Extended Vision Robotic Arm

A systematic analysis for a RRPR robotic arm

Gardner, Gabriel
University of California, Riverside
gabriel.gardner@email.ucr.edu

Tejeda Ocampo, Andrea
University of California, Riverside
andrea.tejedaocampo@email.ucr.edu

***Abstract*—The purpose of this paper is to show the detailed analysis of a vision extending robotic arm in an RRPR configuration. The Forward Kinematics, Inverse Kinematics, Velocity Kinematics, Statics, and Dynamics are all derived through a combination of hand calculations and programmable algorithms created in MATLAB. All but the Dynamics formulations were successfully derived and example values are illustrated for numerical results. The code to reproduce these experiments is available at: https://github.com/DinoDany/Extended_vision_arm**

I. Introduction

In the last decade, cameras have had a key role in robotics development. Several systems use cameras as their main sensor, relying on computer vision to acquire data and feedback signals. A common issue exploratory wheeled robots can experience is a restriction of visual information, especially on confined spaces such as caves. If the vision sensor of a given robot is statically mounted then there is not much that can be done to extend the visual range of the robot. This paper proposes the use of a robotic arm whose purpose is to extend the visual range of a mobile robotic platform where vision would otherwise become obstructed. This added visual capability would prove crucial in rescue scenarios where a scouting robot needs to feed back to human operators as much information as possible about an emergency situation, or in an exploratory environment where researchers are trying to study creatures or objects that are otherwise not easily viewable.We achieve this capability with an RRPR robotic arm configuration with a 360° camera. What follows is the systematic analysis of the Forward Kinematics, Inverse Kinematics, Velocity Kinematics, Statics, and Dynamics of the system.

II. Methods

A. Forward Kinematics

Forward Kinematics (FK) is defined as the process of calculating the position and velocity of the end-effector based on given joint angles and angular velocities [1]. For the purpose of this application, forward kinematics is useful when a specific value of a joint is needed given the constraints in the terrain or workspace. We will explore two main methods to calculate the FK: Product of Exponentials (PoE) and Denavit-Hartenberg (DH) Method.

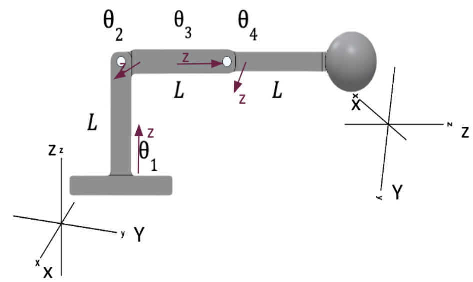

Fig. 1. Home configuration of the RRPR Robotic Arm

1) Product of Exponentials: This method is directly derived from the exponential coordinate representation for rigid-body motions. The fundamental idea behind PoE involves considering each joint as imparting a screw motion to all connected links extending outward in a home-configuration. Fig. 1 ilustrates the zero-configuration of the robotic arm.

We can derive the matrix M to be the position of the last frame in relation to the origin frame when all the joint angles are set to zero

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2L \\ 0 & -1 & 0 & L \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The next step is to calculate the values of the screw axis along each one of the joints. Fig illustrates the results. In this case, we used the fixed-frame formulation.

i	W_i	q_i	S_i
1	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$
2	$(1,0,0)$	$(0,0,L)$	$(0,L,0)$
3	$(0,0,0)$	$-$	$(0,1,0)$
4	$(1,0,0)$	$(0,L,L)$	$(0,L,-L)$

Fig illustrating the values of the screw axis of all the joints in the system

Once the S_i is calculated, you can derived the matrix of the exponential using Rodriguez Formula. Given θ, you can evaluate the product of exponential formula in the space frame (2). The result will be a transformation matrix T_{sb} of the end frame in terms of the origin frame. Refer to the appendix for the result of the T_{sb} matrix.

$$T(\theta) = e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_4]\theta_4} M \qquad (2)$$

2) Denavit-Hartenberg Method : This method requires reference frames to be attached to each link following a set of specific rules of assignment:

The first step is to set the $\hat{z}_i$-axis in the same direction as the joint axis. Then, find a segment that orthogonally intersects the $\hat{z}_i-$axis and $\hat{z}_{i-1}$-axis. The $\hat{x}_i$-axis will be defined along that segment. Now you can define $\hat{y}_i$-axis following the right-hand rule. If $\hat{z}_i$-axis and $\hat{z}_{i-1}$-axis intersect, then the diagonal segment does not exist and your $\hat{x}_i$-axis will be defined as a perpendicular axis along the span of $\hat{z}_i$ and $\hat{z}_{i-1}$-axis. Repeat the process for all the joints in the system.

The next step is to define the parameters. The length of the perpendicular line joining the $\hat{z}_i$-axis, is denoted by the scalar a_{i-1} and will be our first DH parameter. The link twist α_{i-1} is an angle measured from $\hat{z}_{i-1}$ to $\hat{z}_i$ about the $\hat{x}_{i-1}$-axis. The link offset d_i is the distance between the intersection of $\hat{x}_{i-1}$ and $\hat{z}_i$ to the origin of the link-i frame. Lastly, the joint angle ϕ_i is measured from $\hat{x}_{i-1}$ to $\hat{x}_i$ along the $\hat{z}_i$-axis.

Following the rules, we were able to find the DH parameters for each one of the joints in the robotic arm. Figure , shows the results of the system. To calculate the transformation matrix for each link frame, we used the following formula:

$$\mathcal{T}_{i-1,i} = Rot(\hat{x}_i, \alpha_{i-1}) Trans(\hat{x}_i, a_{i-1}) Trans(\hat{z}, d_i) Rot(\hat{z}, \phi_i) \quad (3)$$

Once the transformation matrices are computed, the forward kinematics is obtained by multiplying each one of these link frame transformations. The result will be matrix T_{sb}, a transformation matrix of the end-frame in terms of the origin frame. The final position of the end-effector is computer by the multiplications of the matrix T_{sb} and a initial point P_o at $(0, 0, 0, 1)$ in its homogeneous configuration. (4)

$$\mathcal{P} = T_{sb} * P_o \qquad (4)$$

B. Inverse Kinematics

1) : Inverse Kinematics (IK) solves the problem of determining the set of joint positions that will achieve a specific configuration in the end-effector. For our particular application, IK is useful to guide position the camera at the tip of the arm in a specific angle, focus or move it to avoid collisions. We decided to use a geometric approach to calculate the value of each joint when a particular position of the end.effector is given. Figure [] shows the basic formulation of the geometric shapes used to compute the values. We used the law of cosines to calculate the angles given the measurements of the legs. We defined θ_3 heuristically, a practical application will require some intuition to establish an initial value of θ_3 and iterate in order to fine-tune the optimal angle variable. We defined an equation for each one of the angles in terms of the x, y and z coordinates of the end effector. Refer to the appendix for additional detail on the formula derivation.

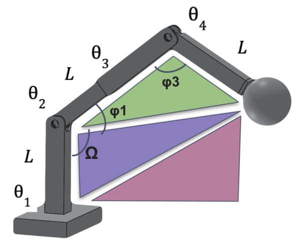

Fig. 2. Trigonometry used to derive the Inverse Kinematics equations

$$\theta_1 = atan2(y, -x) \qquad (5)$$

$$\theta_2 = \Omega(x, y, z) + \phi_1(x, y, z, \theta_3) - \frac{\pi}{2}) \qquad (6)$$

$$\theta_3 = \textit{defined heuristically} \qquad (7)$$

$$\theta_4 = \phi_3(x, y, z, \theta_3) - \pi \qquad (8)$$

C. Velocity Kinematics

1) Task Purpose: When the arm is in operation, we want to be able to record non-blurry videos and images at all times to ensure nothing is missed. This means avoiding moving the camera too fast. This can be accomplished through the velocity kinematics of the end-effector, which relates the end-effector's twist with the linear and angular velocities of each joint [2]. In practice, we would want the inverse velocity kinematics so that we can simply tell the arm the desired twist and have it determine the required joint velocities to comply with the set constraints. However, this falls outside of the scope of this report and as a result has been left out.

2) Derivation of Velocity Kinematics: The velocity kinematics of the system was solved using the Jacobian formulation with respect to the base frame {s} as seen in the following formulas [2].

$$\mathcal{V}_S = J_S(\theta) * \dot{\theta} \qquad (9)$$

where

$$\mathcal{V}_S = \begin{pmatrix} \overrightarrow{\omega_s} \\ \overrightarrow{v_s} \end{pmatrix} \qquad (10)$$

$$J_S(\theta) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & L & 1 & L + \sin(\theta_2)\ (L + \theta_3) \\ 0 & 0 & 0 & -\theta_3\ \cos(\theta_1) \cos(\theta_2) \end{pmatrix} \qquad (11)$$

$$\dot{\theta} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{pmatrix} \qquad (12)$$

Equation (11) illustrates our base-frame Jacobian matrix whose columns are the screw axes of each joint with all prior joint variables $\theta_i \neq 0$. Solving for (9) yields the following result for our end-effector twist:

$$\mathcal{V}_S = \begin{pmatrix} \dot{\theta}_2 + \dot{\theta}_4 \\ 0 \\ \dot{\theta}_1 \\ 0 \\ \dot{\theta}_3 + L\,\dot{\theta}_2 + \dot{\theta}_4\,(L + \sin(\theta_2)\ (L + \theta_3)) \\ -\theta_3\,\dot{\theta}_4\,\cos(\theta_1)\,\cos(\theta_2) \end{pmatrix} \tag{13}$$

Equation (13) illustrates two key aspects of our robotic arm. One, the arm cannot generate any angular velocity in the $\hat{y}$ direction. Two, it cannot generate any linear velocity in the $\hat{x}$ direction.

D. Statics

1) Task Purpose: Once we have found a desirable location to place our camera, we need the arm to keep the camera steady for a potentially long duration of time. For this purpose, we need to determine the statics of our arm in order to determine the required joint torques needed to hold the arm in equilibrium.

2) Derivation of Statics: We make use of the same Jacobian matrix we derived in (11) in order to determine the joint torques of our system [2].

$$\tau_S = J_S^T(\theta) * \mathcal{F}_S \tag{14}$$

where

$$\tau_S = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} \tag{15}$$

$$\mathcal{F}_S = \begin{pmatrix} f_S \\ m_S \end{pmatrix} \tag{16}$$

$$\mathcal{F}_S = \begin{pmatrix} 0 \\ 0 \\ -\mathrm{mg} \\ L\ \cos(\theta_2 + \theta_4)\ \cos(\theta_1) + \mathrm{mg}\ \cos(\theta_1)\ \cos(\theta_2)\ (L + \theta_3) \\ L\ \cos(\theta_2 + \theta_4)\ \sin(\theta_1) - \mathrm{mg}\ \cos(\theta_2)\ \sin(\theta_1)\ (L + \theta_3) \\ 0 \end{pmatrix} \tag{17}$$

Here τ_s is the matrix of joint torques, $J_S^T(\theta)$ is the transpose of our base-frame Jacobian matrix, and $\mathcal{F}_S$ is the expected wrench of the system.

$$\tau_S = \begin{pmatrix} -\mathrm{mg} \\ L\,\sigma_1 \\ \sigma_1 \\ (L + \sin(\theta_2)\ (L + \theta_3))\,\sigma_1 \end{pmatrix} \tag{18}$$

where

$$\sigma_1 = L\,\cos(\theta_2 + \theta_4)\,\sin(\theta_1) - \mathrm{mg}\,\cos(\theta_2)\,\sin(\theta_1)\,(L + \theta_3) \tag{19}$$

Equation (18) illustrates our final joint torque equations. With this, we can determine the torque each joint can expect depending on the joint angles at any given time. It is possible to iterate over all ranges of all four joint angles in order to determine the max torque any one joint might experience during operation. This would be crucial in determining the necessary power of each joint actuator. This, however, is outside of the scope of this paper and has not been conducted.

E. Dynamics

1) Task Purpose: For the same reason we care about velocity kinematics, we also care about dynamics because we want to minimize the acceleration of any given joint to help ensure smooth and non-blurry videos and images are recorded. This is done by determining the equations of motion for each discrete mass that the arm is composed of so that we can control their motions given some joint position, velocity, and acceleration.

2) Derivation of Dynamics: The dynamics of the system were produced using the Lagrangian dynamics formulation which is represented through the kinetic and potential energies of the system as follows [2].

$$\mathcal{L}_i(\theta, \dot{\theta}) = \sum_{i=1}^{n} \mathcal{K}_i(\theta, \dot{\theta}) - \mathcal{P}_i(\theta) \tag{20}$$

Here, we generate n equations for each discrete mass of the system and choosing to locate the mass at the furthest end point of a given link. Once the Lagrangian formulation is derived for each mass, we can then generate the equations of motion for each mass using the following [2]:

$$\tau_i = \frac{\partial}{\partial t}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i} \tag{21}$$

Solving for (21), we generate the following equations of motion for our system:

$$\tau_1 = m\,\ddot{\theta}_1\,(2\,L^2\,\sigma_1 + 4\,L\,\theta_3 + 2\,L^2\,\cos(2\,\theta_2)...$$
$$+\,2\,{\theta_3}^2\,\cos(2\,\theta_2) + L^2\,\cos(2\,\theta_2 + 2\,\theta_4) + 3\,L^2 + 2\,{\theta_3}^2...$$
$$+\,2\,L^2\,\cos(\theta_4) + 2\,L\,\theta_3\,\sigma_1 + 4\,L\,\theta_3\,\cos(2\,\theta_2) + 2\,L\,\theta_3\,\cos(\theta_4))$$

$$\tau_2 = m\,(6\,L^2\,\ddot{\theta}_2 + 2\,L^2\,\ddot{\theta}_4 + 4\,{\theta_3}^2\,\ddot{\theta}_2 - 2\,L\,\ddot{\theta}_3\,\sin(\theta_4)...$$
$$+\,4\,g\,\theta_3\,\cos(\theta_2) + 2\,L^2\,\dot{\theta}_1^2\,\sin(2\,\theta_2) + 2\,{\theta_3}^2\,\dot{\theta}_1^2\,\sin(2\,\theta_2)...$$
$$+\,4\,L^2\,\ddot{\theta}_2\,\cos(\theta_4) + 2\,L^2\,\ddot{\theta}_4\,\cos(\theta_4) + L^2\,\dot{\theta}_1^2\,\sin(2\,\theta_2 + 2\,\theta_4)...$$
$$+\,8\,L\,\theta_3\,\ddot{\theta}_2 + 2\,L\,g\,\cos(\theta_2 + \theta_4) + 4\,L\,g\,\cos(\theta_2)...$$
$$+\,2\,L^2\,\dot{\theta}_1^2\,\sin(2\,\theta_2 + \,\theta_4) + 2\,L\,\theta_3\,\dot{\theta}_1^2\,\sin(2\,\theta_2 + \,\theta_4)...$$
$$+\,4\,L\,\theta_3\,\dot{\theta}_1^2\,\sin(2\,\theta_2) + 4\,L\,\theta_3\,\ddot{\theta}_2\,\cos(\theta_4) + 2\,L\,\theta_3\,\ddot{\theta}_4\,\cos(\theta_4))$$

$$\tau_3 = -m\,(2\,L\,\dot{\theta}_2^2 - 2\,\ddot{\theta}_3 + 2\,\theta_3\,\dot{\theta}_2^2 - 2\,g\,\sin(\theta_2)...$$
$$+\,L\,\ddot{\theta}_2\,\sin(\theta_4) + L\,\ddot{\theta}_4\,\sin(\theta_4) + L\,\dot{\theta}_2^2\,\cos(\theta_4)...$$
$$+\,2\,L\,\dot{\theta}_1^2\cos(\theta_2)^2 + 2\,\theta_3\,\dot{\theta}_1^2\cos(\theta_2)^2 + L\,\dot{\theta}_2\,\dot{\theta}_4\,\cos(\theta_4)...$$
$$+\,L\,\dot{\theta}_1^2\cos(\theta_2)^2\,\cos(\theta_4) - L\,\dot{\theta}_1^2\,\cos(\theta_2)\,\sin(\theta_2)\,\sin(\theta_4))$$

$$\tau_4 = L\,m\,(2\,L\,\ddot{\theta}_2 + 2\,L\,\ddot{\theta}_4 + 2\,g\,\cos(\theta_2 + \theta_4) - 2\,\ddot{\theta}_3\,\sin(\theta_4)...$$
$$+\,2\,\theta_3\,\ddot{\theta}_2\,\cos(\theta_4) + 2\,\dot{\theta}_2\,\dot{\theta}_3\,\cos(\theta_4) + 2\,\dot{\theta}_3\,\dot{\theta}_4\,\cos(\theta_4)...$$

$+ L\,\dot{\theta}_1^2\,\sin(\theta_4) + 2\,L\,\dot{\theta}_2^2\,\sin(\theta_4) + \theta_3\,\dot{\theta}_1^2\,\sin(\theta_4)...$
$+2\,\theta_3\,\dot{\theta}_2^2\,\sin(\theta_4)+L\,\dot{\theta}_1^2\,\sin(2\,\theta_2+\theta_4)+\theta_3\,\dot{\theta}_1^2\,\sin(2\,\theta_2+\theta_4)...$
$+ 2\,L\,\ddot{\theta}_2\,\cos(\theta_4) + L\,\dot{\theta}_1^2\,\sin(2\,\theta_2 + 2\,\theta_4)...$
$+ 2\,L\,\dot{\theta}_2\,\dot{\theta}_4\,\sin(\theta_4) + 2\,\theta_3\,\dot{\theta}_2\,\dot{\theta}_4\,\sin(\theta_4))(22)$

It is important to note that the result illustrated in (22) is faulty in that it claims to represent more links than the arm truly contains. This is due to a fundamental misunderstanding of how to derive the dynamics equations when a system contains two distinct joints on top of each other. While we did not have the time to properly correct this mistake, we can identify two possible avenues of resolving this issue. One, fundamentally change the configuration of the arm such that there is some constant set length between the second revolute joint and the prismatic joint of the arm. This way we can generate an equation of motion for each link while also properly including the effects of each joint. Second, we can redefine the second revolute joint and prismatic joint as a singular cylindrical joint such that a single joint variable controls both the rotation and extension of the second link. This however would introduce a new issue of not being able to separately rotate and extend.

RESULTS

A. Forward Kinematics

The following are numerical illustrations of example configurations for the robotic arm. Since the dynamics formulation is known to be incorrect, there is no numerical result shown. For all of the following results we assume $m_{camera} = 1kg$ and $L = 0.5m$ for each link.

For the Forward Kinematics, assume the following joint angles:

$$\theta = \begin{pmatrix} \theta_1 = \frac{\pi}{4}\,rads \\ \theta_2 = \frac{\pi}{4}\,rads \\ \theta_3 = \frac{L}{2}\,m \\ \theta_4 = -\frac{\pi}{4}\,rads \end{pmatrix} \tag{23}$$

Let the coordinates of the end-effector in the body-frame be represented as:

$$p_b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{24}$$

We can represent the coordinates of the end-effector using the following formula:

$$p_S = T_{SB}p_b \tag{25}$$

Solving (25) using (31) for T_SB we obtain our resulting coordinates of the end-effector in the base frame.

$$p_s = \begin{pmatrix} p_x = -0.729m \\ p_y = 0.729m \\ p_z = 1.03m \end{pmatrix} \tag{26}$$

B. Inverse Kinematics

For the Inverse Kinematics, assume the same coordinates from (26) are used to determine the required joint angles. Using (5), (6), (7), and (8), we successfully recreate the joint configuration that was used in the Forward Kinematics formulation.

$$\theta = \begin{pmatrix} \theta_1 = 0.786\,rads \\ \theta_2 = 0.787\,rads \\ \theta_3 = 0.249\,rads \\ \theta_4 = -0.786\,rads \end{pmatrix} \tag{27}$$

C. Velocity Kinematics

For the Velocity Kinematics, using the same joint angle configuration as in (23) and the following joint velocities:

$$\dot{\theta} = \begin{pmatrix} \dot{\theta}_1 = \frac{\pi}{6}\,\frac{rads}{s} \\ \dot{\theta}_2 = \frac{\pi}{6}\,\frac{rads}{s} \\ \dot{\theta}_3 = \frac{L}{6}\,\frac{m}{s} \\ \dot{\theta}_4 = \frac{\pi}{6}\,\frac{rads}{s} \end{pmatrix} \tag{28}$$

We can solve (13) to obtain the final base frame twist of the end-effector:

$$\mathcal{V}_S = \begin{pmatrix} \omega_x = 1.00\,\frac{rads}{s} \\ \omega_y = 0.00\,\frac{rads}{s} \\ \omega_z = 0.52\,\frac{rads}{s} \\ v_x = 0.00\,\frac{m}{s} \\ v_y = 0.99\,\frac{m}{s} \\ v_z = -0.022\,\frac{m}{s} \end{pmatrix} \tag{29}$$

D. Statics

Using the same joint configuration as (23), we solve (18) to obtain the following:

$$\tau_S(\theta) = \begin{pmatrix} \tau(\theta_1) = -4.91\,Nm \\ \tau(\theta_2) = -0.74\,Nm \\ \tau(\theta_3) = -1.49\,Nm \\ \tau(\theta_4) = -1.53\,Nm \end{pmatrix} \tag{30}$$

DISCUSSION

Future work will explore improvements in the design of the prototype, such as adding an extra joint in the base frame rotating on the x-axis to increase the possible configurations. This change will allowed us to expand the task space and create a more compact resting position, which will be useful for confined space exploration.

CONCLUSIONS

We developed the prototype for extender vision robotic arm with a RRPR structure. We calculated forward kinematics through the PoE and DH method, inverse kinematics with a geometrical approach, velocity kinematics in terms of the space frame, statics and dynamics to calculate the effect of the camera on the tip of the end-effector. The project allowed us to work on some core concepts relevant to the field of robotics , such as the application of the Rodriguez formula and the Lagrangian theorem.

REFERENCES

[1] Marko B. Popovic, Matthew P. Bowers, "Kinematics and Dynamics," Marko B. Popovic. Academic Press, 2019, Pages 11-43. ISBN 9780128129395.
[2] Kevin M. Lynch, Frank C. Park, Modern Robotics: Mechanics, Planning, and Control, Preprint-1st ed., Cambridge University Press, 2017

APPENDIX

A. Inverse Kinematics

$\theta_1 = atan2(y, -x)$
$\theta_2 = \Omega(x, y, z) + \phi_1(x, y, z, \theta_3) - \frac{\pi}{2}$
$\theta_3 =$ *defined heuristically*
$\theta_4 = \phi_3(x, y, z, \theta_3) - \pi$

where

$$\Omega = cos^{-1}\left(\frac{2L^2 - 2Lz}{2L\sqrt{x^2 + y^2 + (z-L)^2}}\right)$$

$$\phi_1 = cos^{-1}\left(\frac{(L+\theta_3)^2 + x^2 + y^2 + (z-L)^2 - L^2}{2(L+\theta_3)\sqrt{x^2 + y^2 + (z-L)^2}}\right)$$

$$\phi_3 = cos^{-1}\left(\frac{L^2 + (L+\theta_3)^2 - x^2 - y^2 - (z-L)^2}{2L^2 + 2L\theta_3}\right)$$

T_{SB} below...

B. Forward Kinematics T_{SB}

$$T_{SB} = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_2+\theta_4)\sin(\theta_1) & -\cos(\theta_2+\theta_4)\sin(\theta_1) & -\sin(\theta_1)\phi_1 \\ \cos(\theta_1) & \sin(\theta_2+\theta_4)\cos(\theta_1) & \cos(\theta_2+\theta_4)\cos(\theta_1) & \cos(\theta_1)\phi_1 \\ 0 & -\cos(\theta_2+\theta_4) & \sin(\theta_2+\theta_4) & L + L\sin(\theta_2+\theta_4) + L\sin(\theta_2) + \theta_2\sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{31}$$

where

$$\phi_1 = Lcos(\theta_2+\theta_4) + Lcos(\theta_2) + \theta_3 cos(\theta_2)$$