Abstract—In this paper, we propose a solution to the motion control problem of a 2-link revolute manipulator arm. We require the end-effector of the arm to move safely to its designated target in a priori known workspace cluttered with fixed circular obstacles of arbitrary position and sizes. Firstly a unique velocity algorithm is used to move the end-effector to its target. Secondly, for obstacle avoidance a turning angle is designed, which when incorporated into the control laws ensures that the entire robot arm avoids any number of fixed obstacles along its path enroute the target. The control laws proposed in this paper also ensure that the equilibrium point of the system is asymptotically stable. Computer simulations of the proposed technique are presented.

Keywords—2-link revolute manipulator, motion control, obstacle avoidance, asymptotic stability.

I. INTRODUCTION

A robot manipulator is a mechanism made up of rigid links connected by different joints [1]. The two basic types of joints commonly found in literature are rotational (revolute) or translational (prismatic). A revolute joint is like a hinge that allows relative rotation between two links whereas a prismatic joint provides a linear sliding movement between two links [2].


In this paper, we provide a relatively simple approach to solve the motion control of a 2-link revolute robot arm which is made up of two rotational joints. We first derive the kinematic model of the robot arm and then model its motion from an initial state to the target in the presence of fixed obstacles. The mechanical singularities associated with the system are treated as artificial obstacles and are avoided using the obstacle avoidance scheme. Our main aim is to design a set of continuous control laws that ensure asymptotic stability of the system. A unique and scalable algorithm for target convergence and obstacle avoidance is proposed that works for any number fixed obstacles of various sizes.

The rest of the paper is organized as follows: In Section II, we derive the kinematic model of the revolute arm. The motion planning and control of the planar robot arm in the absence of obstacles is considered in Section III. In Section IV, we propose an obstacle avoidance technique to control the motion of the robot arm in a workspace cluttered with fixed circular obstacles. Computer simulation of the generated path with the proposed technique is also presented. The stability of the system is studied in Section V. Finally, in Section VI concluding remarks on the contributions and future work are made.

II. MODELLING THE REVOLUTE ARM

We have considered a simple 2-link revolute manipulator arm that has two rotational joints in the \( z_1z_2 \) - plane as shown in the Fig. 1. The arm consists of two rigid links which are connect via revolute joints; the first link can rotate through 360° and the second link which carries the payload at the gripper can rotate through 180° with respect to the first link.

With the help of Fig. 1, we assume that:

i. the 2-link revolute manipulator arm is anchored at the origin;

ii. link 1 has a length of \( l_1 \) and has an angular position \( \theta_1(t) \), measured counterclockwise from the \( z_1 \)-axis at time \( t \);

iii. link 2 has a length of \( l_2 \) and has an angular position \( \theta_2(t) \), measured counterclockwise from link 1 at time \( t \);

iv. the coordinate of the end-effector is \((x,y)\).
Remark: We can express the position of the end-effector completely in terms of the state variables, \( \theta_1(t) \) and \( \theta_2(t) \) as

\[
x = \ell_1 \cos \theta_1 - \ell_2 \cos(\theta_1 + \theta_2),
\]

\[
y = \ell_1 \sin \theta_1 - \ell_2 \sin(\theta_1 + \theta_2).
\]

We first look at the motion of the end-effector of the robot arm. Let a velocity of \( v \) be applied to the end-effector. Then the kinematic equations of the end-effector can be expressed as

\[
\dot{x} = u_1,
\]

\[
\dot{y} = u_2,
\]

where \( u_1 \) and \( u_2 \) are the \( z_1 \) and \( z_2 \) components, respectively, of \( v \). We now look for the kinematic model of this 2-link manipulator arm. Let \( d = \sqrt{x^2 + y^2} \) be the distance from the origin to the end-effector, then using the cosine rule, we see that

\[
d^2 = \ell_1^2 + \ell_2^2 - 2\ell_1 \ell_2 \cos \theta_2.
\]

Differentiate this with respect to \( t \) and simplify, we see that

\[
\dot{\theta}_2 = \frac{\ell_1 \cos \theta_1 - \ell_2 \cos(\theta_1 + \theta_2)}{\ell_1 \ell_2 \sin \theta_2} u_1
\]

\[
+ \frac{\ell_1 \sin \theta_1 - \ell_2 \sin(\theta_1 + \theta_2)}{\ell_1 \ell_2 \sin \theta_2} u_2.
\]

Next, we look at the kinematic equation relating to \( \theta_2(t) \). Let \( \vartheta \) be the angular position of the end-effector relative to the origin, then

\[
\tan \vartheta = \frac{y}{x}
\]

from which we get

\[
\dot{\vartheta} = \frac{u_2 \cos \vartheta - u_1 \sin \vartheta}{d}.
\]

Using the cosine rule again, we see that

\[
\ell_2^2 = \ell_1^2 + d^2 - 2\ell_1 d \cos(\theta_1 - \vartheta).
\]

Differentiate this with respect to \( t \) and simplify, we get

\[
\dot{\theta}_1 = \frac{u_2 \cos \vartheta - u_1 \sin \vartheta}{d} \left( \ell_1 \cos(\theta_1 - \vartheta) - d \right)
\]

\[
+ \left( u_1 \cos \vartheta + u_2 \sin \vartheta \right) \left( \ell_1 \cos(\theta_1 - \vartheta) - d \right)
\]

\[
\ell_1 \sin(\theta_1 + \theta_2) u_1
\]

\[
+ \ell_1 \sin(\theta_1 + \theta_2) u_2.
\]

Thus the kinematic equations for the revolute manipulator arm is

\[
\begin{align*}
\dot{x} &= u_1, \\
\dot{y} &= u_2, \\
\dot{\theta}_1 &= \frac{u_2 \cos \vartheta - u_1 \sin \vartheta}{d} \left( \ell_1 \cos(\theta_1 - \vartheta) - d \right) \\
&+ \left( u_1 \cos \vartheta + u_2 \sin \vartheta \right) \left( \ell_1 \cos(\theta_1 - \vartheta) - d \right) \\
&+ \ell_1 \sin(\theta_1 + \theta_2) u_1, \\
\dot{\theta}_2 &= \frac{\ell_1 \cos \theta_1 - \ell_2 \cos(\theta_1 + \theta_2)}{\ell_1 \ell_2 \sin \theta_2} u_1
\end{align*}
\]  

(1)

system (1) is a description of the instantaneous angular velocities of the revolute manipulator arm. Here \( u_1 \) and \( u_2 \) are classified as the controllers. We shall use the vector notation \( \mathbf{x}(t) = (\theta_1(t), \theta_2(t)) \) to refer to the angular position of the robot in the \( z_1z_2 \)-plane.

III. MOTION CONTROL IN THE ABSENCE OF OBSTACLES

In our motion control problem, we want the end-effector of the robot arm to start from an initial position, move towards its target and converge to the center of the target. The target, \( T \), considered in this paper is a disk of center \( (p_1, p_2) \) and radius \( r_T \) which is described as:
\[ T = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - p_1)^2 + (z_2 - p_2)^2 \leq r_2^2\}. \]

**A. Target Convergence**

For the end-effector of the robot arm to move from its initial position to the target position, we adopt the following form of velocity algorithm from [7] which is depended on the initial and final positions of the robot:

\[
v = |v_0| \left\| \left( \frac{p_1 - x(0), p_2 - y(0)}{p_1 - x(0), p_2 - y(0)} \right) \right\|
\]

where \(|v_0|\) is the initial velocity of the end-effector at \( t = 0 \). The function \( v \) is defined, continuous and positive over the domain

\[ D = \{ x \in \mathbb{R}^2 : (x(0), y(0) \neq (p_1, p_2)) \}. \]

Now if we define

\[
x_\ast = \text{atan2}(p_2, p_1) + \cos^{-1}\left[ \frac{p_1^2 + p_2^2 + \ell^2_1 - \ell^2_2}{2 \ell_1 \sqrt{p_1^2 + p_2^2}} \right] + \cos^{-1}\left[ \frac{\ell^2_2 + \ell^2_1 - p_1^2 - p_2^2}{2 \ell_1 \ell_2} \right] \in D,
\]

then we see that \( x_\ast \) is an equilibrium point of system (1).

For \( x(t) \neq x_\ast \), we further define \( \xi(t) \) as the angular position of the target center relative to the end-effector at time \( t \). The angle \( \xi(t) \) is defined implicitly

\[
\tan \xi(t) = \begin{cases} \frac{p_2 - \ell_1 \sin \theta_1 + \ell_2 \sin (\theta_1 + \theta_2)}{p_1 - \ell_1 \cos \theta_1 + \ell_2 \cos (\theta_1 + \theta_2)} & \text{if } x(t) \neq x_\ast, \\ \tan \xi(t - 1), & \text{if } x(t) = x_\ast. \end{cases}
\]

**B. Mechanical Singularities**

In reality, the motion of the end-effector is restricted in the sense that the second link of the revolute 2-link manipulator can neither be fully stretched nor can it be folded back [5]. That is the angle \( \theta_2 \) is restricted as

\[ 0 \leq |\theta_2| \leq \pi \]

In order to observe this restriction, we treat the line passing through the points \((0,0)\) and \((\ell_1 \cos \theta_2, \ell_2 \sin \theta_2)\) as an artificial obstacle for the end-effector. The end-effector can avoid this line by simply avoiding the closest point on the line [12]. The closest point on the line measured form the position of the end-effector is given by \((x', y', \ell_1 \cos \theta_2, \ell_2 \sin \theta_2)\) where

\[ x' = 1 - \frac{\ell_2 \cos \theta_2}{\ell_1} \]

This closest point on the line is treated as an artificial obstacle and thus it will simply be avoided by carefully defining the controllers.

The distance from the end-effector to the closest point on the line is given by \( R_o = \ell_2 \sin \theta_2 \). Let \( d_{\text{max}} > 0 \) be a pre-determined scalar and define

\[
f_o = \begin{cases} \ell_1 \cos \theta_1 - \ell_2 \cos (\theta_1 + \theta_2) & \text{if } R_o \geq d_{\text{max}} \\ -\ell_1 \sin \theta_1 + \ell_2 \sin (\theta_1 + \theta_2) & \text{if } R_o < d_{\text{max}} \end{cases}
\]

\[
\alpha_o = \begin{cases} 0, & \text{if } f_o \geq 0 \\ -1, & \text{if } f_o > 0 \end{cases}
\]

\[
\beta_o = 	an^{-1} \left( \frac{\alpha_o \beta_o}{R_o} \right)
\]

For the avoidance of the line obstacle, we propose the following form of the controllers \( u_1 \) and \( u_2 \):

\[
u_1 = v \cos (\xi + \epsilon_o), \quad u_2 = v \sin (\xi + \epsilon_o)
\]

**Remark:** With the form of controllers given in (2), we see that as the end-effector comes closer to the line obstacle, then the quantity \( R_o \) will decrease. This will increase \(|\epsilon_o|\) since \( R_o \) appears in the denominator. Hence an increase in \(|\epsilon_o|\) will force the end-effector to move away from the obstacle.

Substituting for \( v \) and \( \xi \) into (2), we obtain

\[
u_1 = |v_0| \left\| \left( \frac{(p_1 - x)R_o - (p_2 - y)\alpha_o \beta_o}{\sqrt{\alpha^2_0 + R_o^2}} \right) \right\|,
\]

\[
u_2 = |v_0| \left\| \left( \frac{(p_1 - x)R_o - (p_2 - y)\alpha_o \beta_o}{\sqrt{\alpha^2_0 + R_o^2}} \right) \right\|
\]

The controllers are bounded and continuous at every point over the domain

\[ D = \{ x \in \mathbb{R}^2 : (x(0), y(0) \neq (p_1, p_2)) \cap R_o > 0 \}. \]

**Simulation 1:** The computer is used to numerically integrate system (1) to obtain the solution \((r(t), \theta(t))\) and plot the trajectory of the end-effector, which converges to the target position \((p_1, p_2)\) and stays there as \( t \to +\infty \). For our simulation, Table 1 gives the values of the different parameters, and Fig. 2 gives the trajectory of the arm. It was noticed that due to the unique forms of the controllers, the arm slowed down as the end-effector approached its target.
TABLE I
VALUES OF DIFFERENT PARAMETERS USED IN THE SIMULATION

<table>
<thead>
<tr>
<th></th>
<th>Initial Configuration</th>
<th>Final Configuration</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial position</td>
<td>(6,7) m</td>
<td>(8,-6) m</td>
<td></td>
</tr>
<tr>
<td>Initial velocity</td>
<td>1 m/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target position</td>
<td>(8,-6) m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radius of the target</td>
<td>0.2 m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workspace dimensions</td>
<td>0 ≤ z₁ ≤ 10, 0 ≤ z₂ ≤ 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robot dimensions</td>
<td>l₁ = 6 m, l₂ = 6 m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensing region</td>
<td>dₘₐₓ = 1 m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Remark:** Assumption 1 is justified since link 1 needs to move freely in the circular region

\[ C = \{(z₁,z₂) \in \mathbb{R}^2 : z₁^2 + z₂^2 ≤ l₁^2\} \]

This also ensures that link 1 is not trapped in between two obstacles.

Thus for the end-effector to converge to its target safely, we need the entire link 2 to avoid a fixed obstacle. For this, we utilize the minimum distance technique (MDT) proposed by Sharma in [12], where the author calculated the minimum distance from a robot to a line segment and the resultant closest point was avoided by the robot. However, in our case, we want the line segment (link 2) to avoid a fixed obstacle.

Adopting the concept from [12], we find the point on link 2 that will be closest to a fixed obstacle. Let \((x', y')\) be a point on link 2 that is closest to the \(l\)th fixed obstacle, then it can be shown that

\[ x' = \ell₁ \cos \theta₁ + (\lambda₁ - 1) \ell₂ \cos (\theta₁ + \theta₂), \]
\[ y' = \ell₁ \sin \theta₁ + (\lambda₁ - 1) \ell₂ \sin (\theta₁ + \theta₂), \]

where

\[ \lambda₁ = 1 + \frac{1}{\ell₂} \left[ (o₁ - l₁ \cos \theta₁) \ell₂ \cos (\theta₁ + \theta₂) \right. \]
\[ + \left. (o₂ - l₁ \sin \theta₁) \ell₂ \sin (\theta₁ + \theta₂) \right]. \]

Note that \(\lambda₁ \in [0, 1]\). If \(\lambda₁ > 1\), then we let \(\lambda₁ = 1\) and if \(\lambda₁ < 0\), then we let \(\lambda₁ = 0\). Otherwise we accept the value of \(\lambda₁\) between 0 and 1.

Now, let \(R = \sqrt{(x' - o₁)^2 + (y' - o₂)^2 - rₖ}\) be the distance from the center of the \(l\)th obstacle to the point \((x', y')\) and define

\[ f₁ = \left[ \ell₁ \sin \theta₁ - \ell₂ \sin (\theta₁ + \theta₂) \right] p₁ \]
\[ - \left[ \ell₁ \cos \theta₁ - \ell₂ \cos (\theta₁ + \theta₂) \right] p₂ \]
\[ \alpha_l = \begin{cases} 0, & \text{if } R_l ≥ dₘₐₓ \\ dₘₐₓ - R_l, & \text{if } R_l < dₘₐₓ \end{cases} \]
\[ \beta_l = \begin{cases} 1, & \text{if } f_l ≤ 0 \\ -1, & \text{if } f_l > 0 \end{cases} \]
\[ eₙ = \frac{\alpha_l \beta_l}{R_l} \]

for \(l = 1,2,...,q\). In order for the point \((x', y')\) to avoid the fixed obstacles and for the end-effector to avoid the artificial obstacle, we modify the controllers \(u₁\) and \(u₂\) given in (2) as:
\[ u_i = v \cos \left( \xi + \tan^{-1} \left( \sum_{l=1}^{q} \frac{e_l}{R_i} \right) \right), \]
\[ u_2 = v \sin \left( \xi + \tan^{-1} \left( \sum_{l=1}^{q} \frac{e_l}{R_i} \right) \right). \]

Substituting for \( v \) and \( \xi \) into (4), we obtain

\[ u_i = \left. \frac{v_0 \left[ (p_1 - x) - (p_2 - y) \sum_{l=1}^{q} \frac{\alpha_l \beta_l}{R_l} \right]}{\| (p_1 - x(0), p_2 - y(0)) \| \left[ 1 + \left( \sum_{l=1}^{q} \frac{\alpha_l \beta_l}{R_l} \right)^2 \right]^{1/2}} \right| \]
\[ u_2 = \left. \frac{v_0 \left[ (p_2 - y) - (p_1 - x) \sum_{l=1}^{q} \frac{\alpha_l \beta_l}{R_l} \right]}{\| (p_1 - x(0), p_2 - y(0)) \| \left[ 1 + \left( \sum_{l=1}^{q} \frac{\alpha_l \beta_l}{R_l} \right)^2 \right]^{1/2}} \right| \]

The controllers are bounded and continuous at every point over the domain

\[ D = \{ x \in \mathbb{R}^2 : (x(0), y(0)) \neq (p_1, p_2) \cap R_l > 0 \text{ for } l = 1, 2, \ldots, q \} \]

**Simulation 2:** To illustrate the effectiveness of our proposed controller, we have generated the trajectories of the revolute arm from some initial configuration to the final configuration. This is shown in Example 1 and Example 2 below.

**Example 1:** The robot arm encounters a fixed obstacle which it has to avoid along its journey to the target. The values of the different parameters used in the simulation (if different from Simulation 1) are given in Table II.

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>VALUES OF DIFFERENT PARAMETERS USED IN THE SIMULATION.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>Initial position</td>
<td>(8, -5) m</td>
</tr>
<tr>
<td>Initial velocity</td>
<td>1 m/s</td>
</tr>
<tr>
<td><strong>Final Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>Target position</td>
<td>(5, 8) m</td>
</tr>
<tr>
<td>Radius of the target</td>
<td>0.2 m</td>
</tr>
<tr>
<td><strong>Obstacle Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Obstacle center</td>
<td>(6, 4) m</td>
</tr>
<tr>
<td>Obstacle radius</td>
<td>0.8 m</td>
</tr>
</tbody>
</table>

Fig. 3 shows the trajectory of the robot arm from the initial to the final states. Note that, given appropriate initial conditions, the robot arm avoided the obstacle and the end-effector converged to its designated target.

**Example 2:** Here, we illustrate the effectiveness of the control laws among several fixed obstacles. The initial and final position of the end-effector is shown in Fig. 4 while the positions and sizes of the fixed obstacles were generated randomly.

Fig. 4 shows the convergence of the end-effector to its designated target in the presence of multiple obstacles in the workspace. The robot arm converges nicely to the target whilst avoiding the obstacles along its path.
V. STABILITY ANALYSIS

The controllers \(u_1\) and \(u_2\) defined by (5) are bounded and continuous at every point on the neighborhood of the equilibrium point, and lie in neighborhood of the equilibrium point for all \(t \geq 0\). Given this, the discussions above yield the stability of system (1):

**Theorem 1:** If the initial position \((x(0), y(0))\) of the end-effector does not intersect with the target position and the Assumption 1 holds, then the point \(x_e\) is an asymptotic stable equilibrium point of system (1).

**Proof.** Consider the Lyapunov function

\[
L(x) = \frac{1}{2} \left[ (p_1 - x, p_2 - y) \right]^2
\]

which is defined, continuous and positive over the domain

\[
D = \{ x \in \mathbb{R}^2 : (x(0), y(0) \neq (p_1, p_2) \land R_i > 0 \text{ for } i = 1, 2, \ldots, q \}
\]

It is clear that \(L(x)\) has continuous first partial derivatives in the neighborhood \(D\) of the equilibrium point \(x_e\) of system (1). Moreover, in the region \(D\), we see that \(L(x_e) = 0\) and \(L(x) > 0\) for all \(x \neq x_e\). Now, the time-derivative of \(L(x)\) along a trajectory of system (1) is given by

\[
\dot{L}(x) = -\left[ \frac{(p_1 - x, p_2 - y)}{\sqrt{1 + \left( \sum_{\alpha=0}^{n} \alpha_{\alpha} \right)^2}} \right]
\]

again, it is clear that in the region \(D\), \(\dot{L}(x_e) = 0\) and \(\dot{L}(x) < 0\) for all \(x \neq x_e\). Hence it can be concluded that the point \(x_e\) is an asymptotic stable equilibrium point of system (1).

VI. CONCLUSION

The paper presents a simple approach for solving the motion control of a 2-link revolute manipulator. A tailored target convergence and obstacle avoidance scheme is developed and the control laws are designed to move the end-effector towards its goal and avoid any fixed obstacles along its path.

The control laws proposed in this paper also ensure an asymptotic stability of the system. This has been proven using the Direct Method of Lyapunov. The stabilization property of the system has also been verified numerically via the computer simulations.

Future work will consider the motion control of 3-dimensional manipulator arms, mobile car-like robots and mobile manipulator arms.

REFERENCES