DEVELOPMENT OF THE FORWARD KINEMATICS FOR ROBOT FINGERS BY USING ROBOREALM

Jamaludin Jalani

Dept. of Mechanical Engineering, University of Bristol, Queen's Building University Walk Bristol BS8 1TR UK,
Bristol Robotics Laboratory (BRL) DuPont Building Bristol Business Park Bristol BS16 1QD UK,
University Tun Hussein Onn Malaysia, 86400 Parit Raja Batu Pahat Johor Malaysia.
Email: meyjj@bris.ac.uk, jamalj@uthm.edu.my

ABSTRACT

This paper presents a new method to capture the parameters from robot fingers by using RoboRealm tool. The unknown parameters of robot fingers can be captured and recorded in real time implementation for developing the forwards kinematics. The use of RoboRealm in this project is due to its powerful vision software application, which is utilized in machine vision, image analysis, and image processing systems. Here, in the case of the Bristol Elumotion Robot Fingers (BERUL), RoboRealm will assist to capture the angular parameters of joint 1 (θ_1), joint 2 (θ_2) and joint 3(θ_3). Then, a linear relationship between θ_1 and θ_2 as well as θ_2 and θ_3 can be computed. Having found these angle relationships, the forward kinematics of robot fingers can be developed easily and accurately as illustrated throughout this paper.

Keywords: *RoboRealm; Forward Kinematics; Underactuated Robot Finger*

1. INTRODUCTION

The end-effector will be at particular position in space when there is a particular set of values of joint angles and distance between the links. The analysis to find both position and orientation of the end-effector with respect to the base frame for the given set of joint parameters is known as the forward kinematics. In order to carry out this analysis, we will attach coordinate frames to each link and then the position and orientation of these frames are used for specifying the links. Similarly to the BERUL fingers, the frame has to be attached on each link to develop it owns kinematics. However, due to the unknown information of joint relationships between θ_1 and θ_2 as well as θ_2 and θ_3 (see Figure 2), they have to be found to obtain a complete derivation of the forward kinematics. Hence, deriving the forward kinematics of the BERUL fingers is not straightforward. The joint relationships have to be computed based on experimental analysis of the robot hand. Manual computation such as using an angle ruler and etc will not be an appropriate because it is always difficult, imprecise and insufficient. In fact, since BERUL fingers are classified as an underactuated system, the measurement is always inaccurate due to the data keep changing and eventually, it will discourage to continue the experiment. All of these problems will be resolved if the manufacturer had provided us sufficient information such as the length of the finger, the gear ratio and etc. Insufficient information by the manufacturer is also resulting in an inaccuracy of kinematics derivation.

Alternatively, we have used the RoboRealm tool [1] to find out the relationship of the joints. In general, Roborealm is an application for use in computer vision, image analysis, and robotic vision systems for which an easy point and click interface RoboRealm simplify vision programming. It uses an inexpensive USB webcam and by adding RoboRealm software in the PC, the computation of joint relationships can be easily obtained. The obtained joint relationships for the BERUL fingers by RoboRealm will lead to the computation of a full forward kinematics transformation. Then, we will exploit the knowledge of this kinematics where it contains position, orientation, and velocity analysis of manipulators which can be used for tracking and grasping of the BERUL fingers. Specifically, we have deployed the Jacobian of the forward kinematics in order to implement a Cartesian coordinate space control. However, in this paper, we only discuss and show the method to derive the joint relationships of the BERUL fingers by using RoboRealm explicitly. The implementation of the Cartesian coordinate space control will not be included here. Previous work on the kinematics based approaches that can be referred to for robot fingers control were in Montana [2, 3, 4, 5] and Hunt et al. [6]. More advanced kinematics analysis to resolve uncertainties in the kinematics which also can be referred to was in Doulgeri and Arimoto [7].

2. THE BERUL FINGER

Figure 1 shows the constrained underactuated BERUL fingers for which the kinematics analysis is developed. Strictly speaking, the BERUL fingers have many problems such as significant friction and stiction and unknown parameters; it is light in weight and fragile. Moreover, the hand has 5 fingers with 16 degrees of freedom and most of the fingers are underactuated. All of these problems need to be resolved in order to achieve a successful grasping task. In the next section, we propose a RoboRealm tool to find the joint relationships of the BERUL fingers. This will speed up a derivation of the BERUL fingers's kinematics as well as obtaining more accurate data.

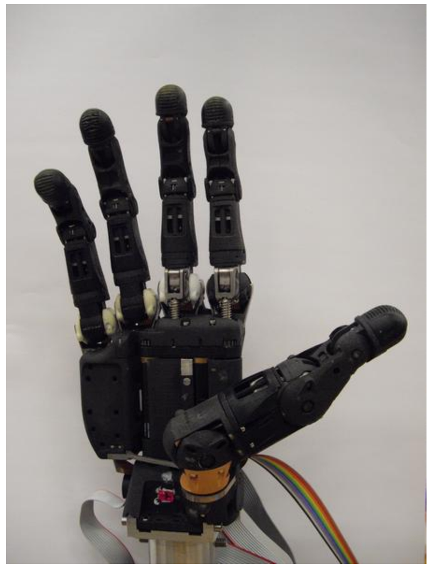


Figure 1: Underactuated BERUL fingers

3 . ROBOREALM

Actuating a single finger and moving it to different steady state positions allows us to capture kinematic data with Roborealm. The capturing process is carried out automatically and can be repeated easily such that the obtained data become more accurate. In our case we captured the angular data of joint 1 (θ_1), joint 2 (θ_2) and joint 3(θ_3) (see Figure 2).

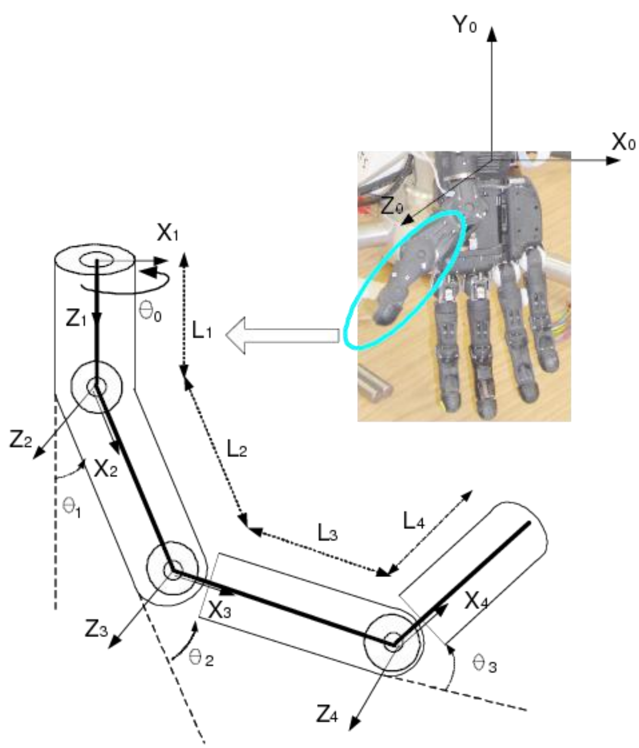


Figure 2: A four-link constraint thumb

Figure 3 shows a RoboRealm environment where a video processing for capturing the angle parameters can be executed. In order to identify the joints of the BERUL fingers, a color patch has to be placed on each joint by selecting one of the colors in RoboRealm library suited to the *RGB Filter Module*. Here we have selected five different colors namely *green, red, blue, magenta* and *yellow* which must be labeled by using the *Marker Module*. Then, these colors will be captured by a webcam. However, the webcam can only distinguish these colors if they are different in contrast and/or brightness. As a result, capturing the patches on the BERUL fingers in a dark area will be difficult. We have used appropriate light so that the patches can be distinguished easily. Nevertheless, the *Erode Module* can also assist the webcam to capture the patches by eliminating the unwanted blob. Patches that are connected with other patches will become separated. In addition, this module is useful for removing noise from a patch, besides larger patches will have smoother boundaries. The next step is to assign the center of gravity by using the *Center of Gravity Module* for each patch before a line can be made to connect two patches by choosing the *Display Angle Module*. A desired connecting line between two patches can only be accomplished if we assign variables to each color patch. This can be realized by using the *VBScript Program Module*. Once the two lines are connected, the joint angles can be computed by the *Calculate Angle Module* (three patches are needed to perform this). At this stage, by using the *Display Variables Module*, we can see the computed angles on a current video. These display data can be recorded in an array format and saved in $*.csv$ file. For this, the *Write Variables Module* is employed. In the case of the BERUL fingers array 1 will be joint 1, array 2 will be joint 2 and array 3 will be joint 3. Then, the relationship of the joints can be easily plotted by Matlab tool.

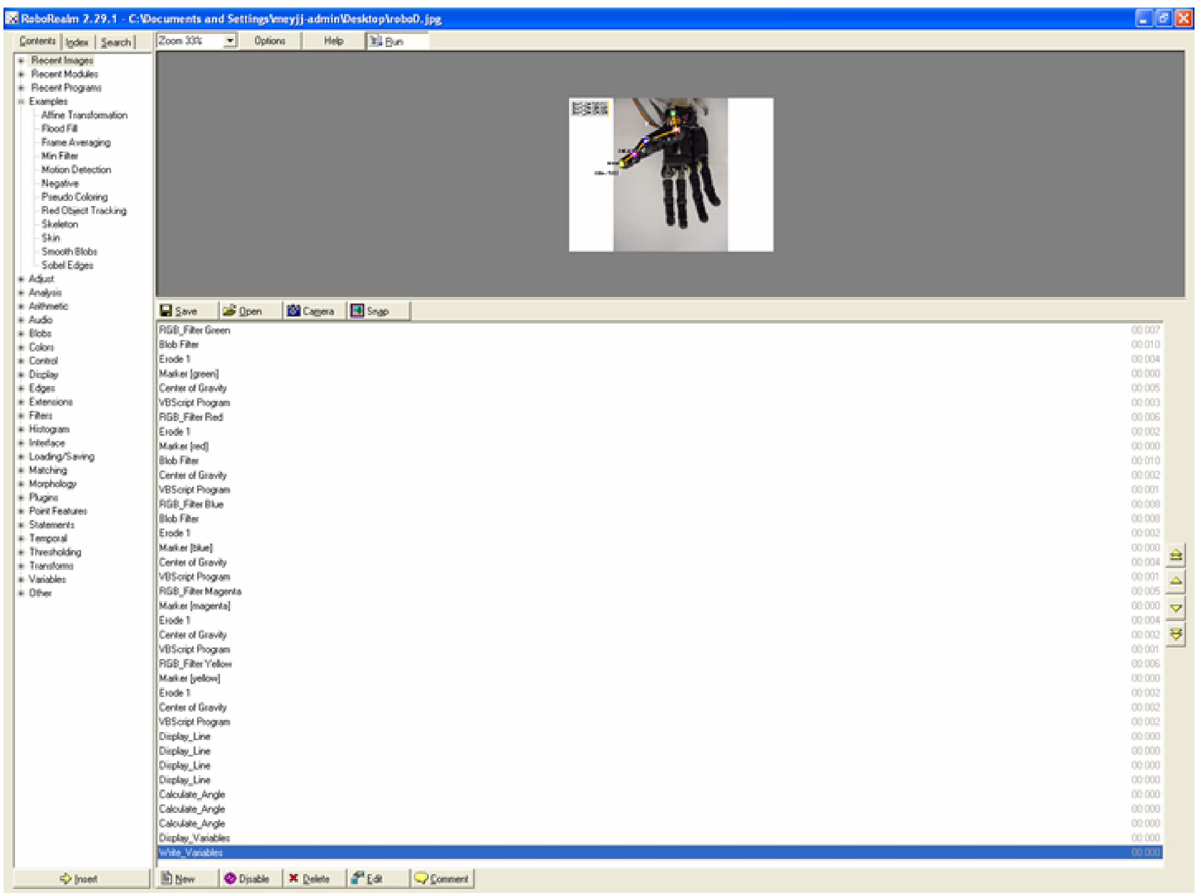


Figure 3: RoboRealm Environment

See Figure 4 in order to have better insight of Roborealm used in the BERUL fingers. Note that the calculated angle in RoboRealm is in a clockwise rotation for instance, *joint_1* is the angle starts from point green to point blue. As a result, to get a desired joint angles (see Figure 2) we need to use equation (1). It is much easier to compute a desired angle where only five colors are needed as shown in Figure 4, then uses the equation (1). Moreover, Figure 5(a) and Figure 5(b) illustrate the motion of a thumb finger from a resting state to a fully flexed position. The joint's parameters are displayed at the upper left corner of each motion finger and has been set in degree ($joint\_1$ reflects θ_{1b}, $joint\_2$ reflects θ_{2b} and $joint\_3$ reflects θ_{3b}). Here, θ_{1b} is the angle created by green, red and blue patches, θ_{2b} is the angle formed by the line connecting the red, blue and magenta patches and θ_{3b} is the angle between point blue, magenta and yellow. The actual angles as required for Figure 2 are computed as follows:

$$Desired\ angels\ of\ \theta_{1/2/3} = 180\,deg - captured\ angles\ of\ \theta_{1b/2b/3b} \tag{1}$$

Alternatively to the process above, we can use more than five colors by using $RGB\ Filters\ Module$ to get θ_1, θ_2 and θ_3 directly as in Figure 2, but it is not recommended due to the limitation of the colors used in RoboRealm library. In addition, as mentioned earlier the selected colors must be different in contrast and/or brightness such that they can be distinguished by a webcam easily.

In summary, we have used the following modules to capture the joint parameters:

1. ***RGB Filter*** : to focus the attention towards the primary RGB colors.
2. ***Erode*** : performs an erosion routine.
3. ***Marker*** : provides us with a way to identify or label a spot within the processing pipeline. Pipeline is the area where the module is added and processed.
4. ***Center of Gravity (COG)***: calculates where the COG of the image lies.
5. ***VBScript Program*** : provides a way to create custom Visual Basic scripts that can be used to process image statistics.
6. ***Display Line*** : provides a way to draw lines based on line coordinates.
7. ***Calculate Angle*** : provides a way to easily calculate the angle between two lines defined by three points.
8. ***Display Variables***: draws variables and their values into the current video .
9. ***Write Variables*** : provides an interface to write RoboRealm variables to disk.

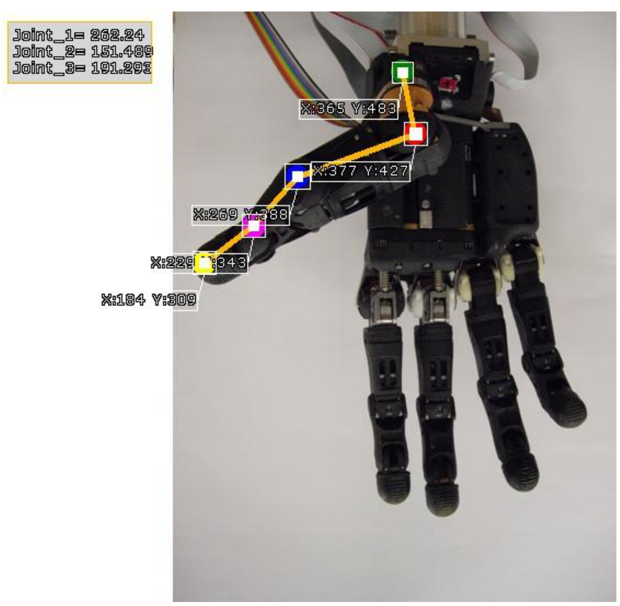
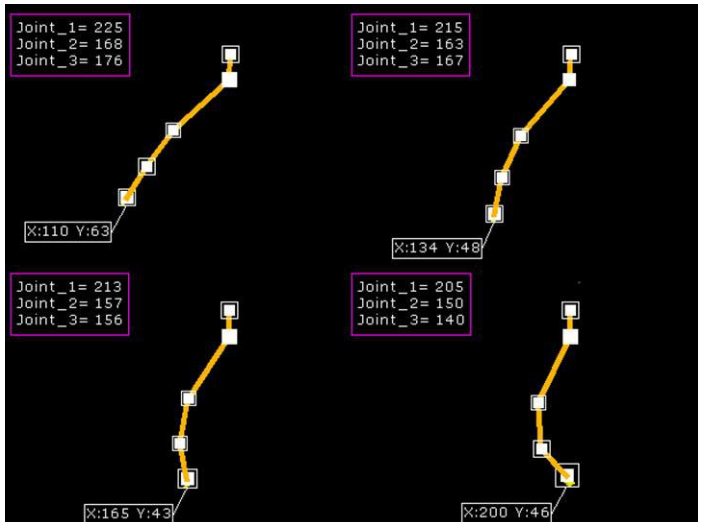
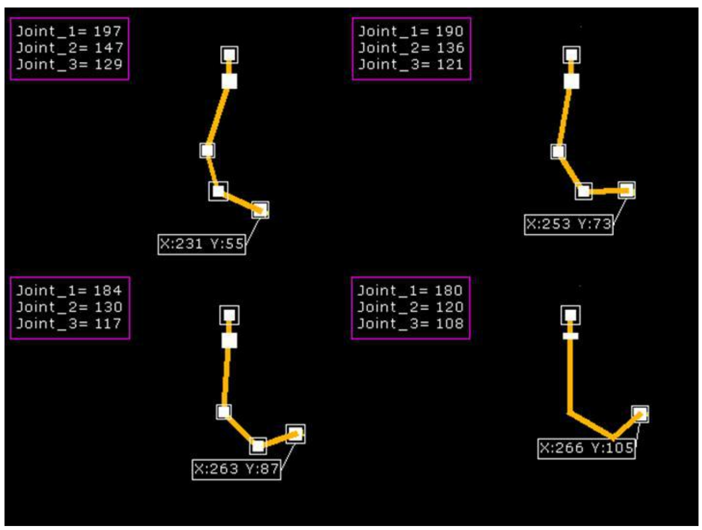


Figure 4: Measuring the relationships between joint 1(θ_1) and joint 2(θ_2), joint 2(θ_2) and joint 3(θ_3)



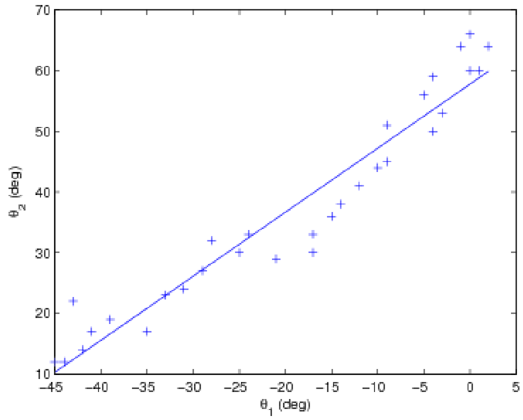
(a) Angle starts from a resting state



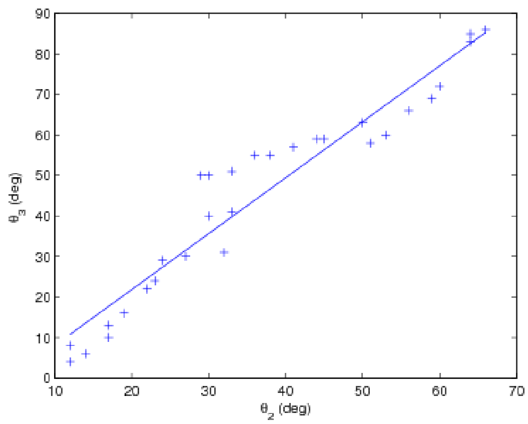
(b) Angle rotates to a fully flexed position

Figure 5: Fingers Kinematics starts from a resting state to a fully flexed position

Finally, with the help of Matlab, a linear relationship between θ_1 and θ_2, θ_2 and θ_3 are found and shown in Figure 6(a) and Figure 6(b) respectively. For this, we recorded several static images of each joint angle relationship to reliably derive the linear relationship between the joint angles using *Least Square* method [8]. As a result, equation (2) and equation (3) are obtained. Equation (2) shows the relationship between θ_1 and θ_2 while equation (3) shows the relationship between θ_2 and θ_3 for the thumb finger.



(a) A linear relationship between θ_1 and θ_2



(b) A linear relationship between θ_2 and θ_3

Figure 6: Matlab computes linear relationships

A linear relationship between θ_1 and θ_2 for a thumb finger can be written as

$$\theta_2 = 1.0552\theta_1 + 57.7723. \tag{2}$$

A linear relationship between θ_2 and θ_3 for a thumb finger can be written as

$$\theta_3 = 1.3805\theta_2 - 5.7672. \tag{3}$$

Since the default unit in Matlab is in radian, thus at this point, we have converted all the obtained results from degree to radian for the convenience. Hence, equation (2) becomes equation (4) and equation (3) will be equation (5).

A linear relationship between θ_1 and θ_2 for a thumb finger in radian is

$$\theta_2 = 1.0552\theta_1 + 1.0078 \tag{4}$$

and a linear relationship between θ_2 and θ_3 for a thumb finger in radian is

$$\theta_3 = 1.3805\theta_2 - 0.1006. \tag{5}$$

4. DH REPRESENTATION

Having found the joint relationships of equation (4) and equation (5), a full kinematics representation of the BERUL fingers can be computed by the Denavit Hartenberg technique [9]. Here, only the example of a four-link thumb is outlined as shown in Figure 2; it has a more complicated mechanical design structure as compared to the other fingers. The same figure also shows the reference frame of (X_o, Y_o, Z_o) where it is located at the middle of the hand's wrist. Table 1 illustrates its transformation parameters for each link. Substituting equation (4) and equation (5) into Table 1 yields transformation parameters for each link as shown Table 2.

Table 1: Link Parameter of the four-link planar manipulator

i	α_i (rad)	a_i (m)	d_i (m)	Θ_i (rad)
1	0.78+ α_0	-0.02	L1	Θ_0
2	0	L2	0	Θ_1
3	0	L3	0	Θ_2
4	0	L4	0	Θ_3

Table 2: A New Link Parameter of the four-link planar manipulator

i	α_i (rad)	a_i (m)	d_i (m)	Θ_i (rad)
1	0.78+ α_0	-0.02	L1	Θ_0
2	0	L2	0	Θ_1
3	0	L3	0	$1.0552\theta_1 + 1.0078$
4	0	L4	0	$1.3805\theta_2 - 0.1006$

The general transformation formula is as below:

$$T_i^{i-1} = \begin{bmatrix} Cos\theta_{i-1} & -Sin\theta_{i-1}Cos\alpha_{i-1} & Sin\theta_{i-1}Sin\alpha_{i-1} & a_{i-1}Cos\theta_{i-1} \\ Sin\theta_{i-1} & Cos\theta_{i-1}Cos\alpha_{i-1} & -Cos\theta_{i-1}Sin\alpha_{i-1} & a_{i-1}Sin\theta_{i-1} \\ 0 & Sin\theta_{i-1} & Cos\alpha_{i-1} & d_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Substituting the link parameters shown in Table 2, the individual transformation from each link can be computed as follows:

$$T_1^0 = \begin{bmatrix} Cos\theta_0 & -Sin\theta_0 Cos(0.78+\alpha_0) & Sin\theta_1 Sin(0.78+\alpha_0) & a_0 Cos\theta_0 \\ Sin\theta_0 & Cos\theta_0 Cos(0.78+\alpha_0) & -Cos\theta_1 Sin(0.78+\alpha_0) & a_0 Sin\theta_0 \\ 0 & Sin\theta_0 & Cos\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$T_2^1 = \begin{bmatrix} Cos\theta_1 & -Sin\theta_1 & 0 & a_1 Cos\theta_1 \\ Sin\theta_1 & Cos\theta_1 & 0 & a_1 Sin\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$T_3^2 = \begin{bmatrix} Cos(1.0552\theta_1 + 1.0078) & -Sin(1.0552\theta_1 + 1.0078) & 0 & a_2 Cos(1.0552\theta_1 + 1.0078) \\ Sin(1.0552\theta_1 + 1.0078) & Cos(1.0552\theta_1 + 1.0078) & 0 & a_2 Sin(1.0552\theta_1 + 1.0078) \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

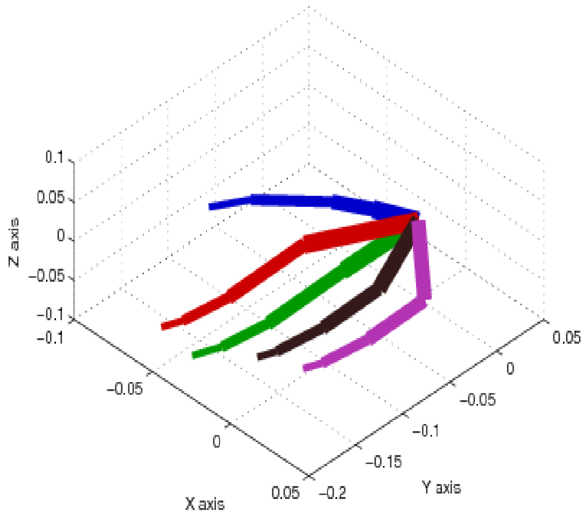
$$T_4^3 = \begin{bmatrix} Cos(1.3805\theta_2 - 0.1006) & -Sin(1.3805\theta_2 - 0.1006) & 0 & a_3 Cos(1.3805\theta_2 - 0.1006) \\ Sin(1.3805\theta_2 - 0.1006) & Cos(1.3805\theta_2 - 0.1006) & 0 & a_3 Sin(1.3805\theta_2 - 0.1006) \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, the full transformation will be as follows:

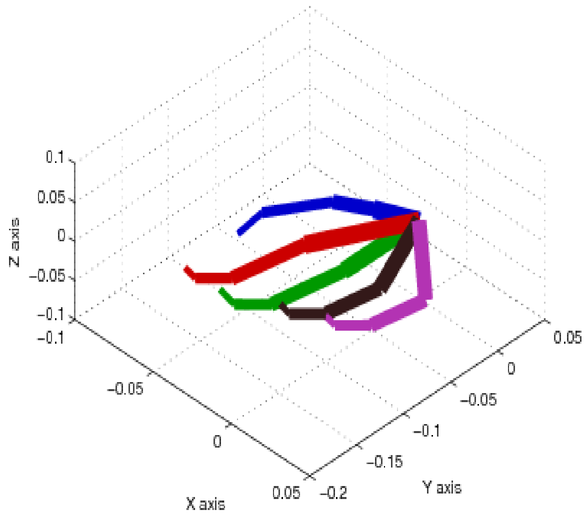
$$T_4^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \tag{11}$$

5. KINEMATICS RESULTS

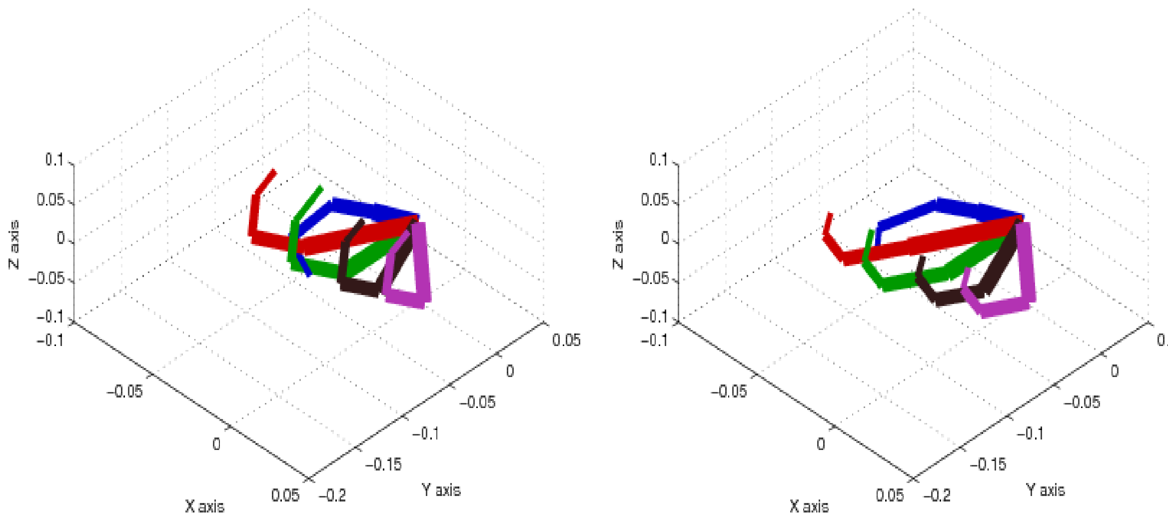
The same method by using RoboRealm and DH transformation will be applied to other BERUL fingers namely index, middle, ring and small fingers. It is to note, there is an angle sensor for each actuated joint. A sensor signal M_θ relates to θ_1 in a linear manner, e.g. for the thumb $\theta_1 = -0.2924 M_\theta - 0.0419$. The result of the kinematics for each finger is exhibited in Figure 7(a), Figure 7(b), Figure 7(c), and Figure 7(d) when $M_\theta = 0\,rad$, $M_\theta = -0.2\,rad$, $M_\theta = -0.4\,rad$ and $M_\theta = -0.6\,rad$ respectively. It is found that the results of the forward kinematics of the BERUL fingers is correct and sufficient enough to implement in the Cartesian space control in future. Although the kinematics can be derived manually, the method will be difficult and the obtained data will be inaccurate and insufficient.



(a) Fingers Kinematics in resting state



(b) Position for a $M_\theta = -0.2\,rad$ (for all fingers)



(c) Position for a $M_\theta = -0.4\,rad$ (for all fingers) (d) Position for a $M_\theta = -0.6\,rad$ (for all fingers)

Figure 7: Results of Forward Kinematics

6. CONCLUSIONS

In this paper, the joint relationships between θ_1 and θ_2 as well as θ_2 and θ_3 have been computed by RoboRealm. Then a complete derivation of the forward kinematics for the BERUL fingers are developed easily. Here, RoboRealm provides us simple and easy method for capturing the angle parameters for each joint and it is more accurate. The results of the forward kinematics show that the motion of the BERUL fingers is correct and sufficient enough to implement in the Cartesian space control in future.

7. ACKNOWLEDGEMENT

The CHRIS (Cooperative Human Robot Interaction Systems) project is funded by the European Commission's Seventh Framework Programme (FP7) and will run from 2008-2012. This research is also partially funded by the Malaysian Government.

8. REFERENCES

[1]. Steven Gentners. Roborealm, January 1996. http://www.roborealm.com/.
[2]. D.J. Montana. Contact stability for two-fingered grasps. In *IEEE Transactions on Robotics and Automation*, volume 8, pages 421 –430, aug 1992.
[3]. David J. Montana. The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3): 17–32, 1988.
[4]. D.J. Montana. *Tactile sensing and the kinematics of contacts*. PhD thesis, 1986.
[5]. D.J. Montana. The kinematics of contact and grasp. *International Journal Robotics Research*, 8(4):421–430, aug 1995.
[6]. K.H. Hunt, A.E. Samuel, and P.R. McAree. Special Configurations of Multi-finger Multi- freedom Grippers- A Kinematic Study. *The International Journal of Robotics Research*, 10(2):123–134, 1991.
[7]. Zoe Doulgeri and Suguru Arimoto. A force commanded impedance control for a robot finger with uncertain kinematics. *The International Journal of Robotics Research*, 18(10):1013–1029, 1999.
[8]. Wolfram, S. *The Mathematica Book*. Addison-Wesley Publishing Company, 4th edition, 1999.
[9]. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, second edition, 1955.