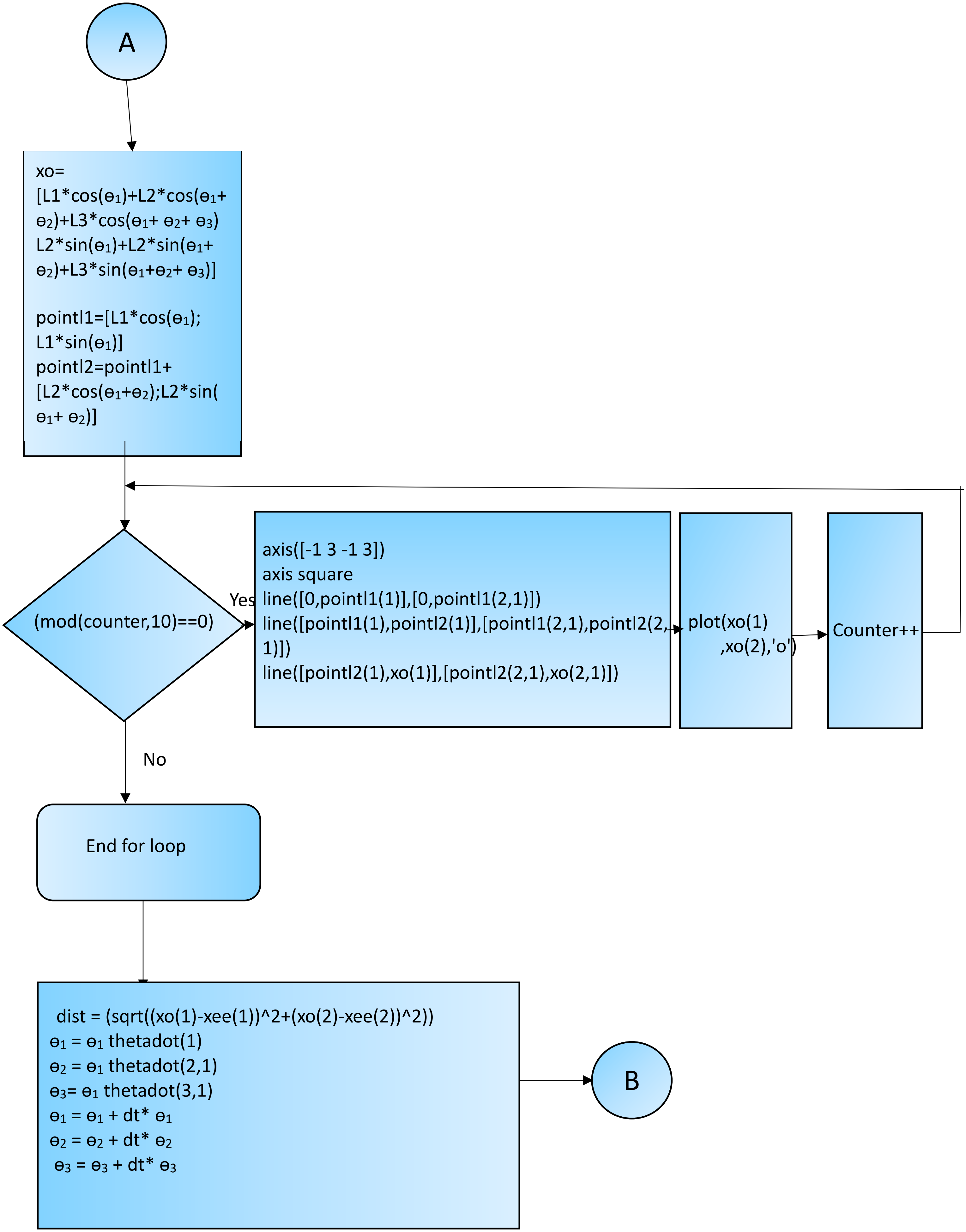


P
I
D
INTEGRAL
DERIVATION
k_p
$k_p * error$
$k_d \frac{de}{dt}$
reference value
+ -
error
d/dt
k_d
+
input volt
measured value
$\int e\,dt$
k_i
$k_i \int e\,dt$
Position sensor
Feedback control
actuator
Robotic arm

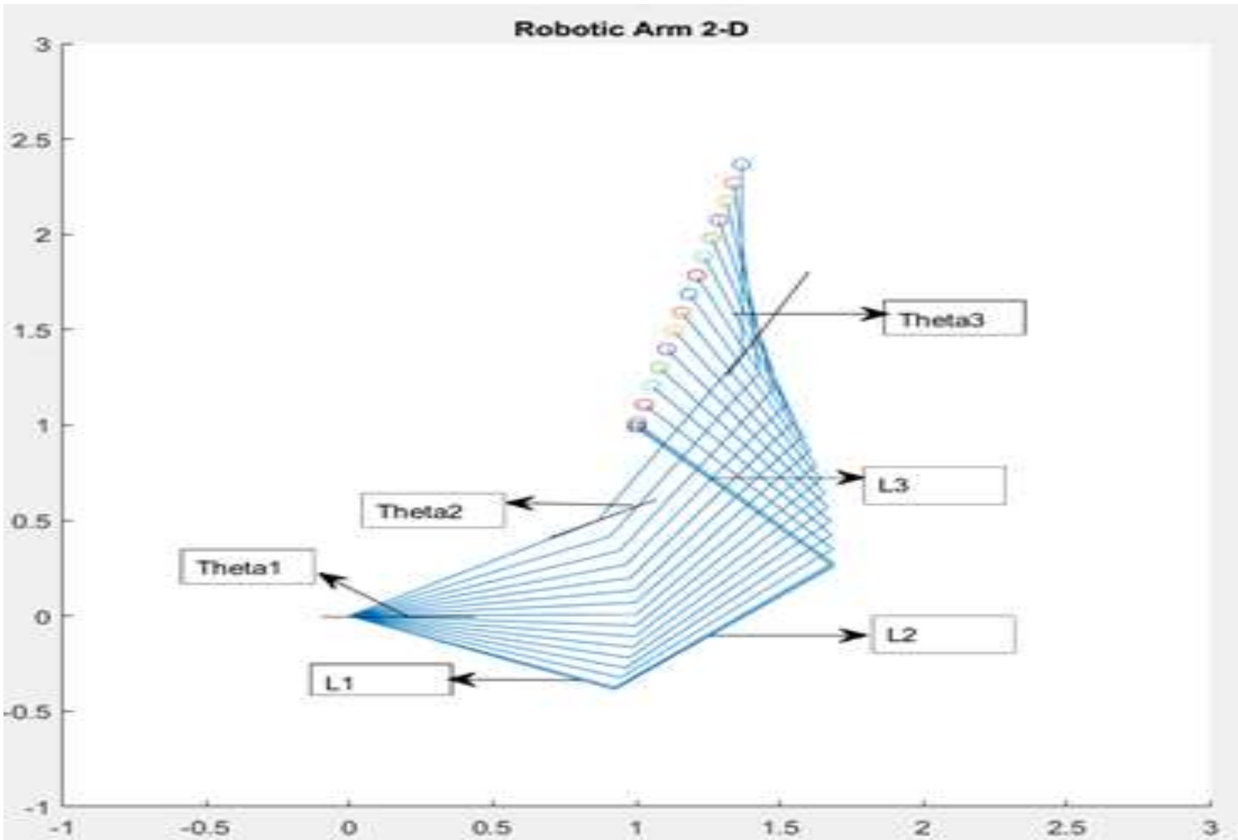
7)Flow chart for mat lab modeling and results

Now we can introduce the explanation of mat lab results using flow chart as follows

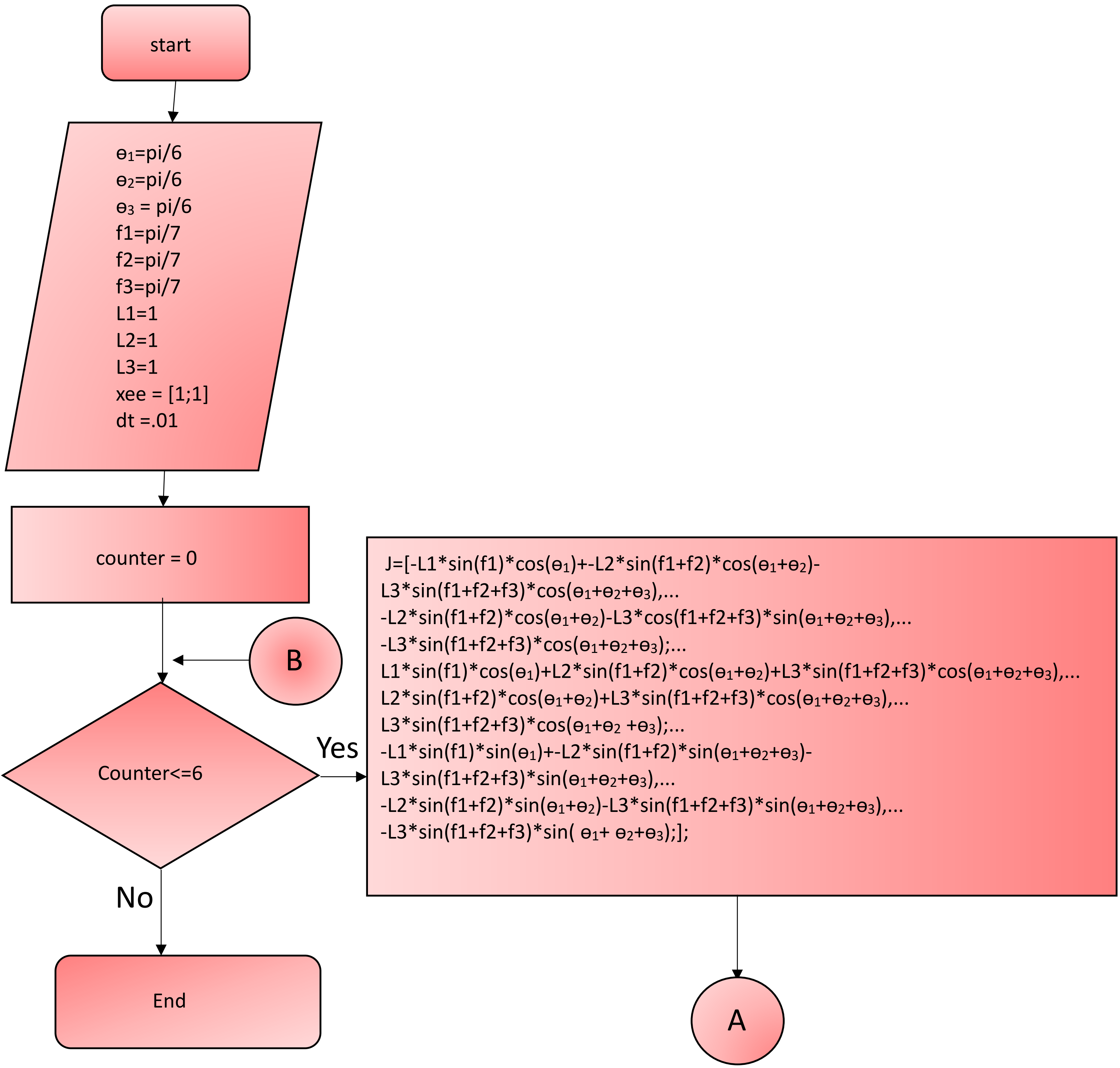


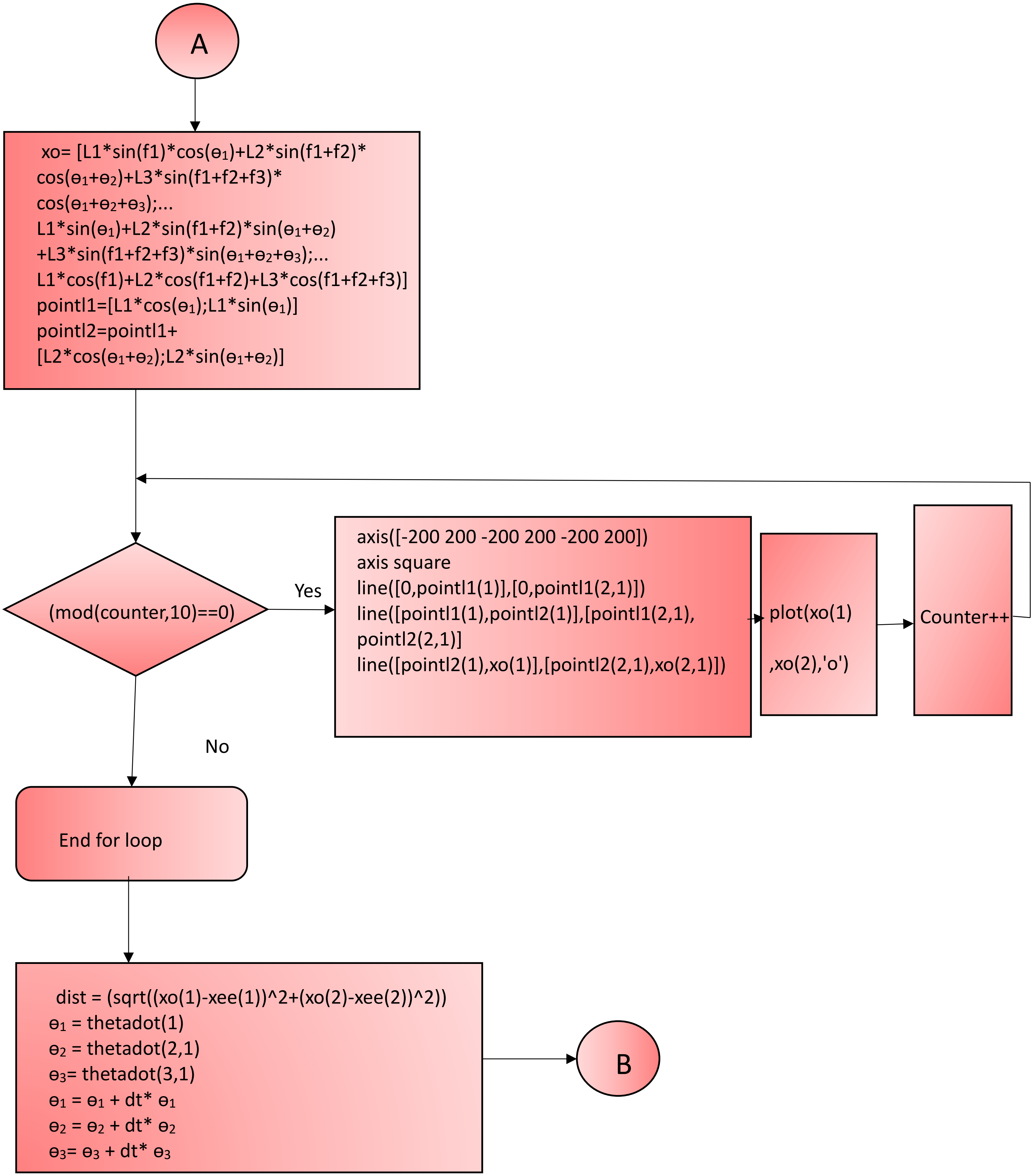

A
xo=
[L1*cos(ɵ1)+L2*cos(ɵ1+
ɵ2)+L3*cos(ɵ1+ ɵ2+ ɵ3)
L2*sin(ɵ1)+L2*sin(ɵ1+
ɵ2)+L3*sin(ɵ1+ɵ2+ ɵ3)]
pointl1=[L1*cos(ɵ1);
L1*sin(ɵ1)]
pointl2=pointl1+
[L2*cos(ɵ1+ɵ2);L2*sin(
ɵ1+ ɵ2)]
(mod(counter,10)==0)
Yes
axis([-1 3 -1 3])
axis square
line([0,pointl1(1)],[0,pointl1(2,1)])
line([pointl1(1),pointl2(1)],[pointl1(2,1),pointl2(2,
1)])
line([pointl2(1),xo(1)],[pointl2(2,1),xo(2,1)])
plot(xo(1)
,xo(2),'o')
Counter++
No
End for loop
dist = (sqrt((xo(1)-xee(1))^2+(xo(2)-xee(2))^2))
ɵ1 = ɵ1 thetadot(1)
ɵ2 = ɵ1 thetadot(2,1)
ɵ3= ɵ1 thetadot(3,1)
ɵ1 = ɵ1 + dt* ɵ1
ɵ2 = ɵ2 + dt* ɵ2
ɵ3 = ɵ3 + dt* ɵ3
B

By putting the previous equations on matlab we can see this result :

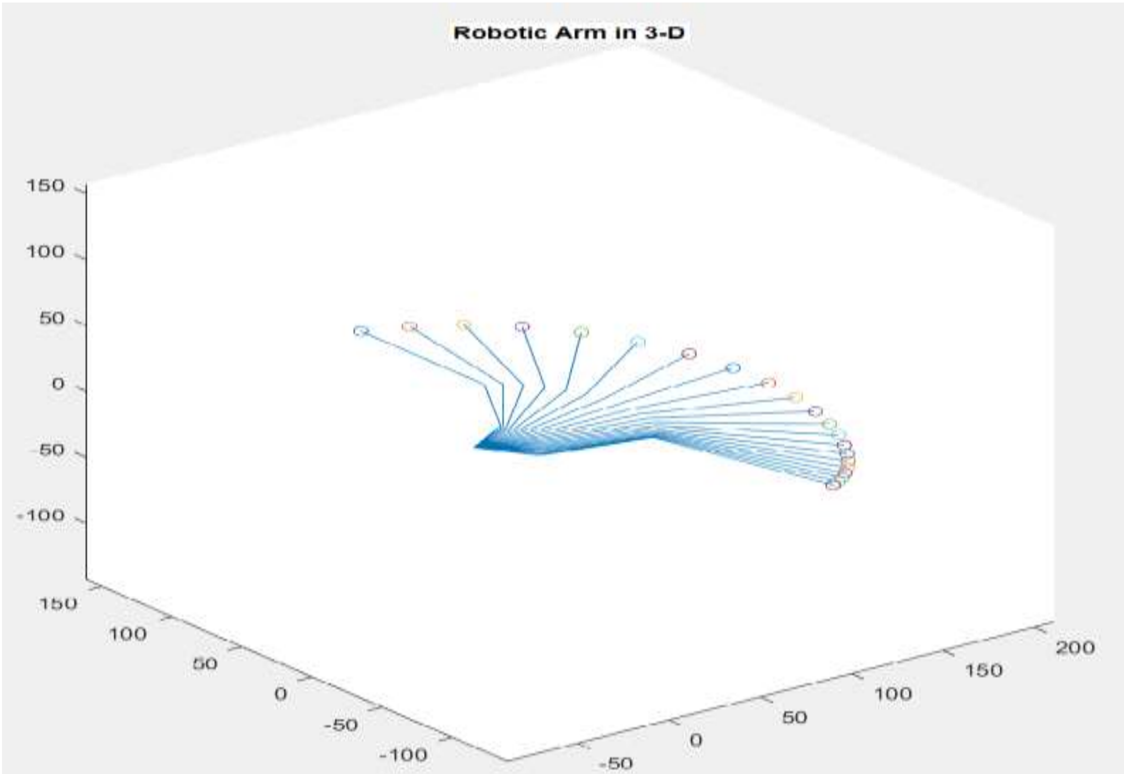


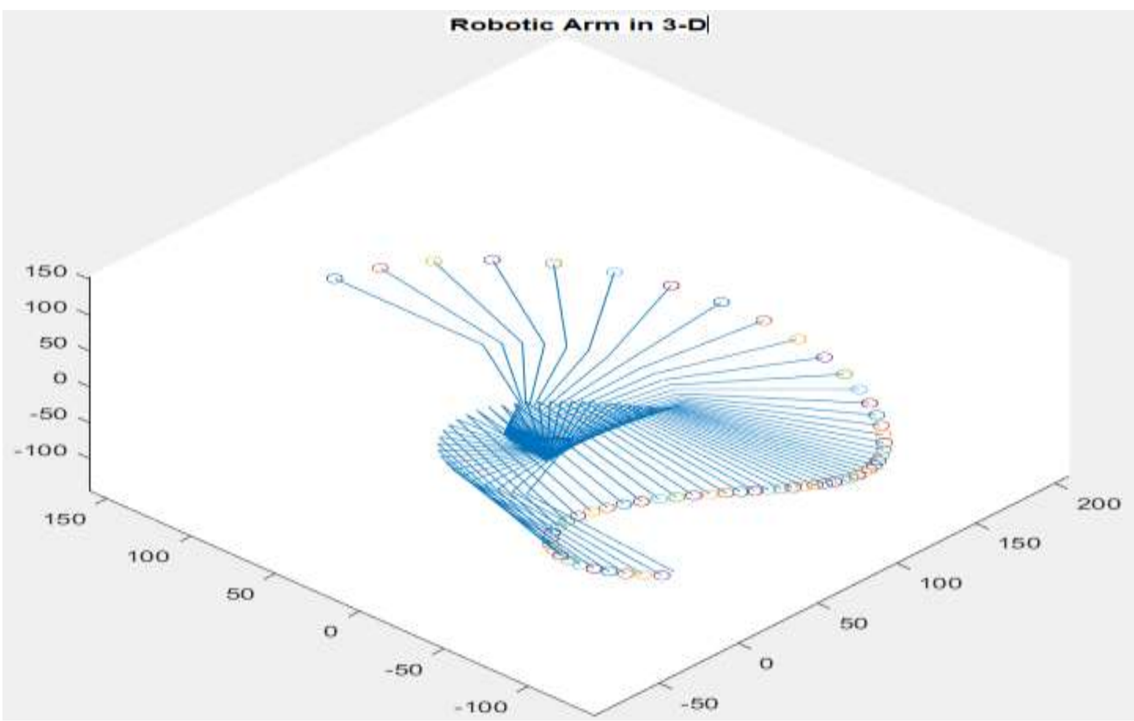
We can also modify the code to make it 3-D as follows:




A
xo= [L1*sin(f1)*cos(ɵ1)+L2*sin(f1+f2)*
cos(ɵ1+ɵ2)+L3*sin(f1+f2+f3)*
cos(ɵ1+ɵ2+ɵ3);...
L1*sin(ɵ1)+L2*sin(f1+f2)*sin(ɵ1+ɵ2)
+L3*sin(f1+f2+f3)*sin(ɵ1+ɵ2+ɵ3);...
L1*cos(f1)+L2*cos(f1+f2)+L3*cos(f1+f2+f3)]
pointl1=[L1*cos(ɵ1);L1*sin(ɵ1)]
pointl2=pointl1+
[L2*cos(ɵ1+ɵ2);L2*sin(ɵ1+ɵ2)]
(mod(counter,10)==0)
Yes
axis([-200 200 -200 200 -200 200])
axis square
line([0,pointl1(1)],[0,pointl1(2,1)])
line([pointl1(1),pointl2(1)],[pointl1(2,1),
pointl2(2,1)]
line([pointl2(1),xo(1)],[pointl2(2,1),xo(2,1)])
plot(xo(1)
,xo(2),'o')
Counter++
No
End for loop
dist = (sqrt((xo(1)-xee(1))^2+(xo(2)-xee(2))^2))
ɵ1 = thetadot(1)
ɵ2 = thetadot(2,1)
ɵ3= thetadot(3,1)
ɵ1 = ɵ1 + dt* ɵ1
ɵ2 = ɵ2 + dt* ɵ2
ɵ3= ɵ3 + dt* ɵ3
B

+ By putting these equations on matlab we can see this result :





8)Applications

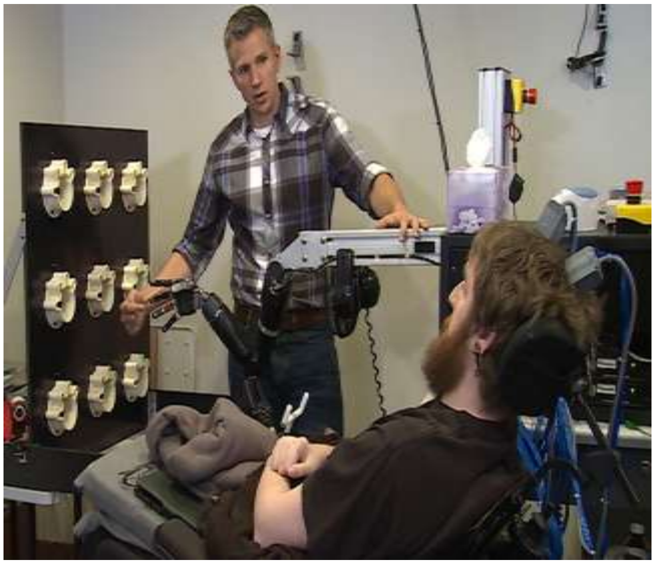
❖ Flute Player:

In 1737,JacquesVaucanson built the flute player , he built it to play different tones , but it was a paved road to the prosthetic robotic arm , as he inserted it in the skin[11].

❖ Chess Player :

In 1769,Wolfgang Von Kemplen built an automated chess player .his body was from wood ,but the left arm and the hand was fully automated. The arm was controlled humanly by the director. He first orders the limb to move, the center of the arm to be over the chest piece then the arm would grasp it[12].

❖ Mind controlled prosthetic robot arm



Scientists improved the first mind controlled prosthetic arm to people who are deprived from sensation. By this they could feel different temperature, textures, and determine which finger is being touched. It provides almost natural sensation. They begin by developing mind controlled robotic arm, but including sensation was the miracle in the invention.”The amazing part was the ability to sense again after ten years of paralysis “this was said by Nathan Copeland, the first one to be introduced into this kind of robotic arm. Nathan had an accident which left him with unmovable arms and legs and also, without sensation in them.This is done by inserting tiny microelectrodes in the places of sensation in the brain .The electrodes read the electrical signal from the brain neurons and relate it with another region to control sensation[13].

❖ Robotic Arm surgery :

Diathermy has great importance in the medical field. However, doctor’s hand isn’t of high degree of accuracy. Robotic arm had many uses in surgery such as[14]:

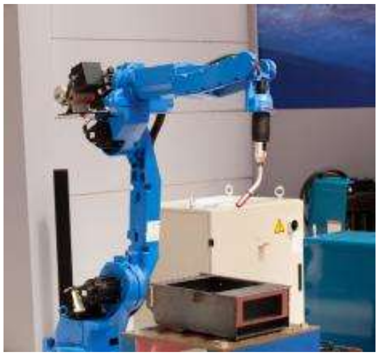
1) It provides accuracy as same as three doctors.
2) It is remote controlled which allows the doctor to do the operation thousands of miles away, which saved the lives of many patients who died because of late intervention.
3) It is highly specific, which means it do spots not scars like doctors , allowing the patient to leave hospital in couple of hours.
4) It can go to very wide parts of the body, even to repair a hole between two chambers of the heart.
5) Neurosurgery: it enables us to put electrodes in the brain in case of treating nerve disease like: epilepsy.

- RIO Robotic Arm:

Knee resurfacing is a very difficult operation;, it requires high accuracy and restrictions. So, many doctors refuse to do it, preferring to remove knee totally which widens the area of the recurred patients. Now, using robotic arm in orthopedic introduce us into new decade, it provide high degree of accuracy in operation and placement of implants[15].

- Children:

Using robotic arm in surgery is faster, less pain, fewer scars, less time to stay at hospitals, less pain which make it better for children.



- Cyber Knife :

It is the use of robotic arm in treatment of cancer. It has many advantages like: First, It provides high accuracy than normal surgery. second, It saves healthy tissues. Third, it is painless[16].

- Mouth cancer treatment:

It enables doctors to remove cancer in the mouth without leaving a permanent sign. Before using the robotic arm, the way to cure it was by surgery including of splitting of the jaw .Due to its hardness doctors preferred chemo and radio therapy which had disadvantages too. Thanks to the robotic arm, the use of surgery and harmful chemotherapy has been greatly reduced

9)conclusion

All in all, we first define what the robotics is in general and its importance and show how it help human in his life especially in work. Then we show that there are many types of robotics and this type depends on its function. From these different types, we choose the robotic manipulator to show its types and to study its motion which consists of two main description of motion: dynamics and kinematics. The robot's equations of motion describe the relation between input like joint torque and output motion.

Kinematics deals with velocities, accelerations and trajectories with neglecting the forces that causes the motion. On the other hand, dynamics describe the time rate change in the robot motion.

Second import thing to complete description of robot motion, we talked about, is control system and its importance. control system is used to calculate the error which happen as when the robot take the signal to go to fixed position , it doesn't reach due to this error . We come over this problem by control system which sends a feedback. Control has many different methods. The robot task is responsible for choice of the control method .Finally, we show importance of the robotic arm by giving some application[17].

10)References:

[1] Allonrobots.com. (2018). *Types of robots*. [online] Available at: http://www.allonrobots.com/types-of-robots.html [Accessed 6 Apr. 2018].
[2] Slideshare.net. (2018). *Fundamental of robotic manipulator*. [online] Available at: https://www.slideshare.net/snkalepvpit/fundamental-of-robotic-manipulator [Accessed 6 Apr. 2018].
[3] **Control**, R. (2018). *Robot Dynamics and Control - PDF Drive*. [online] Pdfdrive.net. Available at: https://www.pdfdrive.net/robot-dynamics-and-control-webnode-e8408999.html [Accessed 6 Apr. 2018].
[4] Craig, J. (2018). *Introduction to robotics*. NY NY: Pearson.
[5] Bing.com. (2018). [online] Available at: http://www.bing.com/cr?IG=109441B9088942368292C8A2DB7915CB&CID=2629045F9394666B35A20F

96923B67A6&rd=1&h=Jv5A-QTJPt2-3XDMXu5rbq8x4Nb7-YcHLJirAgldS6c&v=1&r=http%3a%2f%2fwww.cds.caltech.edu%2f%7emurray%2fbooks%2fMLS%2fpdf%2fmls94-manipdyn_v1_2.pdf&p=DevEx.LB.1,5559.1 [Accessed 6 Apr. 2018].
[6] YouTube. (2018). *Understanding the Euler Lagrange Equation*. [online] Available at: https://www.youtube.com/watch?v=08vJyA-XD3Q [Accessed 6 Apr. 2018].
[7] Bing.com. (2018). [online] Available at: https://www.bing.com/cr?IG=0019EABE86C54018BCF25A36C77B1A85&CID=07244492CBB769183F974F5BCA18686C&rd=1&h=vdV-z3eGlGYI7UNCOqj38QNOC1pz-a3t4tatXmnq8_0&v=1&r=https%3a%2f%2focw.mit.edu%2fcourses%2fmechanical-engineering%2f2-12-introduction-to-robotics-fall-2005%2flecture-notes%2fchapter5.pdf&p=DevEx.LB.1,5542.1 [Accessed 6 Apr. 2018].
[8] Electrical4u.com. (2018). *Control System | Closed Loop Open Loop Control System*. [online] Available at: https://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/ [Accessed 6 Apr. 2018].
[9] Bing.com. (2018). [online] Available at: https://www.bing.com/cr?IG=0019EABE86C54018BCF25A36C77B1A85&CID=07244492CBB769183F974F5BCA18686C&rd=1&h=vdV-z3eGlGYI7UNCOqj38QNOC1pz-a3t4tatXmnq8_0&v=1&r=https%3a%2f%2focw.mit.edu%2fcourses%2fmechanical-engineering%2f2-12-introduction-to-robotics-fall-2005%2flecture-notes%2fchapter5.pdf&p=DevEx.LB.1,5542.1 [Accessed 6 Apr. 2018].
[10] Pacontrol.com. (2018). *PID Control Theory*. [online] Available at: http://www.pacontrol.com/PID.html [Accessed 6 Apr. 2018].
[11] YouTube. (2018). *WF-4RII Flight of the Bumble Bee*. [online] Available at: https://www.youtube.com/watch?v=0wDZI15tiR0 [Accessed 6 Apr. 2018].
[12] YouTube. (2018). *Robotic Arm playing Chess (homemade)*. [online] Available at: https://www.youtube.com/watch?v=x5r_znpqW14 [Accessed 6 Apr. 2018].
[13] Devlin, H. (2018). *Mind-controlled robot arm gives back sense of touch to paralysed man*. [online] the Guardian. Available at: https://www.theguardian.com/science/2016/oct/13/mind-controlled-robot-arm-gives-back-sense-of-touch-to-paralysed-man [Accessed 5 Apr. 2018].
[14] YouTube. (2018). *MAKO robotic arm assisted surgery*. [online] Available at: https://www.youtube.com/watch?v=KtEGIM0D278 [Accessed 6 Apr. 2018].
[15] **YouTube**. (2018). *RIO Robotic Arm System-The Future in Orthopedics*. [online] Available at: https://www.youtube.com/watch?v=EXX0zru1wqo [Accessed 6 Apr. 2018].
[16] YouTube. (2018). *Radiosurgery: What is CyberKnife Robotic RadioSurgery?*. [online] Available at: https://www.youtube.com/watch?v=ZblOZFJ_mf0 [Accessed 6 Apr. 2018].
[17] **Bing.com**. (2018). [online] Available at: http://www.bing.com/cr?IG=205BF497DBD648FFAA3B3249BBA7D8AD&CID=0A9FC5157DC46B0E1B0FCEDC7C6B6A6D&rd=1&h=KqQ_-9SKBm6jENicydHrEQTHa126dh4lLHrK3k7vXxs&v=1&r=http%3a%2f%2fwww.emo.org.tr%2fekler%2f4f226e3f94cd5bb_ek.pdf&p=DevEx.LB.1,5749.1 [Accessed 6 Apr. 2018].

11)APPENDIX:

The Code in mat lab in 2-D Coordinates:

```
clc
 clear all
 close all
 %Initial conditions:
 Theta1=pi/6;%the angle with x-axis for first link
Theta2=pi/6;%the angle with x-axis for second link
Theta3 = pi/6;%the angle with x-axis for third link
L1=1;%the length of first link
L2=1;%the length of second link
```

```
L3=1;%the length of third link
W=[1  0  0;  0  1  0;  0  0  1];  %weighting  matrix  (identity  all  joints  penalized  equally
fig()applying the code in matlab
xee = [1;1]; %Desired end effector location
dt =.01; %time step
counter = 0;
 for i = 0:dt:2 %End time should be adjusted for the desired accuracy
 %Jacobian (2x3) Partial xee with respect to Theta1,2,3
                                                                                          %code in 2-D
 J= [-L1*sin(Theta1)-L2*sin(Theta1+Theta2)-L3*sin(Theta1+Theta2+Theta3),...
-L2*sin(Theta1+Theta2)-L3*sin(Theta1+Theta2+Theta3),...
-L3*sin(Theta1+Theta2+Theta3);...

L1*cos(Theta1)+L2*cos(Theta1+Theta2)+L3*cos(Theta1+Theta2+Theta3),...
L2*cos(Theta1+Theta2)+L3*cos(Theta1+Theta2+Theta3),...
L3*cos(Theta1+Theta2+Theta3)];
 %Pseudoinverse given the Jacobian and Weighting matrix
 Jpseudo = inv(W)*J'*inv(J*inv(W)*J');
 %End - effector location
xo = [L1*cos(Theta1)+L2*cos(Theta1+Theta2)+L3*cos(Theta1+Theta2+Theta3);...
L2*sin(Theta1)+L2*sin(Theta1+Theta2)+L3*sin(Theta1+Theta2+Theta3)];
 %Calculate the location of the middle two joints
 pointl1 = [L1*cos(Theta1) ; L1*sin(Theta1)];
pointl2 = pointl1 + [L2*cos(Theta1+Theta2);L2*sin(Theta1+Theta2)];
 %Plot
 if (mod(counter,10)==0) %plots every 10 iterations
axis([-1 3 -1 3])
axis square
line([0,pointl1(1)],[0,pointl1(2,1)])
hold on
line([pointl1(1),pointl2(1)],[pointl1(2,1),pointl2(2,1)])
line([pointl2(1),xo(1)],[pointl2(2,1),xo(2,1)])
plot(xo(1),xo(2),'o')
pause(.1)
 end
 dist = (sqrt((xo(1)-xee(1))^2+(xo(2)-xee(2))^2)); %Distance between
 %current EE location and Final EE location is used to normalize the error
%value (x dot)
 %Angular velocity of each angle:
 thetadot = Jpseudo*((xee-xo)/dist);
 theta1 = thetadot(1);
theta2 = thetadot(2,1);
 theta3 = thetadot(3,1);
 %Updtate each angle based on calculated angular velocity
 Theta1 = Theta1 + dt*theta1;
Theta2 = Theta2 + dt*theta2;
 Theta3 = Theta3 + dt*theta3;
counter = counter +1;
```

```
 end
```

The code in 3-D Coordinates changing The angle with X-Y Coordinates (Theta):

```
clc
 clear all
 close all
 %Initial conditions:
 Theta1=pi/6;%angle with x-axis for first link
Theta2=pi/6;%angle with x-axis for second link
Theta3 = pi/6;%angle with x-axis for third link
f1=pi/7;%angle with z-axis for the first link
f2=pi/7;%angle with z-axis for the second link
f3 = pi/7;%angle with z-axis for the third link
L1=70;%length of first link
L2=70;%length of second link
L3=70;%length of third link
 W=[1 0 0; 0 1 0; 0 0 1]; %weighting matrix (identity all joints penalized equally
xee = [1;1;1]; %Desired end effector location
dt =.01; %time step
counter = 0;
 for i = 0:dt:6 %End time should be adjusted for the desired accuracy
 %Jacobian (3x3) Partial xee with respect to Theta1,2,3
%the first derivative w.r.t theta
 J= [-L1*sin(f1)*cos(Theta1)+-L2*sin(f1+f2)*cos(Theta1+Theta2)-
L3*sin(f1+f2+f3)*cos(Theta1+Theta2+Theta3),...
-L2*sin(f1+f2)*cos(Theta1+Theta2)-L3*cos(f1+f2+f3)*sin(Theta1+Theta2+Theta3),...
-L3*sin(f1+f2+f3)*cos(Theta1+Theta2+Theta3);...
%the angles with axis
L1*sin(f1)*cos(Theta1)+L2*sin(f1+f2)*cos(Theta1+Theta2)+L3*sin(f1+f2+f3)*cos(Theta1+The
ta2+Theta3),...
L2*sin(f1+f2)*cos(Theta1+Theta2)+L3*sin(f1+f2+f3)*cos(Theta1+Theta2+Theta3),...
L3*sin(f1+f2+f3)*cos(Theta1+Theta2+Theta3);. % fig()changing Theta for i=0:dt:2
%the second derivative w.r.t theta
-L1*sin(f1)*sin(Theta1)+-L2*sin(f1+f2)*sin(Theta1+Theta2)-
L3*sin(f1+f2+f3)*sin(Theta1+Theta2+Theta3),...
-L2*sin(f1+f2)*sin(Theta1+Theta2)-L3*sin(f1+f2+f3)*sin(Theta1+Theta2+Theta3),...
-L3*sin(f1+f2+f3)*sin(Theta1+Theta2+Theta3);];
 %Pseudoinverse given the Jacobian and Weighting matrix
%End - effector location
 xo =
[L1*sin(f1)*cos(Theta1)+L2*sin(f1+f2)*cos(Theta1+Theta2)+L3*sin(f1+f2+f3)*cos(Theta1+Th
eta2+Theta3);...
L1*sin(Theta1)+L2*sin(f1+f2)*sin(Theta1+Theta2)+L3*sin(f1+f2+f3)*sin(Theta1+Theta2+Thet
a3);...
L1*cos(f1)+L2*cos(f1+f2)+L3*cos(f1+f2+f3)];
 %Calculate the location of the middle two joints
 pointl1 = [L1*sin(f1)*cos(Theta1) ; L1*sin(f1)*sin(Theta1)];
pointl2 = pointl1 + [L2*sin(f1+f2)*cos(Theta1+Theta2);L2*sin(f1+f2)*sin(Theta1+Theta2)];
 %Plot
```

```
if (mod(counter,10)==0) %plots every 10 iterations
axis([-100 200 -100 200 -100 200])
axis square
line([0,pointl1(1)],[0,pointl1(2,1)])
hold on
line([pointl1(1),pointl2(1)],[pointl1(2,1),pointl2(2,1)])
line([pointl2(1),xo(1)],[pointl2(2,1),xo(2,1)])
plot(xo(1),xo(2),'o')
 pause(.1)
 end
 dist = (sqrt((xo(1)-xee(1))^2+(xo(2)-xee(2))^2)); %Distance between
 %current EE location and Final EE location is used to normalize the error
%value (x dot)
%Angular velocity of each angle:
thetadot = ((xee-xo)/dist);                                    % fig()changing Theta for i=0:dt:6
theta1 = thetadot(1);
theta2 = thetadot(2,1);
 theta3 = thetadot(3,1);
 %Updtate each angle based on calculated angular velocity
Theta1 = Theta1 + dt*theta1;
 Theta2 = Theta2 + dt*theta2;
 Theta3 = Theta3 + dt*theta3;
counter = counter +1;
 end
```