CUSP Kinematic Model Development

Lucia Martinez *M. John D. Hayes*

Department of Mechanical & Aerospace Engineering, Carleton University
1125 Colonel By Drive, Ottawa, ON, Canada, K1S 5B6

1. Motivation

The objective of the Carleton University Simulator Project (CUSP) is to design and build a complete reconfigurable motion simulation facility accommodating different vehicle types. The design of the motion platform is one of the essential components of the facility. Kinematic modeling is crucial for motion platform analysis and control [1]. Since the platform requires output in real time, developing a computationally efficient kinematic model is essential. A simple study is currently being performed in order to determine the most efficient kinematic modeling technique. The results of this study will determine the future path of the CUSP kinematic model development.

2. Planar Parallel 3-RPR Manipulator

In order to study different kinematic models, a simple planar parallel manipulator was chosen. Choosing a planar over a spatial manipulator is advantageous mainly because of its analytical simplicity; it is easier to analyze a 2D planar mechanism as compared to a 3D space mechanism. The 3-RPR planar manipulator selected for this study and presented in Figure 1, consists of a platform connected to the fixed base by three independent kinematic chains having one d.o.f joints, one of which is actuated [2]. Each chain is of RPR configuration where there are two passive revolute (R) joints and one active prismatic (P) joint in between. Two methods were selected to analyse the kinematics of the planar parallel 3-RPR manipulator: the first uses geometry in the Euclidian plane [1,3] and the second uses kinematic mapping [4].

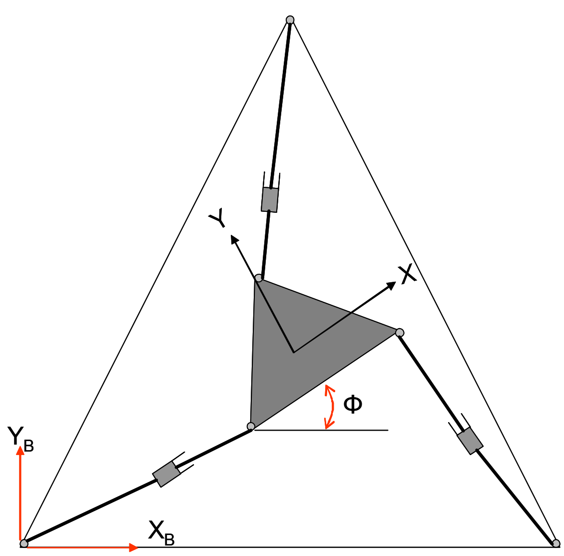

Figure 1: 3-RPR Manipulator

3. Kinematics Using Plane Euclidian Geometry

Solving the kinematics of the planar 3-RPR manipulator using Euclidian geometry requires detailed analysis. The solution of the inverse pose kinematics is quite trivial but solving the forward pose kinematics requires in depth analysis. To solve for the inverse pose kinematics Williams approach [3] was used. As for the forward pose kinematics, Gosselin's methodology [1] is to be exercised.

3.1 Inverse Pose Kinematics

The inverse pose kinematics problem requires the calculation of the lengths L_i between each set of two revolute joints connected by the prismatic joint. To calculate these lengths, the desired Cartesian pose $X = \{x\ y\ \Phi\}^T$, is given. The revolute joint C_i located on the moving platform can easily be located with the given information. To calculate the lengths, one requires to find the vector length between revolute joint C_i and its corresponding revolute joint A_i, located on the fixed base [4]. For each RPR chain, the following closed loop equation may be used:

$$C_i = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{bmatrix} {}^M C_i \qquad (1)$$

Where ${}^M C_i$ represents the revolute joint located in the moving platform and its location is given with respect to the moving frame.

Once C_i is found, the lengths L_i are calculated using the Euclidian norm:

$$L_i = \| C_i - A_i \| \qquad (2)$$

3.2 Forward Pose Kinematics

The forward pose kinematics requires one to calculate the position and orientation of the platform $X = \{x\ y\ \Phi\}^T$, when all that is given are the lengths L_i. There is no closed loop form for solving the forward pose kinematics. If Gosselin's methodology is followed, one needs to compute a univariate polynomial of order 6. The solution of this polynomial is easily obtained by using Matlab's function ROOTS(C), which computes the roots of the polynomial whose coefficients are the elements of the vector. Gosselin's method is outlined in detail in [1] and because of its lengthy analysis, the details are not shown here.

4. Kinematic Mapping

Kinematic mapping was selected as the second method to be studied to solve the kinematics of the planar parallel 3-RPR manipulator. One of the biggest advantages of planar kinematic mapping is that its constraint equations can be used by any three legged planar platform possessing three degrees of freedom [5]. This method is explained in detail in [4,5] and for the purpose of this paper kinematic mapping will just be briefly introduced.

4.1 Method

Any planar displacement (a, b, φ), such as the one experienced by a point on the moving platform of a 3-RPR manipulator, may be described by a reference coordinate system E relative to a fixed plane with coordinate system ∑. This displacement is mapped to a distinct point in a 3-D projective image space as seen in figure 2.

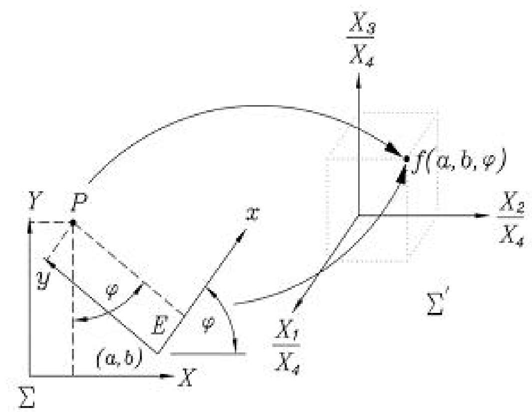

Figure 2: Kinematic mapping.

The kinematic mapping image coordinates are defined with respect to the Cartesian displacement (a, b, φ) as follows.

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} a\sin(\varphi/2) - b\cos(\varphi/2) \\ a\cos(\varphi/2) + b\sin(\varphi/2) \\ 2\sin(\varphi/2) \\ 2\cos(\varphi/2) \end{bmatrix}. \quad (3)$$

Using various trigonometric substitutions, and the relationships in (3), any Euclidian planar displacement can be written in terms of image points.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_4^2 - X_3^2 & -2X_3X_4 & 2(X_1X_3 + X_2X_4) \\ 2X_3X_4 & X_4^2 - X_3^2 & 2(X_2X_3 - X_1X_4) \\ 0 & 0 & X_3^2 + X_4^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4)$$

4.2 Inverse Pose Kinematics

In order to solve the inverse kinematics or the 3-RPR manipulator, the active P joint is locked in position and one R joint is fixed in ∑ while the other R joint represented by (a , b, φ) is constrained to rotate in a circle of fixed radius, clearly this radius represents the leg length L_i which we are trying to solve for. Writing the quadratic equation of this circle with respect to the image space coordinates and then solving for the unknown radius leads to the solution of the inverse pose kinematics problem.

4.3 Forward Pose Kinematics

Even with the kinematic mapping approach, there are no closed form solutions to the forward pose kinematics problem. One needs to solve 3 constraint equations simultaneously, or extract a 6^{th} order univariate, which gives 6 possible solutions.

5. Results of Methods' Efficiency

In order to determine how efficient a certain method was, Matlab testing was performed. The Matlab function FLOPS which returns the cumulative number of floating point operations was used for each method. CPU time was also accounted for. Table 1 outlines the results for each method implementing the inverse pose kinematics problem.

Table 1: Results of Inverse Kinematics Problem

	Euclidean Approach	Kinematic Mapping
# of Flops	68	386
CPU time (s)	0.11	1.26

6. Conclusion

The Matlab test results indicate that the method which uses Euclidian geometry may be the most efficient. These two methods will soon be tested for the forward pose kinematics problem. Once all the testing is performed, the results will be used to determine whether CUSP will engage in using Kinematic mapping or the Euclidian approach for further kinematic model development.

References

[1] J.P Merlet, *Parallel Robots*, Kluwer Academic Publishers, 2000.
[2] J.P Merlet, "Direct kinematics of planar parallel manipulators", *IEEE International Conference on Robotics and Automation*, 1996, 4: 3544-3749 .
[3] R.L Williams II, "Planar Parallel 3-RPR Manipulator", *Proc. of the Sixth Conference on Applied Mechanism and Robotics*, 1999.
[4] O. Bottema and B. Roth, *Theoretical Kinematics*, Dover Publications, 1990.
[5] P.J Zsombor-Murray, C. Chen, MJD Hayes, "Direct Kinematic Mapping for General Planar Parallel Manipulators", *Proc. CSME Forum*, 2002.