**Modeling and Control of 2-DOF Robot Arm**

Nasr M. Ghaleb¹ and Ayman A. Aly¹,²

¹Mechanical Engineering Department, College of Engineering, Taif University, PO Box 888, Taif, Saudi Arabia

²Mechanical Engineering Department, Faculty of Engineering, Assiut University, PO Box 71516, Assiut, Egypt

*Corresponding Author: Nasr M. Ghaleb, Mechanical Engineering Department, College of Engineering, Taif University, PO Box 888, Taif, Saudi Arabia*

**INTRODUCTION**

Robotics is defined practically as the study, design and use of robot systems for manufacturing and generally are used to perform highly repetitive, unsafe, hazardous, and unpleasant tasks. Robotics many different functions that used either in industry and manufacture or in complex, clutter and changing environment such as pick and place, assembly, drilling, welding, machine tool load and unload functions, painting, spraying, etc. or in a delivery in a hospital and Hotels, Discovering the space As a results of these different tasks there are different robot arm configuration such as rectangular, spherical, cylindrical, revolute and prismatic jointed.

A pick and place robot arm is used to ease the process of moving materials and supplying the motion required in the manufacturing processes. The transfer process of the materials is usually being accomplished, using man power and as the transfer process is repeated for a period of time, it can cause injuries to the operator. the robot arm preventing injuries and increasing the efficiency of the work, with reducing the human being errors that cost highly time and martial.

The proportional-integral-derivative (PID) control has simple structure for its three gains. The control performances are acceptable in the most of industrial processes. Most robot manipulators found in industrial operations are controlled by PID algorithms independently at each joint.

There are many control techniques used for controlling the robot arm.

The most familiar control techniques are the PID control, adaptive control, optimal control and robust control. As the final goal is to design and manufacturing real robots, it’s helpful doing the simulation before the investigations with real robots, to enhance the final real robot performance and behavior.

**ROBOT SPECIFICATION AND KINEMATICS**

**Robot Specification**

A two degree of freedom robot arm is described in Figure(1) which consists primarily of two links with the following specifications in OXY coordinates:

\[ L_1 = 1 \text{ m is the length of the first link.} \]
\[ L_2 = 1 \text{ m is the length of the second link.} \]
\[ m_1 = 1 \text{ kg is the mass of the first link.} \]
\[ m_2 = 1 \text{ kg is the link of the second link.} \]
\[ \theta_1 = \text{ the rotation angel of the first link.} \]
\[ \theta_2 = \text{ is the rotation angel of the second link.} \]
Robot Kinematics

Forward Kinematics

The Forward kinematics of a robotic arm is determined a group of parameters called Denavit-Hartenberg (DH) parameters which used for deriving the homogenous transformation matrices between the different frames assigned on the robot arm structure. The DH parameters for a two degree of freedom robotic arm are defined as follows:

Table 1. DH-parameters for the 2-DOF robotic arm

<table>
<thead>
<tr>
<th>Link</th>
<th>( a_i )</th>
<th>( a_i )</th>
<th>( d_i )</th>
<th>( \theta_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( L_1 )</td>
<td>0</td>
<td>0</td>
<td>( \theta_1 )</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>( \theta_2 )</td>
</tr>
</tbody>
</table>

The homogenous transformation matrices for the 2-DOF robotic arm shown in Figure(1) are derived as follows:

\[
^{0}T_1 = \begin{bmatrix}
\cos \theta_1 & -\sin \theta_1 & 0 & L_1 \cos \theta_1 \\
\sin \theta_1 & \cos \theta_1 & 0 & L_1 \sin \theta_1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\( \text{(1)} \)

\[
^{1}T_2 = \begin{bmatrix}
\cos \theta_2 & -\sin \theta_2 & 0 & L_2 \cos \theta_2 \\
\sin \theta_2 & \cos \theta_2 & 0 & L_2 \sin \theta_2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\( \text{(2)} \)

Using the Eq. (1) and (2), the homogenous transformation matrix \(^2T_0\) can be derived as follows:

\[
^{2}T_0 = \begin{bmatrix}
\cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \\
\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\( \text{(3)} \)

Therefore,

\[
^H_T = \begin{bmatrix}
x & \alpha_x & \alpha_x & px \\
y & \alpha_y & \alpha_y & py \\
z & \alpha_z & \alpha_z & pz \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\( \text{(4)} \)

\[
^{0}T_2 = ^H_T
\]

\( \text{(5)} \)

From Eq. (5), the position coordinates of the manipulator end-effector is given by:

\[
P_x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad \text{(6)}
\]

\[
P_y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad \text{(7)}
\]

And the end-effector's orientation matrix is defined by the first three rows and three columns of the transformation matrix in Eq.(3).

Inverse Kinematics

The inverse kinematics of a robotic arm is a solution of finding the robot arm joint variables of given the position Cartesian coordinates of the end-effector. The mathematical equations used to solve the inverse kinematics problem can be derived either algebraically or geometrically. The geometrical approach is considered to be much easier for robot arms of high degrees of freedom. In our Case, we solved the inverse kinematics equations for the 2-DOF robotic arm shown in Figure(2) using the geometrical method.

Figure 2. Two degree of freedom Robot Arm Inverse Kinematic

From Figure2, a mathematical equation for...
Modeling and Control of 2-DOF Robot Arm

solving the elbow joint angle $\theta_2$ can be derived using Pythagoras theorem as follows:

$$p_x^2 + p_y^2 = L_1^2 + L_2^2 + 2L_1L_2\cos \theta_2$$  (8)

$$\cos \theta_2 = \frac{1}{2L_1L_2} (p_x^2 + p_y^2 - L_1^2 - L_2^2)$$  (9)

$$\sin \theta_2 = \pm \sqrt{1 - \cos \theta_2^2}$$  (10)

Therefore,

$$\theta_2 = \pm \arctan \frac{\sin \theta_2}{\cos \theta_2}$$  (11)

For The joint variable $\theta_1$:

$$p_x = (L_1 + L_2\cos \theta_2)\cos \theta_1 - L_2\sin \theta_1 \sin \theta_2$$  (12)

$$p_y = L_2 \sin \theta_2 \cos \theta_1 + (L_1 + L_2\cos \theta_2)\sin \theta_1$$  (13)

$$\Delta = \begin{bmatrix} L_1 + L_2\cos \theta_2 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2\cos \theta_2 \end{bmatrix}$$  (14)

$$p_x^2 + p_y^2 = (L_1 + L_2\cos \theta_2)^2 + (L_2\sin \theta_2)^2$$  (15)

$$\Delta \sin \theta_1 = \begin{bmatrix} L_1 + L_2\cos \theta_2 & p_x \\ L_2 \sin \theta_2 & p_y \end{bmatrix}$$  (16)

$$\Delta \cos \theta_1 = \begin{bmatrix} p_x \\ p_y \\ L_1 + L_2\cos \theta_2 \end{bmatrix}$$  (17)

$$\sin \theta_1 = \frac{\Delta \sin \theta_1}{\Delta} = \frac{(L_1 + L_2\cos \theta_2)p_x - L_2\sin \theta_2 p_x}{p_x^2 + p_y^2}$$  (18)

$$\cos \theta_1 = \frac{\Delta \cos \theta_1}{\Delta} = \frac{(L_1 + L_2\cos \theta_2)p_y + L_2\sin \theta_2 p_y}{p_x^2 + p_y^2}$$  (19)

$$\theta_1 = \arctan \frac{\sin \theta_1}{\cos \theta_1} = \arctan \frac{(L_1 + L_2\cos \theta_2)p_x - L_2\sin \theta_2 p_x}{(L_1 + L_2\cos \theta_2)p_y + L_2\sin \theta_2 p_y}$$  (20)

**ROBOT DYNAMICS**

The dynamic model of a robot is concerned with the movement and the forces involved in the robot arm and establishes a mathematical relationship between the location of the robot joint variables and the dimensional parameters of the robot. There are two methods for performing the dynamics equations of a robot.

After simplification,

$$KE = \frac{1}{2}(m_1 + m_2)l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_1^2 \dot{\theta}_1^2 + m_2 l_2^2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2}m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \cos \theta_2 \left( \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2 \right)$$  (25)

The Potential energy is defined as:

$$PE = m_1gl_1 \cos \theta_1 + m_2gl_1 \cos \theta_1$$  (26)

Substitute Eq.(25) and(26) in Eq.(21) to form the lagrangian equation as:

$$\mathcal{L} = \frac{1}{2}(m_1 + m_2)l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_1^2 \dot{\theta}_1^2 + m_2 l_2^2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2}m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \cos \theta_2 \left( \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2 \right) - m_1 gl_1 \cos \theta_1 - m_2 gl_1 \cos \theta_1$$  (27)

To calculate the force applied to the robot, we form the Lagrange-EularEq.(21)with the Lagrangian(\mathcal{L})

$$F_{\theta_1,2} = \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}_{1,2}} \right) - \frac{\partial \mathcal{L}}{\partial \theta_{1,2}}$$
Modeling and Control of 2-DOF Robot Arm

After simplification, the force applied at joint 1 is given by

\[
F_{\theta_1} = (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos \theta_2) \ddot{\theta}_1 + (m_2l_2^2 - m_2l_1l_2 \cos \theta_2) \ddot{\theta}_2 - m_2l_1l_2 \sin \theta_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2 - (m_1 + m_2)l_1 \sin \theta_1 - m_2l_2 \sin(\theta_1 + \theta_2)) \tag{28}
\]

and the force applied at joint 2 is given by

\[
F_{\theta_2} = (m_2l_2^2 + m_2l_1l_2 \cos \theta_2) \ddot{\theta}_1 + m_2l_2^2 \ddot{\theta}_2 - m_2l_1l_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2l_2 \sin(\theta_1 + \theta_2) \tag{29}
\]

The motion of the system is given by the following form of a nonlinear equation:

\[
F = B(\dot{q}) + C(q, \dot{q}) + g(q) \tag{30}
\]

where,

\[
F = \begin{bmatrix}
F_{\theta_1} \\
F_{\theta_2}
\end{bmatrix}
\]

\[
B(\dot{q}) = \begin{bmatrix}
(m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos \theta_2) & (m_2l_2^2 - m_2l_1l_2 \cos \theta_2) \\
m_2l_2^2 + m_2l_1l_2 \cos \theta_2 & m_2l_2
\end{bmatrix}
\]

\[
C(q, \dot{q}) = \begin{bmatrix}
-m_2l_1l_2 \sin \theta_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\
-m_2l_1l_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2
\end{bmatrix}
\]

\[
g(q) = \begin{bmatrix}
-(m_1 + m_2)l_1 \sin \theta_1 - m_2l_2 \sin(\theta_1 + \theta_2) \\
-m_2l_2 \sin(\theta_1 + \theta_2)
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix}
\]

**Mathematical Modeling of Actuating System**

The Permanent Magnet Direct Current PMDC motor is used to actuate the system, which has an Electrical part and a Mechanical part, as seen in Eq.(44) and Eq.(45). Describing the Electrical and Mechanical characteristics of PMDC motor, respectively. Using Newton's law, Kirchhoff's law and Ohm's law.

\[
[V_m(s) - K_b \omega_m] * \frac{1}{(I_a s + R_a)} = I_a(s) \tag{31}
\]

\[
G_{speed}(s) = \frac{\omega(s)}{V_m(s)} = \frac{s^2 + (R_a J_m + b_m L_a)s + (R_b b_m + K_a K_b)}{I_a}
\]

**PID Controller Design**

The nonlinear equation that are derived from Euler-Lagrange Eq. (30), where input variable F which represents the torque applied to the robot is unknown, so it's requires a control in the Force applied of the joints to reach the final position. In our case, we use the classical linear PID. Particularly we need Two PID controls since the first arm motion is dependent from the second arm motion. In fact, still having a strong interaction between the two arms. The classical linear PID law is performed;

\[
F = K_p e + K_D \dot{e} + K_i \int e \, dt \tag{35}
\]

where \( \omega = \dot{q}^d - q, \dot{q}^d \) is desired joint angle, \( K_p, K_i, K_d \) are proportional, integral and derivative gains of the PID controller, respectively. This PID control law can be expressed via the following equations.

\[
F = K_p e + K_D \dot{e} + \xi \tag{36}
\]

\[
\dot{\xi} = K_i e; \quad \xi(0) = \xi_0
\]

where \( \xi \) represents the additional state variable that is the integral action of the PID control law formed and it's time derivative is \( \dot{\xi} = K_i e \). The closed-loop equation is obtained by substituting the control action F from Eq.(32) into the robot model Eq.(30), gives:
Modeling and Control of 2-DOF Robot Arm

\[ B(\dot{q}) + C(\dot{q}, q) + g(q) = K_F e + K_D \dot{e} + \xi \quad (37) \]

From the system's model Eq.(30), we can have

\[ \ddot{q} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + F \quad (38) \]

with

\[ \ddot{F} = B(q)^{-1} F \Rightarrow F = B(q)\ddot{F} \quad (39) \]

so, we decoupled the system to have the new (non-physical) input

\[ \ddot{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (40) \]

however, the physical torque inputs to the system are

\[ \begin{bmatrix} f_{01} \\ f_{02} \end{bmatrix} = B(q) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (41) \]

The error signals of the system are

\[ e(\theta_1) = \theta_{1f} - \theta_1 \]
\[ e(\theta_2) = \theta_{2f} - \theta_2 \quad (42) \]

where \( \theta_f \) is the final positions. The final position is given by

\[ \begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} \pi/2 \\ -\pi/2 \end{bmatrix} \]

and the initial position is given by

\[ \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + \begin{bmatrix} K_{P1} (\theta_{1f} - \theta_1) + K_{D1} \dot{\theta}_1 + \xi_1 \\ K_{P2} (\theta_{2f} - \theta_2) + K_{D2} \dot{\theta}_2 + \xi_2 \end{bmatrix} \quad (43) \]

\[ \theta_0 = \begin{bmatrix} -\pi/2 \\ \pi/2 \end{bmatrix} \]

so, in our case:

\[ f_1 = K_{P1} (\theta_{1f} - \theta_1) + K_{D1} \dot{\theta}_1 + K_{I1} \int e(\theta_1) \, dt \]
\[ f_2 = K_{P2} (\theta_{2f} - \theta_2) + K_{D2} \dot{\theta}_2 + K_{I2} \int e(\theta_2) \, dt \]

However, the complete system equations with control would be

\[ \ddot{q} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + \ddot{F} \quad (44) \]

recalling the physical actual torques

\[ \begin{bmatrix} f_{01} \\ f_{02} \end{bmatrix} = B(q) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (45) \]

Recalling Eq.(32) and Eq.(38), gives:

\[ \xi_1 = K_{I1} \int e(\theta_1) \, dt \Rightarrow \dot{\xi}_1 = K_{I1} e_1 \]
\[ \xi_2 = K_{I2} \int e(\theta_2) \, dt \Rightarrow \dot{\xi}_2 = K_{I2} e_2 \quad (46) \]

So, the system equations are

\[ \dot{\xi}_1 = K_{I1} e_1 \]
\[ \dot{\xi}_2 = K_{I2} e_2 \]

SIMULATION AND RESULTS

Approximate mathematical models can be obtained and then simulated in combination with the designed control law, for providing a more realistic validation of the system behavior and control performance. First we made a mathematical model of the PMDC motor to determine the transfer function and then start the simulation of the actuator system without controller using MATLAB /Simulink as shown in Figure (3) to study the system performance and then improving the design by adding PID Controller as shown in Figure(4) and (5) for a more realistic validation of the system performance.

The system reaches steady state value of 0.778 rad/s in 1.27 s, with small overshoot and the system response is very fast as shown in Figure(3), so we need to add controller to improve the system behavior. PID controller simulink is built to control the angular speed at a desired set point value (K_{a,t} = 1) and improving the system performance as illustrate in Figure4.
We have noticed from Figure(5) and (6) that the system is second order system (Under damped) and the values of the controller are getting with PID tuning by trial and error with system characteristics values for the best performance and behavior: $K_p=13$, $K_i=39$, $K_d=1.056$, OS (%) < 5, settling time < 2 sec, steady-state error < 1% and $K_{DC} = 1$.

After tuning by trial and error we got with the PID controller values : $K_p=40$, $K_i=13$, $K_d=5$ for the best performance, Also we have noticed from Figure(8) that the position error reaches at a considerable time to zero steady-state of constant input.

The force or torque applied can be shown in Figure(9) is zero at steady state.

Then we have studied the two degree of freedom with PID control as a whole system by trial and error.

The PID parameters are manually tuned to get the best performance of the system. The best behavior and performance of the controller parameters values are as follows:

$K_{p1} = 250$  
$K_{p2} = 250$  
$K_{i1} = 200$  
$K_{i2} = 200$  
$K_{d1} = 30$  
$K_{d2} = 30$

By our approach trial and error, we have noticed that $K_{p1}$ is related to direct error and to speed of evolution, $K_{p1}$ is related to speed of interaction.
with change in states and $K_{P1}$ is related to overall error cancellation.

We have seen in the Figures (10) and (11) that the position error reaches zero at a reasonably fast time.

By the experiment, we got that the slightest change in the controller parameters yields more overshoots and oscillations due to highly sensitive to initial and final positions.

**CONCLUSION**

The main topic of this paper is modeling, simulation and control of two degree of freedom robot arm.

The mathematical models of the actuator and whole system are illustrated. Lagrangian and Euler-Lagrange used to derive a dynamic model that simulated the actual robot movement in real life and obtain sufficient control over the robot joint positions. The system was controlled in order to reach a desired joint angle position through simulation of PID controllers using MATLAB/Simulink.

Also, the result showed that a slight changes in initial joint angle positions of the robot arm resulted in different desired joint angle positions and this necessitated that the gains of the PID controllers need to be adjusted and turned at every instant in order to prevent overshoot and oscillation that associated with the change in parameters values.

**REFERENCES**


Modeling and Control of 2-DOF Robot Arm


Citation: Nasr M. Ghaleb and Ayman A. Aly, "Modeling and Control of 2-DOF Robot Arm", International Journal of Emerging Engineering Research and Technology, 6(10), pp.24-31

Copyright: © 2018 Nasr M. Ghaleb. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.