Modelling and simulation of SISO model prosthetic limb by Open Modelica an acausal and cyber physical modelling language

Sachu Reji, Sanu K Sibichan, Vargheses Lejo Raju, V Venkatesh and T K Madhukumar

Department of Mechanical Engineering, Rajiv Gandhi Institute of Technology, Kottayam, Kerala, India

Abstract. Design of cost effective transradial prosthesis has been in the research field for some time. Instead of complex systems with less dexterity, we put forth a design considering solely the wrist movement of the prosthetic hand, driven by a flex sensor-Arduino system which translates the motion of natural hand to the prosthetic one. Objective of this paper is to conduct a simulation experiment of this design, using Open Modelica, by creating a virtual lab environment. This paper describes the present state of the research, simulation along with the results obtained and the conclusions we could reach at this period in comparison with the initial hypothesis.

1. Introduction
A person with a limb amputation faces various problems in their daily life and most do not have the fortune of affording the modern prosthetic. Transradial limb amputees constitute around 60% of the total limb amputation [1]. A need for low cost transradial prosthetic was recognized and was worked on. The design thus developed was based on a novel theory in the field of prosthetics, called mirror theory, which allowed the motion of the natural (master) hand to control the prosthetic (slave) hand [2]. Hence, a 3D printed prosthetic limb with an artificial wrist movement, as shown in Fig. 1, which allows grabbing motion at any position of the forearm, was developed. The model is a modification, with official collaboration, of the wrist amputee model from E-Nable to make it functional for transradial [3]. The actuation of the slave hand is through the movement of the master hand, which is detected with the help of a ex sensor, as shown in Fig. 5, and eventually sent to the motor input of the slave hand via a microprocessor with a simple PID feedback system [4]. Open Modelica, the tool used in this research, is a high level object oriented software for modelling and simulation that stands apart from its parallels with the use of component based modelling as opposed to the block diagram based modelling that allows it to be much simpler to control [5]. Being an acausal method of modelling, Modelica is not governed or operated by the laws of cause and effect [6]. Thus, Open Modelica could be studied extensively by understanding its history, advantages over other comparable tools, and most importantly, with a physical example of a project which would explain the tool in a much wider sense, so that in the future, it could be helpful for students or professionals in their projects and bringing this open source tool to the mainstream.
2. System design model

The design is based on the working that the amputees natural (master) hand acts as the mirroring body to the prosthetic (slave) hand of the design. It is made possible through an actuating mechanism that would replicate the wrist motion of the master hand. The entire system has been designed in such a way that when the prosthetic hand produces the wrist movement of flexion or extension, the finger grabbing motion will simultaneously be produced along with it. It is made in such a way that it does not act as a full link but rather only the wrist. A gear mesh is the base of the mechanism where the worm gear and spur gear are the main parts, as shown in Fig. 2. A worm gear has been placed which receives input from the servo motor placed behind it. The servo motor receives information from an Arduino Uno which in-turn receives the information from the Master hand. The servo motor when actuates the worm gear, the worm of the system starts to move which will thereby produce a speed reduction to the worm wheel. The worm wheel is connected to two spur gears through its shaft. These two spur gears are mated with the spur gears on the palm of the slave hand which thus produces the movement and hence mechanism starts working. When the spur gear starts to rotate and the component starts to move, it will replicate the motion of a wrist flexion-extension. This wrist movement is now directly linked with a finger grabbing movement. The design of hand is in such a way that different joints in the hand acts together to produce the grabbing motion. The proposed reduction mechanism with prosthetic hand is shown in Fig. 3. Tendon strings have been used to produce this motion. The strings are fixed at the tip of the fingers and are guided through the joints to the back of the palm where it is set. The mechanism simply works in such a way that when the string is pulled, the finger starts to go to the direction of the pull and due to the structure of the joints; it will form a grabbing motion. The system when worked in its structure will control the position of the slave hand with respect to the angle change from the master hand. The control system as shown in Fig. 4, works with the actual angle being taken from the master hand through a ex sensor, sent to the
microprocessor where it is taken as pulse width, amplified and sent to the motor on the slave hand. The signal is then encoded and a feedback is sent with the error signal processed and controlled. The entire PID is controlled through program in the microprocessor.

3. Open Modelica: Mixed domain modelling model

Modelling and Simulation mostly takes place in signal orientated programs like Matlab/ Simulink. For some years the object-orientated modelling language Modelica has become well known. The program created by Hilding Elmqvist provides a different approach to modelling where the simulation program, as far as possible, builds up the signal structure and manipulates the needed equations with a claim to make modelling much easier and more systematic. The main objective is to make it easy to exchange models and model libraries. The design approach builds on non-causal modelling with true ordinary differential and algebraic equations and the use of object oriented constructs to facilitate reuse of modelling knowledge [5]. It is a high level object oriented modelling which is component based as opposed to other block diagram based programs [6]. The language allows defining models in a declarative manner, modularly and hierarchically and combining various formalism expressible in the more general Modelica formalism. The multi-domain capability of Modelica allows combining electrical, mechanical, hydraulic, thermodynamic, etc., model components within the same application model [10]. This language others several important advantages, they are:

Object-oriented mathematical modelling: The object-oriented modelling makes it possible to create model components, which are employed to support hierarchical structuring, reuse, and large and complex models covering multiple technology domains.

Acausal modelling: The acausal modelling is based on mathematical equations instead of assignment statements as in traditional input/output block abstractions. The equations increase re-usability of model components. A simple Modelica program is shown below.

```modelica
Class HelloWorld
  Real x(start = 1);
  parameter Real a = 1;
  equation
  der(x) = -a*x;
end HelloWorld
```

Physical modelling of multiple application domains: Model components can correspond to physical objects in the real world. The physical components are particularly easy to combine into simulation models using a graphical editor.

4. Open Modelica: Mixed domain modelling model

In Modelica, users can use Modelica Standard libraries of components to produce more complex models [13]. Extensive libraries are available in the web. Each physical component has icons, and each connector represents a physical connection. The Modelica translators convert Modelica object oriented models into flattened models.

4.1. Single link manipulator mathematical model

The plant of the present paper is a DC servo motor and gear mechanism. These electrical and mechanical domains are modelled by component modelling. Even though the modelling is based on graphical method, the mathematical model of the present paper is summarized below. The single-input, single-output (SISO) control system approach is presently adopted by most of the industrial robot suppliers [8]. A gear mesh is the base of the mechanism where the worm gear and spur gear are the main parts. The prosthetic (slave) hand is the one - link manipulator being driven by a DC servo motor through a worm gear and spur gear reducer. The prosthetic hand rotation is denoted by \( \theta_L \) and that of the motor by \( \theta_m \). The overall speed ratio, SR, of the reduction mechanism is taken as 130. The
The J_m, f_m, and T_m are the inertia, viscous friction and torque of the motor, respectively given in Table 1

### Table 1. DC motor armature parameter (PITTMAN 9413) [8]

| Terms | Description                     | Value       |
|-------|---------------------------------|-------------|
| R_a   | armature resistance             | 5.35 ohm    |
| L_a   | armature inductance             | 3.93 mH     |
| J_m   | moment of inertia of rotor      | 2.75 ×10^6 kgcm^2 |
| K_t   | motor torque constant           | 0.0316 Nm/A |
| K_b   | back emf constant               | 0.0316 V rad/s |
| T_m   | motor torque                    |             |

The J_L, f_L, and T_L are the inertia, viscous friction and torque of the link and driven gear, respectively as given in Table 2. The equation of motion of one-degree of freedom link as [8,9].

\[
(J_m + n^2 J_L)\dddot{\theta}_L + (f_m + n^2 f_L)\dot{\theta}_L = T_m + n T_L
\]  

### Table 2. Gear and load parameters

| Terms | Description                     | Value       |
|-------|---------------------------------|-------------|
| J_L   | moment of inertia of load       | 2.75 ×10^6 kgcm^2 |
| f_b   | bearing friction load           | 0.25 kgcm^2 |
| T_L   | gear and external load torque   | 5.905 ×10^6 kgcm2 |
| SR    | Speed ratio of gear train       | 130         |
| N     | gear ratio of gear train        | 1/SR        |

### 4.2. DC servo motor modelling

Figure 9 shows the circuit diagram of a permanent-magnet DC servo-motor. The parameters for the DC motor are shown in Table 1. The dynamics of the circuit can be modelled by a first-order differential equation as [9]:

\[
L_a i_a + R_a i_a + K_y i_a = V_a
\]  

### 5. Block model: Algorithm models

#### 5.1. Flex circuit Algorithm Model

The Flex sensor is a variable printed resistor that produces a resistance output correlated to the bend angle [11]. The resistance increases with bend angle. The sensor FS-L-0095-103-ST has 10 kΩ at resistance and bends resistance of minimum 2 times greater than the flat resistance at 180° pinch bend.
Assuming linear variation of flex sensor resistance with master hand angular position, the flex resistance as a function of angle is obtained by Eq. 3.

\[ R_{\text{flex}} = a \times \theta_{\text{hand}} + b \]  

(3)

Here constant ‘b’ is the flat resistance and constant ‘a’ is obtained from maximum resistance. The basic Flex Sensor Circuit as shown in Figure 5, is a single sided operational amplifier, used with exc sensor, R_{flex} and R_2 form voltage divider. The algorithm model of Flex circuit is shown in Figure 9. The output voltage of operational amplifier circuit is given by Eq. 4. Master hand angular position vs. op amp output is shown in Fig. 6.

\[ V_o = V_{\text{ref}} \frac{R_2}{R_2 + R_{\text{flex}}} \]  

(4)

Figure 5. Flex sensor with op amp circuit

Figure 6. Master hand angular position vs. op amp output

Figure 7. Op amp output voltage vs. analogRead()

Figure 8. analogRead() vs. mapped analogRead()

5.2. Flex circuit algorithm model

The Arduino Uno is a micro controller board based on the ATmega328 microprocessor. The Arduino Uno can be programmed with the Arduino software similar to C language. In the loop section of the algorithm, the reading of the flex sensor voltage in the range of 3.16 to 5 voltage is converted to digital signals by ADC using analogRead() function which is in the range from 647 to 1021 as shown in Figure 7. The analogRead() value of the Arduino uno is in the range of 0 to 1023. So for getting maximum range of output voltage, the analogRead() value in the range of 647 to 1021 is mapped 0 to 1023, as shown in Fig. 8. The analogRead() value in the range of 0 to 1023 is then converted to DC equivalent pulse width modulated (PWM) signal by analogWrite() function. The analogWrite() function requires the value in the range of 0 to 255. So it is required to map this analogRead value in the range 0 to 255, which is shown in Fig. 10. This value is equivalent to DC voltage at output pin of the Arduino uno. The actual output voltage equivalent of PWM signal which can be obtained from duty cycle of the PWM signal is shown below.

PWM, or pulse width modulation is a technique which allows adjusting the average value of the voltage that is applied to the PID controller. For controlling the rotation direction, we need to inverse the direction of the current flow through the motor. One of the methods to do so is by using an H-Bridge. An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the
motor at the centre forming an H-like configuration. In this paper we are interested in the performance of the control system; therefore, H-Bridge circuit is not implemented in this algorithm model. The algorithm model of Arduino uno is shown in Fig. 9.

Figure 9. Hierarchical model of the closed-loop prosthetic limb system, including components such as DC servo motor, single-link manipulator gear drive system, bearings, control software inside the Arduino uno block, ex sensor, operational amplifier, PID, position feedback, and external load.

5.3. **PID controller algorithm model**

The parameters of the PID controller are manually adjusted by performing simulations of the closed loop system and using the following strategy [12]. Manually adjust parameters k, Ti, Td (time constant of derivative block) and perform simulations such that the output of the PID controller goes in its limits. Tune Ni (Ni × Ti is the time constant of the anti-windup compensation) such that the input to the limiter block goes quickly enough back to its limits. The PID algorithm model is shown in Fig. 9. The output voltage of PID controller is shown in Fig. 11.

5.4. **Feedback algorithm model**

The feed-back system measures the absolute angle \( \theta \) of the link in an ideal way by Angle Sensor and provides the result as output signal. This angle is fed back to Arduino uno. The angle is then converted by ADC analogRead() function to the range of 0 to 1023. This signal then converted to the range of 0 to 255 by analogWrite() function as shown in Fig. 9. The Controller library contains input/output blocks, PID, feedback etc. [13]. The feedback compares the signal from ex circuit and Angle Sensor.
and produces the error signal. This error signal (PWM signal) is the output of the Arduino Uno and is applied to PID controller.

5.5. External Loading Algorithm Model
A constant torque of 5.905 N-m applied on the prosthetic limb which is modelled as block as shown in Fig. 9.

6. Result and Discussion
The current research is done purely as a simulation experiment avoiding the design of control system by Laplace transform. The values given to the controller system is by trial and error method from which the PID gain is found. Moreover, being at the simulation stage, experimental validation is not available for accurate motor parameters or a friction model. The following PID parameters are first selected: total gain of the controller, $K = 5$, the time constant of the integrator, $T_i = 0.01$ and time constant of derivative block, $T_d = 0.01$. The simulation result of required angular position is compared with desired angular position as shown in Fig. 12. The result shows some oscillation, so the PID is tuned by trial and error method. The following values give reasonable performance output of the control system as shown in Fig. 13. The tuned PID parameters are: total gain of the controller, $K = 15$, the time constant of the integrator, $T_i = 0.01$ and time constant of derivative block, $T_d = 0.1$. The overall simulation experiment has been found to be in good control; however a slight variation is noted from the desired angular position. The primary sought-after result was to obtain 0 to 90 rotations in 1 second, but as observed, the result gained is leading or overshooting from the desired one. Furthermore, it is inferred that the inaccurate friction model, which causes damping, and assumption of ideal gear might have affected the result.

7. Conclusions
The discussed model was aimed to be cost effective and available to whomever the need is from. As much as it is important, accuracy of the design was put secondary. Learning from our simulation, the aim moving forward would be experimentation which could provide experimental validation to help attain accurate parameter values and friction model that could provide a more precise simulation. Future scope of the model could be to focus on improving the design at a multi objective stage. Keeping our cost effective design as a base, research could be made to better the design through weight reduction or improving motion by increasing the degree of freedom.
References
[1] Belter J T, Segil J L, Dollar A M and Weir R F 2013 Mechanical design and performancespecifications of anthropomorphic prosthetic hands: a review Journal Rehab. Res. Dev. 50(5) 599
[2] Kehli S, Scharfenberger T, and Dietz V 2020 Coordination of bilateral synchronous and asynchronous hand movements. Neurosci Lett. 720 134757
[3] https://enablingthefuture.org/
[4] Vyawahare V M and Pardhi D 2017 Design of a prosthetic arm using flex sensor International Journal of Electronics and Communication Engineering and Technology 8(2) 01
[5] Elmqvist H and Mattsson S E 1997 An introduction to the physical modeling language modelica Proceedings of the 9th European Simulation Symposium 1(1) 1
[6] Fritzson P and Thiele B 2016 Introduction to Object-Oriented Modeling, Simulation, Debugging and Dynamic Optimization with Modelica using OpenModelica https://www.openmodelica.org/doc/ModelicaShortCourse/
[7] Virgala I, Frankovsky P and Kennderova M 2013 Friction effect analysis of a DC motor American J. of Mechanical Engineering 1(1) 1-5
[8] Craig J J 2003 Introduction to robotics mechanics and control, Pearson Education (Singapore), India, Fourth Edn.
[9] Ghosal A 2009 Robotics fundamental concepts and analysis, Oxford University Press, New Delhi Fourth Edn.
[10] Fritzson P 2011 Modelica-A cyber-physical modeling language and the OpenModelica environment Wireless Communications and Mobile Computing Conference 1648-1653
[11] https://www.spectrasymbol.com/product/ex-sensors/
[12] https://build.openmodelica.org/Documentation/Modelica.LinearSystems2.Controller.LimPID.html
[13] Baur M, Otter M and Thiele B 2009 Modelica Libraries for Linear Control Systems Proceedings 7th Modelica Conference Como, Italy, Sep. 20-22 043 593-602