[AMN03] Trajectory planning for smooth motion of a surgical robot

Lee Jer Vui[1], Zahari Taha[1], Yap Hwa Jen[1], Iskandar[1], Mohamed Razif Mohamed Ali[2]

[1]Centre of Product Design & Manufacturing, Department of Engineering Design & Manufacture, Faculty of Engineering, University of Malaya, 50603, Kuala Lumpur, Malaysia.
[2]Department of Orthopaedic Surgery, Faculty of Medicine, University of Malaya, 50603, Kuala Lumpur, Malaysia.

Introduction

Surgical robotics offers a number of attractive features. It can be used to filter out surgeon's hand tremor, enhance a surgeon's motions and perform microscale movements that are otherwise impossible for a human to perform. Because the control workstation can be located away from the surgical theatre, it enables a surgeon to manipulate an instrument during radiation or chemical treatment, via a robot, distancing the surgeon from harmful effects. Computer aided surgical simulation via virtual reality improves the process of surgical training. New surgeons could ultimately be trained entirely by computer simulation. Furthermore, the virtual environments provide an environment where there is no risk to a patient.

Figure 1 shows the overview of the developed robot surgical system. A surgeon can interact with his/her environment by visually observing the robots and directing their motion to patients via a virtual reality (VR) system. The VR system involves computer-based generation of three-dimensional (3D) visual, auditory, and tactile environments, and a set of interface tools e.g. sensing glove that allows surgeons to navigate and interact with objects e.g. bones in the computer-generated environment. The models e.g. tissues, organs and fracture bones are in the form of digital 3D model, which is generated from CAT scan slices. Then the models are imported into the simulator. Various types of surgical tools and implants can be loaded into the simulator from the database library.

The surgical simulator acts as the heart of the robot surgical system. The surgical simulator supports training of surgical skill, planning and decision making in the surgical operation. The procedures can be simulated many times and the most perfect procedures will be selected for the real surgical operation. Industrial robot (Model: Kuka KR-6) will be used to execute the surgical operation.

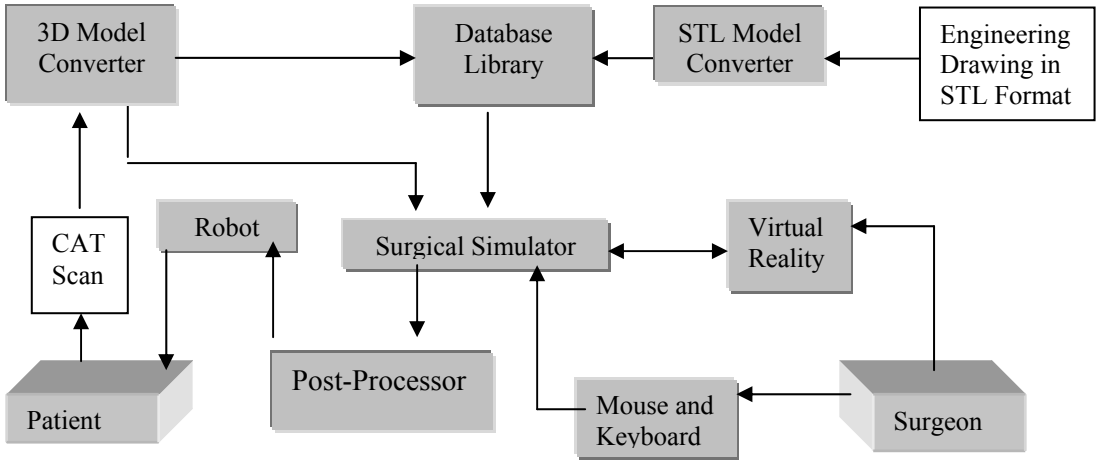

FIGURE 1 The overview of the developed robot surgical system.

Methodology

The simulator is designed and developed based on C++ programming language with Open GL. Open GL is a generic version of an interface made available openly in the interest of standardization of the language. Open GL

handles graphics primitives, 2D and 3D transformations, lighting, shading, Z buffering, hidden surface removal, and a host of other features.

In this system, the surgeon wears a data glove to pick and place the virtual objects or grasp a tool, according to the desired task as shown in Figure 2 and Figure 3. The system tracked the process steps and paths. The sensor tracker and data glove generates 25 data per second to reduce latency and flickering problems. It records the positions (X,Y and Z), orientations (roll and yaw) and a signal D1 of the surgeon's hand. Signal D1 indicates either on or off of the surgical tool that being used. For example, the drill will be on when signal D1 = 1. On the other hand, the drill will be off when signal D1 = 0. Figure 4 shows the flowchart that show how the data is being manipulated. The data recorded will go through a filtering process in order to keep the data to a minimum. The noise signal generated by hand tremor is also cut off. Repeated data e.g. the data glove stopping at a specific position is eliminated. Moreover, redundant data e.g. glove moving in a straight direction is eliminated. Two points will be enough to generate a straight line.

FIGURE 2 The data glove.

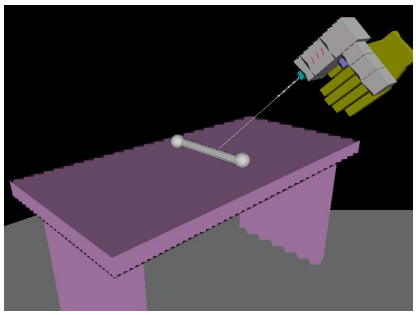

FIGURE 3 The virtual hand grasping a power drive.

After the filtering process, the data (X, Y and Z) is normalized: time (s) and displacement (m) is set to start from zero. Then, data smoothing is carried out. Data are fitted with polynomials ranging from the 5th to 9th degree. For m^{th} order polynomial:

$$x = a_0 + a_1 t + a_2 t^2 + ... + a_m t^m + e,$$

where e = residuals [1]

Velocity of the movement can be obtained by using central difference method

$$X_i' = \frac{x_{i+1} - x_{i-1}}{2h} \quad [2]$$

And acceleration can be derived by the same method by substituting x with x', thus,

$$X_i'' = \frac{x_{i+1}' - x_{i-1}'}{2h} \quad [3]$$

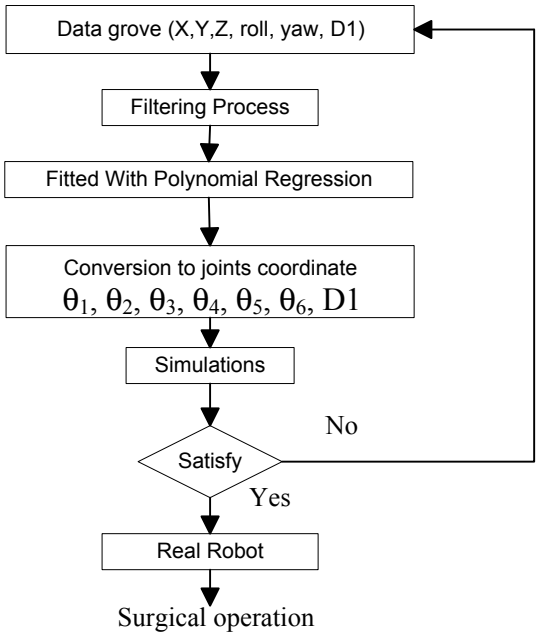

FIGURE 4 The flowchart of the data manipulation.

After fitting with polynomial, these data is sent to the simulator. These data is required to be converted to joint variables $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$, D1 for robot simulation. Kuka KRC6 is characterized by a spherical wrist, which the inverse kinematics problem can be separated into positional part *(θ_1, θ_2, θ_3)* and an orientation part *(θ_4, θ_5, θ_6)*. Then, it can be solved independently by using Equations [4-9]. The simulator has the capacity of real-time simulation of a virtual Kuka robot through 3D animation on a personal computer. The surgeon can control and visualize the virtual robot by mouse or keyboard through a

user-friendly interface window. The movement of the virtual robot can be simulated many times after the data has been loaded. The surgeon can change the viewing angle or zoom in from any position as shown in Figure 5. The simulator is used to:

I. Compare the filtered data with the original data.
II. Check the dynamic behaviour. Since the teaching system is based on the human hand, we need to check the robot limitations.
III. Check the robot orientations, reaches, clearances and collision detections.

After the surgeon is satisfied with the procedure, the database can be translated into a Kuka Robot Language (KRL) through a developed postprocessor. These KRL commands can be loaded into the controller of the real Kuka robot for real surgery.

$$\theta_1 = A\tan 2(P_{yw} / P_{xw}) \quad \text{and} \quad \theta_1 \pm 180° \qquad [4]$$

$$\theta_2 = A\tan 2\left[\frac{(d_1+d_2-P_{zw})(C_3a_4-S_3d_4+a_3)-(C_1P_{xw}+S_1P_{yw}-a_2)(S_3a_4+C_3d_4)}{(d_1+d_2-P_{zw})(S_3a_4+C_3d_4)+(C_1P_{xw}+S_1P_{yw}-a_2)(C_3a_4-S_3d_4+a_3)}\right] \qquad [5]$$

$$\theta_3 = A\tan 2\left[\frac{d_4}{a_4}\right] + A\tan 2\left[\frac{\pm\sqrt{a_3^2+d_4^2-[P_{xw}^2+P_{yw}^2+(d_1+d_2-P_{zw})^2-a_2^2-a_4^2-a_3^2-d_4^2]^2}}{P_{xw}^2+P_{yw}^2+(d_1+d_2-P_{zw})^2-a_1^2-a_3^2-a_4^2-d_4^2}\right] \qquad [6]$$

$$\theta_4 = a\tan 2\left(\frac{-S_1m_{13}+C_1m_{23}}{-C_1C_{23}m_{13}-S_1C_{23}m_{23}+S_{23}m_{33}}\right) \text{ and } \theta_4 \pm 180^o \qquad [7]$$

$$\theta_5 = a\tan 2\left(\frac{\pm\sqrt{(-S_1m_{13}+C_1m_{23})^2+(-C_1C_{23}m_{13}-S_1C_{23}m_{23}+S_{23}m_{33})^2}}{-C_1S_{23}m_{13}-S_1S_{23}m_{23}-C_{23}m_{33}}\right) \qquad [8]$$

$$\theta_6 = a\tan 2\left(\frac{C_1S_{23}m_{12}+S_1S_{23}m_{22}+C_{23}m_{32}}{-C_1S_{23}m_{11}-S_1S_{23}m_{21}-C_{23}m_{31}}\right) \text{ and } \theta_6 \pm 180^o \qquad [9]$$

($P_{XW} = X - d_6m_{13}$, $P_{YW} = Y - d_6m_{23}$, $P_{ZW} = Z - d_6m_{33}$, $S_n = \sin_n$, $C_n = \cos_n$, $a_2 = 300mm$, $a_3 = 650mm$, $a_4 = 115mm$, $d_1 = 300mm$, $d_2 = 375mm$, $d_4 = 600mm$ and $d_6 = 125mm$)

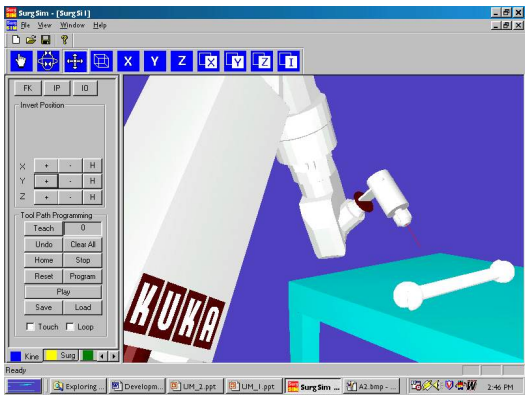

FIGURE 5 The developed robot simulator.

Results

Figure 6a to 6c show the graphs of displacement versus time respectively to X, Y and Z direction. For simplicity, only the actual displacement, 5th, 7th and 9th of order polynomial are shown. Graphs on velocities and accelerations versus time respectively to X, Y and Z direction are attached to Appendix.

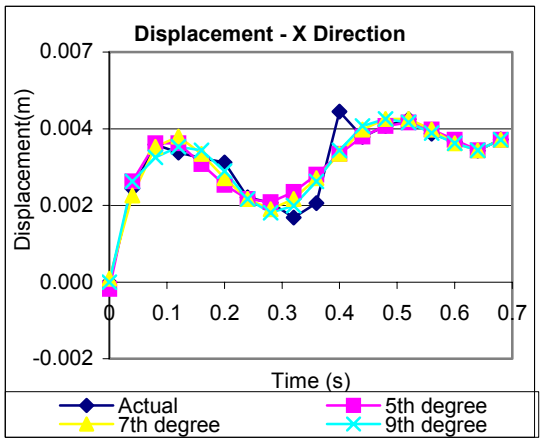

FIGURE 6A Displacement in X Direction

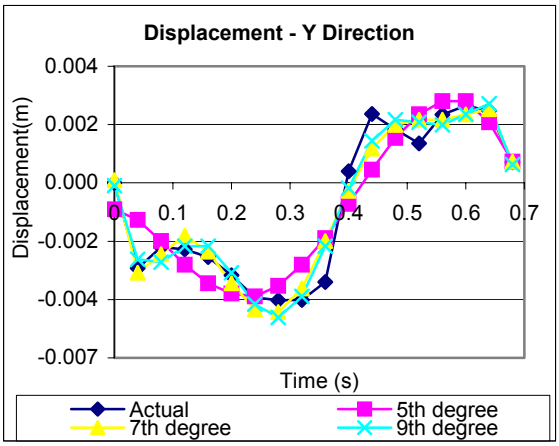

FIGURE 6B Displacement in Y Direction

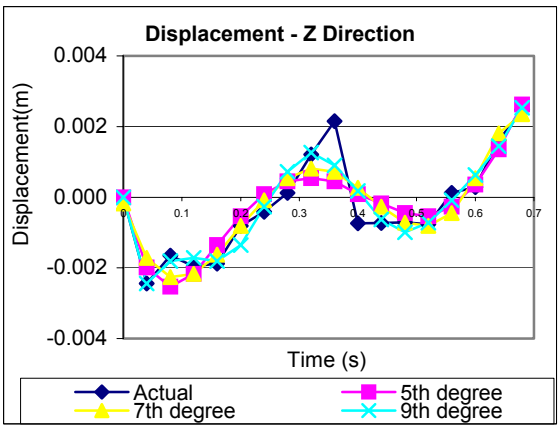

FIGURE 6C Displacement in Z Direction

Discussions

It is observed that a common pattern in the graphs analysis of accuracy of polynomial regression curve fitting technique. The level of accuracy is increasing with the increasing degree of polynomial regression. In this case, it is found that 9th degree is always more accurate than other lower order of polynomial regression.

A smoother robot path is observed after the implementation of polynomial regression curve fitting technique. Velocities and accelerations obtained from equations [2] and [3] are important for generating parameters for frames in between the key frames.

It is suggested that higher order of polynomial regression to be implemented in the future work. However, higher order of polynomial regression, require higher level of computation. It will slow down the processing power. We do not have this problem since we do the curve fitting off-line before loading into the simulator.

Conclusion

This paper addresses the development of a robot surgical system and how to obtain a smooth path of robot manipulation. The development of the robot surgical system is still in the preliminary stage but it enable surgeons to be trained where there is no risk to patients in the surgical simulations. Surgeons retain authority in surgical planning. The robot surgical system is not going to replace surgeons, but it will provide intelligent assistance with routine tasks, highly accurate, reductions in radiation exposure, and improves in training of surgeons.

The developed surgical robot is operated smoother after the implementation of polynomial regression curve fitting technique in the teach path process. Finally, it is found that 9th degree is always more accurate than 5th degree polynomial regression.

References

Madhavan, G., Thanukachalam, S., Krukenkamp, I.B. and Saltman, A. (2002), Robotic Surgeon, *IEEE Potential*, 4-7.

Chaptra, S.C. and Canale, R. P. (1998) Numerical Methods for Engineers, WCB/McGraw-Hill.

Parent, R. (2002). Computer Animation: Algorithm and Techniques, Morgan Kaufmann Publishers.

Iskandar & Taha, Z. (2003). Comparison of Motion Data from Video Cameras and Accelerometers of Human Walking, *7th Southeast Asian Ergonomics Society and 4th Malaysian Ergonomics Conference (SEAMEC)* ,334-340.

Appendix

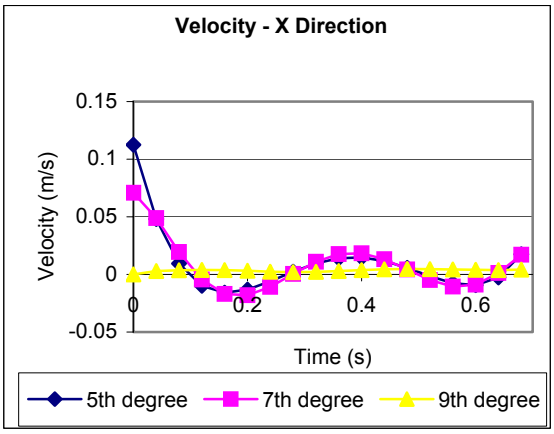

FIGURE 7A Velocity vs Time (X Direction)

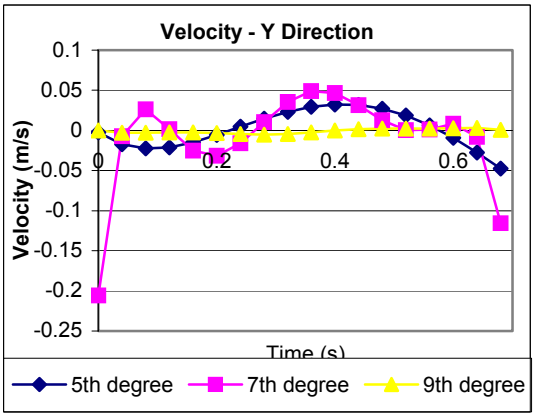

FIGURE 7B Velocity vs Time (Y Direction)

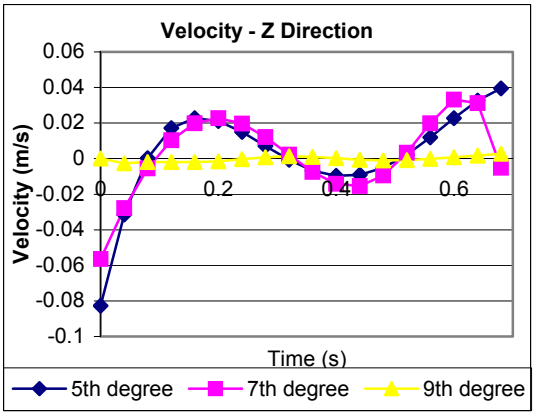

FIGURE 7C Velocity vs Time (Z Direction)

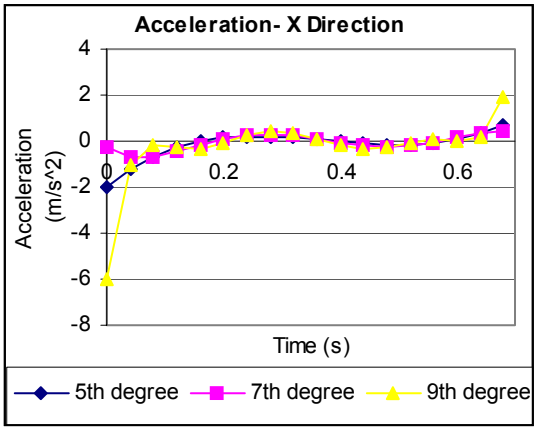

FIGURE 8A Acceleration vs Time (X Direction)

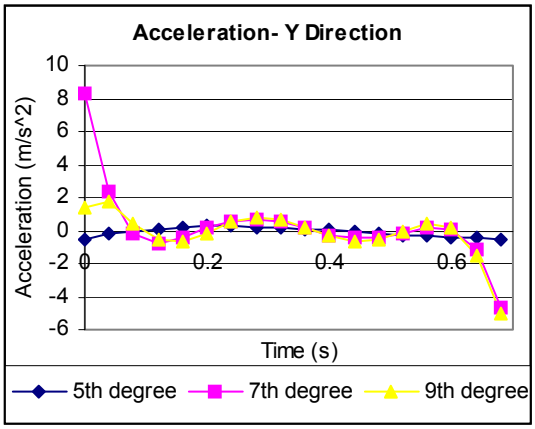

FIGURE 8B Acceleration vs Time (Y Direction)

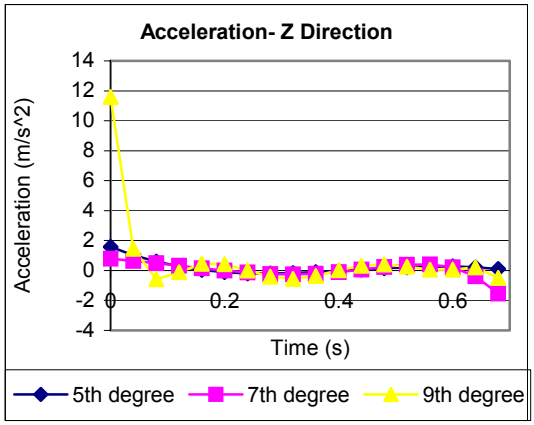

FIGURE 8C Acceleration vs Time (Z Direction)