Modelling and Simulation of the Knee Joint with a Depth Sensor Camera for Prosthetics and Movement Rehabilitation

S. Risto and M. Kallergi
Department of Biomedical Engineering, Technological Educational Institute of Athens, Athens 12243, Greece

E-mail: stamtherist@hotmail.com

Abstract. The purpose of this project was to model and simulate the knee joint. A computer model of the knee joint was first created, which was controlled by Microsoft’s Kinect for Windows. Kinect created a depth map of the knee and lower leg motion independent of lighting conditions through an infrared sensor. A combination of open source software such as Blender, Python, Kinect SDK and NI_Mate were implemented for the creation and control of the simulated knee based on movements of a live physical model. A physical size model of the knee and lower leg was also created, the movement of which was controlled remotely by the computer model and Kinect. The real time communication of the model and the robotic knee was achieved through programming in Python and Arduino language. The result of this study showed that Kinect in the modelling of human kinematics and can play a significant role in the development of prosthetics and other assistive technologies.

1. Introduction

There are millions of people worldwide living with a lower limb loss, about 2 million of them in the United States alone.[1] The main causes of a limb loss are disease, such as diabetes and peripheral arterial disease, trauma, and congenital deformities.[2] A prosthesis is a common solution to upper and lower limb amputation and it may be either below or above the joint, i.e., the elbow or the knee. The creation of a model for the knee joint and simulation of the knee movement is an essential step in prosthesis design and calibration, movement rehabilitation, or diagnosis of human knee disabilities. Such model requires a multidisciplinary approach from physiology to computer science and mechanical and electrical engineering.

Microsoft’s Kinect sensor has already been used in combination with computer vision algorithms to analyse motion patterns of healthy humans and amputees in an effort to improve prostheses design and other treatments.[3] Kinect offers low cost, computationally inexpensive motion and voice recognition capabilities with a variety of biomedical engineering applications.[4] The purpose of this work was to improve an existing physical model of the knee and lower limb and use Kinect to control the knee motion of the artificial limp remotely.

2. Methods

This study continued prior work, during which a physical model of the knee and lower limb was created and the knee motion was analysed. Current work was divided into three steps: First, the physical robotic knee joint was optimized and tested for a wider range of motion. Second, a computer model was created and controlled by the Kinect depth sensor. Third, the physical robotic knee and the computer model were combined, programmed, and tested for real time movement.

2.1. Physical Robotic Knee Joint Simulation

In previous work, a robotic lower limb was created to simulate the movement of the human knee.[5] The original knee joint included a stepper motor, an Arduino with a motor shield, two circle gears of 27 tooth each and a vertical connection with 1:1 ratio. The knee joint was connected to a
construction simulating the lower leg made of light plastic material and 45 cm in length. In this work, all parts of the original construction were optimized. The following paragraphs discuss the modifications to each element of the physical model separately and their impact on the performance of the knee model.

2.1.2. Arduino

Arduino is a modern open source platform based on a microcontroller that can be reprogrammed through C language. Arduino is suitable for interactive projects as it has the ability to receive, digital or analog data. Arduino can be extended with shields. Shields are boards with microprocessors or microcontrollers that can be added on the Arduino's main board, to match the needs of a specific project. For this project, a motor shield was used that allowed to the control of the servo motor with a simple algorithm. An interface was also developed for the communication of Arduino with a PC computer and other software through a serial port.[6]

2.1.3. Servo Motor and Worm Gears

A servo motor replaced the original stepper motor of the original knee model to take advantage of its smaller size and greater power.[5] The servo motor does not accept degrees as an input since it rotates by "reading" pulses, i.e., analog signals. Consequently, the angles need to be translated from degrees to spins and from spins to an analog type of data. The analog data are then fed into the servo motor that starts spinning in the specified direction with the specified speed. The speed of the motor depends on the power applied to it, which is proportional to the distance it needs to travel (proportional control).[7] The rotation speed and the gear ratio are used to convert degrees to spins.

Several tests were performed to determine the association between degrees of rotation and time of our servo motor. The conclusion was that the motor rotated 360 degrees in 1320 milliseconds (ms) or 1.32 s. Table 1 lists the parameters of the motor servo and the gears for seven different angles that correspond to lower leg positions. These angles were selected from published work as well as physiology data for the human knee motion.[8] Note that the gear data correspond to a worm gear set that replaced a pair of wheel gears originally used in the knee model to allow for movement of larger weights. The ratio of the worm gear is 1:X, where X is the number of teeth of the second gear attached to the worm gear (driving gear). In our case X=75 so our ratio was 1:75. Our servo motor’s torque was 32 oz-in (or 2.3 kg-cm) at 5V power supply and, consequently, its torque was 32x75=2400 oz-in (or 172.5 kg-cm). The speed of the servo motor was 39 rotations per minute (RPM).

2.2. Modeling of Knee Joint

A computer model of the knee was created with the help of Blender software. Blender is a free and open source 3D creation suite. It is a complete 3D modeling suite that supports design, rigging, animation, simulation, rendering, compositing and motion tracking, video editing and game creation.[9] Blender supports add-ons or scripts written in Python language. The designed model may be controlled by the Kinect sensor allowing for real time interaction.

2.2.1. Kinect Sensor

The Kinect (Microsoft Corp, WA, USA) sensor uses an infrared projector and a monochrome complimentary metal-oxide semiconductor (CMOS) based camera to create a depth map of an environment independent of lighting conditions. The infrared beam hits an object and its reflection is detected and processed by the CMOS to create a depth map and then infer body position allowing the recognition of movement through this process. There is also an RGB video camera recording in real time that can be used for voice recognition. Only the depth sensor was used in this study.
Table 1: Servo motor and gear measurements used to convert degrees of rotation to motor spin. The degrees correspond to various physical leg positions or knee flexions as indicated in the first column.

| Knee motion Degrees | Second Gear Number of teeth moving | Driving Gear Number of rotations | Servo Motor's Number of rotations | Time needed for servo's one full rotation (s) | Time needed for the servo to rotate (s) |
|---------------------|------------------------------------|---------------------------------|-----------------------------------|--------------------------------------------|--------------------------------------|
| 67 (normal gait/level surfaces) | 14 | 14 | 14 | 1.32 | 18.48 |
| 83 (stair climbing) | 17 | 17 | 17 | | 22.44 |
| 90 (sitting/rising from chairs) | 19 | 19 | 19 | | 25.08 |
| 100 (advanced function) | 21 | 21 | 21 | | 27.72 |
| 106 (advanced function) | 22 | 22 | 22 | | 29.04 |
| 130 (squatting) | 27 | 27 | 27 | | 35.64 |

2.2.2. Blender Model, NI_Mate & Python Scripts

The creation of a model on Blender was a relatively easy process. The control of the model directly by the Kinect sensor for motion tracking was the part that posed the greatest challenge. The control was not consistently reproducible and there were often crashes in the communication between the two interfaces. To achieve a smoother communication between Kinect and Blender, the NI_Mate program was used in combination with Blender. The NI_Mate records the coordinates of the live model from the Kinect sensor and matches them to the model's coordinates created in Blender. The result is the prosthetic leg's movement according to the live model's movement. A Python script was finally implemented to calculate and display the correspondence between the computer model's knee joint degrees and the interactive model's knee joint degrees.

The block diagram of Fig. 1 presents with images the sequence of processes involved in using live knee movement to control the robotic knee.

Figure 1: Block diagram of the processes involved in using a live model to control robotic knee movement.
3. Results & Conclusion

Based on the speed of the servo motor (39 RPM) and the time required for the motor to rotate for each movement (Table 1), the angular and tangential velocities for each of the six knee flexions of the simulated leg were calculated. The average angular velocity was 3.6 degrees/s with a standard deviation of 0.04. The tangential velocity was 1.6 m/s with a standard deviation of 0.02. Both results indicated a smooth and reproducible movement of the prosthetic leg independent of angle of rotation.

The time required for the motion information to be transferred from the live model to the computer model and then to the prosthetic knee was negligible compared to the time required to complete the flexion in one direction. The angular velocity was the same in both directions of the movement.

In conclusion, a series of processes were implemented to achieve movement of a prosthetic lower limb from a live model. The implementation was based on Microsoft’s Kinect depth camera, which, in combination with open source software communicated with a microprocessor (Arduino) embedded in a prosthetic knee to control its flexion via a stepper motor. The advantage of the current implementation is that it used low cost materials, widely available hardware, and open source software, which, however, are not always compatible. In addition to prosthetics, the proposed setup could be used for patient rehabilitation either at the hospital or at home via a remote control operation, for accurate measurements of the knee’s stress and strain, for remote control of invasive procedures.

The disadvantages of this project were the high use of data from each software and hardware with the result of crashes once a time.

References

1. *The Rehabilitation of People with Amputations*. World Health Organization, United States Department of Defense, MossRehab Amputee Rehabilitation Program, MossRehab Hospital, USA. 2004.
2. Ziegler-Graham K, MacKenzie EJ, Ephraim PL, Travison TG, and Brookmeyer R. Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050. Archives of Physical Medicine and Rehabilitation 2008; 89(3):422-429.
3. Strbac M, Kocovic S, Markovic M, and Popovic DB. Microsoft Kinect-Based Artificial perception System for Control of Functional Electrical Simulation Assisted Grasping. BioMed Res Intern 2014; Article ID 740469, 12 pages.
4. Webster D and Celik O. Systematic review of Kinect applications in elderly care and stroke rehabilitation. J Neuroeng Rehabil 2014; 11:108-132.
5. Velmahos I and Zekio M. Robotic knee model. Department of Biomedical Engineering, Technological Educational Institute of Athens. May 2014.
6. CEO, "Arduino," Arduino Organization, [Online]. Available: http://arduino.cc/en/guide/introduction.
7. http://mechatronics.mech.northwestern.edu/design_ref/actuators/servo_motor_intro.html
8. Chakravarthy YK, Kumar RH, Deepika B, and Srinath A. Haptics based robotic prosthetic leg. Advances in Robotics, Mechatronics, and Circuits. ISBN: 978-1-61804-242-2.
9. Blender Foundation, "Blender," Blender Organization, [Online]. Available: http://www.blender.org/features/.