DESIGN AND CONTROL OF AN ACTIVE PROSTHETIC LEG

Hoang Trung Ngo¹, Thien Duc Ngo², Danh Ngoc Nguyen³, Hoai Nam Le³

¹FPT Global Automation; hoangtrungngovan.td@gmail.com
²R&P Logistics Company Limited; ngothienduc94@gmail.com
³University of Science and Technology, The University of Danang; ndngoc@dut.udn.vn, lehoa@nam@dut.udn.vn

Abstract - The purpose of this paper is to present the design, prototype and control of an active prosthetic leg. The research scope encompasses the mechanical design, the electronic design and the control system design. The mechanical design covers kinematic diagram design, dynamic system modelling and 3D design. Meanwhile, the electronic one includes PCB mainboard design and sensor system design. The control system design consists of system parameter estimation and self-tuning regulator adaptive controller design, creating the adaptive controller which plays a role as the brain of the prosthetic leg to control the whole system. The main objective of this work is to produce a very first prototype of an active prosthetic leg which mimics the movement, experimenting on a healthy man before being adopted on amputees. In addition, the prosthetic leg can be also served as a test platform for testing the control methods.

Key words - Active prosthesis; control design; adaptive control; leg trajectory; communication protocols.

1. Introduction

In order to improve the mobility of people with limb amputation, intensive research activities on the prosthetic field have been developed during the last decades. Various prosthetic designs were introduced which can be divided into two kinds of devices: passive and active. Passive prosthetics require the user to move it with their own effort [1]. This can make the disabled people feel uncomfortable. In such a case, different kinds of active prosthetic with energy sources and auxiliary sensors were developed and commercialised [2]. The prototype SmartLeg was introduced [3] which applies machine learning to provide optimal gait to the user by active movement controlling in knee and ankle joints through embedded hydraulic actuators. [4] presents the mechanical design and controller design of a semi-active above-knee. Somes different kinds of control algorithm for active prosthetic leg were proposed [5, 6, 7]. Current active prosthetics such as the Ottobock C-Leg offer a high level of comfort for patients, but most of them have a high price for most customers, especially ones in Vietnam. The objective of this work is to develop a flexible active above-knee prosthetic leg model with a considerable lower cost. To this aim, a prototype with a sophisticated mechanism of above-knee [8] was built with 3D printing technology. For the control purpose, a novel leg’s trajectory construction method and an adequate adaptive control design are proposed. Besides, a variety of system modelling methods, electronic interface protocols, PCB design as well as mechanical design were brought into the leg. The developed model can be also served as a platform for testing and applying the control methods, notably for mechatronic students.

The remainder of this paper is organised as follows. Section II is devoted to the description of the system design. Section III describes the characteristics of the built prototype. Some numerical results are also discussed here. Finally, the conclusion drawn from this work and possible ways for further studies are given.

2. System design

This section describes the overall structure of the designed prosthetic leg and its control system.

2.1. General structure

Figure 1 presents the active prosthetic leg overall structure which includes two principal parts: offline and online process. The online process is the indispensable task as in every active prosthetic. Firstly, the sensors acquire information about the leg’s state, then convert it into the data which is transferred to the set point generator. The generator compares the data with the leg’s trajectory dictionary, produces set points for above-knee and ankle and then converts them into screw nut positions. Based on these desired positions, the controller controls the prosthetic system. Besides, different states of the leg would have a different need of force because of various loads; therefore, the controller must use the data from sensors to decide how much energy should be used for the contemporary state.

In addition, because the human leg has an untold number of states, an offline process must be introduced to undertake this crucial task, collecting the various states of the human leg to conduct a leg’s trajectory. This process is done using motion tracking sensor system attached to the human leg to detect the angles of thigh, calf and foot.

2.2. Mechanical design

In order to build a prosthetic leg, a kinematic diagram was introduced. As illustrated in Figure 2, the prosthetic leg can be divided into two main mechanisms: the above-knee and the ankle. Besides, two axes were put into the diagram in order to calculate the angles of thigh, calf and foot.

In the human leg, the flexible of above-knee comes from ligaments, but the contact place of two bones and muscles are responsible for the body’s load [9]. By the way of changing
the shape, the quadrilateral structure plays the same role as ligaments in the above-knee mechanism. In addition, a cam was integrated into the quadrilateral structure to act as the contact place of two bones to strengthen the load bearing capacity. Finally, the screw mechanism and timing belt with motor operate like muscles to control the leg’s movement in a considerably adequate way.

**Figure 2. Kinematic diagram of the prosthetic leg**

In terms of movement, the angle between the calf and the foreshortened leg is a small range (less than 15 degrees). In terms of dimension, the calf is far longer than the thigh. From these points, without changing the whole structure of the kinematic diagram, the combination structure of quadrilateral structure and calf can be completely simplified to become a pivot with its heart located in the intersection of the thigh and calf (of the quadrilateral structure).

**Figure 3. Brief kinematic diagram: (a) the lower leg part with ankle and (b) the higher leg part with above-knee**

In order to determine the leg’s movement, both movement sensors and encoder sensors (which track the movement by evaluating the position of screw nuts) can be used. So, how to evaluate the exact movement of the leg by encoders? In fact, encoders can only indirectly calculate the position of screw nuts by determining the rotation of motors. At the same time, we must have some functions to transform the desired knee’s angle and ankle’s angle into the desired position of screw nuts. Figure 3(a), 3(b) and the following functions have met these urgent needs.

**Table 1. The dimension of prosthetic leg’s prototype**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_1 )</td>
<td>Angle between calf and foot</td>
</tr>
<tr>
<td>( E_1 )</td>
<td>Distance between the screw 1 and the pivot</td>
</tr>
<tr>
<td>( F_1 )</td>
<td>Distance between the pivot in ( E_1 ) and ankle</td>
</tr>
<tr>
<td>( G_1 )</td>
<td>Distance between two pivots of foot</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>Distance between the pivot and the screw nut 1</td>
</tr>
<tr>
<td>( H_1 )</td>
<td>Distance between the pivot in foot and ( E_1 )</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>Distance between the screw nut 1 and position switch</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>Angle between thigh and calf</td>
</tr>
<tr>
<td>( E_2 )</td>
<td>Part of prosthetic thigh to fasten the pivot</td>
</tr>
<tr>
<td>( F_2 )</td>
<td>Part which sticks to the remnant human thigh’s part</td>
</tr>
<tr>
<td>( G_2 )</td>
<td>Distance between two pivots in the higher leg part</td>
</tr>
<tr>
<td>( H_2 )</td>
<td>Distance between the screw 2 and the pivot</td>
</tr>
<tr>
<td>( L_2 )</td>
<td>Distance between the pivot and the screw nut 2</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>Distance between pivot in calf with ( H_2 )</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>Distance between the screw nut 2 with position switch</td>
</tr>
</tbody>
</table>

Figure 3(a) includes:

\[
\theta_1 = 2 \arccos \left[ \frac{F_1^2 + G_1^2 - H_1^2 - E_1^2}{2 F_1 G_1} \right] + 90 \tag{1}
\]

\[
\theta_1 = (\text{calf's angle} + 90) + (180 - \text{thigh's angle}) + 180
\]

By using the data from movement tracking sensors, we have the calf’s angle and the thigh’s angle. From (1), we can completely calculate \( H_1 \). \( x_1 \) can be then calculated using the following equation:

\[
x_1 = H_1 - L_1 \tag{2}
\]

Like the Figure 3(a), Figure 3(b) includes:

\[
\alpha_2 = \arccos \left[ \frac{K_2^2 + F_2^2 - E_2^2}{2 K_2 F_2} \right] + \arccos \left[ \frac{K_2^2 + G_2^2 - I_2^2 - H_2^2}{2 K_2 G_2} \right] \tag{3}
\]

\[
\theta_2 = \text{thigh's angle} + 180 - \text{calf's angle}
\]

For more details, \( K_2 = \sqrt{E_2^2 + F_2^2 - 2 E_2 F_2 \cos \alpha_2} \), and \( x_2 \) can be then calculated by the following equation:

\[
x_2 = I_2 - L_2 \tag{4}
\]

\( x_1 \) and \( x_2 \) can be used for controlling the prosthetic leg’s states. To mimic the movement of a real leg, a leg’s trajectory or phases of human gait cycle [1,10] must be built in the offline process.

The self-tuning PID controller is embedded in dsPIC33 will calculate and conduct voltage value to control the motor’s rotation through the driver according to the position of screw nuts. The encoder data feedback is used to ensure these positions are accomplished. Apart from that, an RF module is used to send data of prosthetic leg to another CPU for visualizing and building the initial transfer functions.
2.3. Control system design

Figure 4 describes the prosthetic leg’s system architecture, including control algorithm and sensors. The motion tracking sensor system measures the contemporary state of the leg. Based on this state, the last nearest state of leg and the leg’s trajectory dictionary, the setpoint generator produces the desired calf’s angle and the foot’s angle. These desired angles can be then converted into the positions of screw nuts using the equations from (1) to (4).

\[ y_k = a_1 y_{k-1} + a_2 y_{k-2} + b_1 u_{k-1} + b_2 u_{k-2} \]  \hspace{1cm} (6)

Parameters vectors can be easily updated by using recursive least squares filter [11,13]:

\[ \hat{\theta} k = \hat{\theta} k-1 + L_k \varepsilon k \]  \hspace{1cm} (7)

\[ \varepsilon k = y_k - \varphi^T k \cdot \theta k-1 \]  \hspace{1cm} (8)

\[ L_k = \frac{P k-1 \varphi^T k}{\lambda + \varphi^T k P k-1 \varphi k} \]  \hspace{1cm} (9)

\[ P k = \frac{1}{\lambda} [I - L_k k \varphi^T k] P k-1 \]  \hspace{1cm} (10)

With:

\[ \varepsilon k = a_1 k-1 \cdot y k-1 + a_2 k-1 \cdot y k-2 + y k - b_1 k-1 \cdot u k-1 - b_2 k-1 \cdot u k-2 \]  \hspace{1cm} (11)

\[ \varphi k = \begin{bmatrix} -y k-1 \\ -y k-2 \\ u k-1 \\ u k-2 \end{bmatrix}^T \]  \hspace{1cm} (12)

For adapting the rapid dynamic of the system, a self-tuning PID controller is adopted. At each instant, the setpoint generator uses the measurement of prosthetic leg and the leg’s trajectory dictionary to determine the set-point (angles of thigh, calf and foot). Using the estimated prosthetic leg transfer function, a self-tuning regulator is used to change the parameters of the PID controller. The control system can be illustrated by Figure 4.

According to (5), open system transfer function illustrates the relationship between the position of screw nut \((x_1, x_2)\) and the PWM value:

\[ G_p z = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} \frac{T}{z-1} I_z \]  \hspace{1cm} (13)

With: \(T\) – sample time, \(t_s\) – dimension of the screw’s pitch.

To control the screws, a PI controller is clearly simple and sufficient, with the controller’s transfer function [11,13]:

\[ G_c z = K_p + K_i t_s \frac{z+1}{2(z-1)} \]  \hspace{1cm} (14)

The characteristic equation of the closed loop:

\[ 1 + G_c z \cdot G_p z = 0 \]  \hspace{1cm} (15)

From (13), (14) and (15) we have:

\[ 2 \cdot K_p z^3 + \begin{bmatrix} 2 a_1 - 1 \cdot K_p + T \cdot a_1 + 1 \cdot K_i \end{bmatrix} z^2 + \begin{bmatrix} 2 a_2 - a_1 \cdot K_p + T \cdot a_2 + a_1 \cdot K_i - 2 \cdot T t_s b_1 \end{bmatrix} z + a_2 \cdot T \cdot K_i - 2 \cdot a_2 \cdot K_p - 2 \cdot T t_s b_2 = 0 \]  \hspace{1cm} (16)

Open system (13) is a third order strictly proper transfer function with a pole of \(z = 1\). According to [13], the further the pole of the system is from the imaginary axis, the smaller the time constant and the faster the time responds to the system. As a result, the system can be approximate to the second order system. In this research, the quality control includes:

Damping constant: with the ITAE standard, the second order system should completely have a damping constant \(\xi = 0.707\).

Natural oscillation frequency: according to 5% standard, with the average setting time of 1s:
Choose \( \omega_n = 5 \) (rad/s)

From (16) the characteristic equation is a third order one. In addition, this equation must have two conjugate poles corresponding to \( \xi = 0.707 \) and \( \omega_n = 5 \):

\[
z + a \ z^2 + 2\xi\omega_n z + \omega_n^2 = 0
\]

\[
\Leftrightarrow z + a \ z^2 + 2\times0.707\times5z + 5^2 = 0
\]

\[
\Leftrightarrow z^3 + 7.07 + a \ z^2 + 2 \times 7.07a + 25 \ z + 25a = 0
\]

From (16) and (17), we have:

\[
\begin{align*}
2 \ a_1 - & 1 \ . K_P + T \ a_1 + 1 \ . K_I \\
= & \ 7.07 + a \ 2 \ . K_P + T \ . K_I
\end{align*}
\]

\[
\begin{align*}
2 \ a_2 - & a_1 \ . K_P + T \ a_2 + a_1 \ . K_I - 2 \ . T \ . t_1 \ . b_1 \\
= & \ 25 + 7.07a \ 2 \ . K_P + T \ . K_I
\end{align*}
\]

\[
\begin{align*}
-2 \ a_2 \ . K_P + T \ a_2 \ . K_I - 2 \ . T \ . t_1 \ . b_2 \\
= & \ 25a \ 2 \ . K_P + T \ . K_I
\end{align*}
\]

From 7.07\times(18)–(19) and 25\times(18)–(20), we eliminate the variable \( a \):

\[
\begin{align*}
16.14 \times a_1 - 2a_2 - 64.14 & \ . K_P \\
+ T \ 6.07 \times a_1 - a_2 - 17.93 & \ . K_I = - 2 \ . T \ . t_1 \ . b_1
\end{align*}
\]

\[
\begin{align*}
50a_1 + 2a_2 - 403.5 & \ . K_P \\
+ T \ 50a_1 - a_2 - 126.75 & \ . K_I = - 2 \ . T \ . t_1 \ . b_2
\end{align*}
\]

Set:

\[
\begin{align*}
A & = \begin{bmatrix} 16.14 \times a_1 - 2a_2 - 64.14 & T \ 6.07 \times a_1 - a_2 - 17.93 \\
50a_1 + 2a_2 - 403.5 & T \ 50a_1 - a_2 - 126.75 \end{bmatrix} \\
B & = \begin{bmatrix} - 2 \ . T \ . t_1 \ . b_1 \\
- 2 \ . T \ . t_1 \ . b_2 \end{bmatrix}, \quad X = \begin{bmatrix} K_P \\
K_I \end{bmatrix}
\end{align*}
\]

We have the equation: \( AX = B \)  

Finally:

\[
X = \begin{bmatrix} K_P \\
K_I \end{bmatrix} = A^{-1}B
\]

In conclusion, corresponding to system, parameters have been continuously estimated and the PI controller’s parameters \( K_P \) and \( K_I \) are also updated by using the equation (24).

3. Implementation and result

3.1. 3D prosthetic leg prototyping

The entire prosthetic leg is fabricated from plastic using 3D printing technology. A sophisticated flexible mechanism which consists of a cam and a quadrilateral structure was used for imitating the above-knee, called “remotion knee” [8]. Figure 5 illustrates the 3D model in CAD software, the real prototype and the real prototype attached to a human body for operational testing.

![Figure 5. 3D model, real prototype with motion tracking sensors, real prototype attached on human](image)

The prototype hardware configuration is resumed in Figure 4. The motors are high-torque DC brushed motors with a peak operating speed of 468 RPM. To measure the angle of the thigh, we use an InvenSense MPU6050 sensor [14, 15] which contains a MEMS accelerometer and a MEMS gyroscope in a single chip. The foot and calf angles are deducted by measurements of encoder integrated into two motors which move the foot and the calf joints. To control the motors, to process the data from sensors and to perform the self-tuning PID controller, a 16-bit PIC microcontroller is used [16].

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Dimension</th>
<th>Symbol</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1 )</td>
<td>17.0 mm</td>
<td>( E_2 )</td>
<td>70.0 mm</td>
</tr>
<tr>
<td>( F_1 )</td>
<td>136.0 mm</td>
<td>( F_2 )</td>
<td>35.5 mm</td>
</tr>
<tr>
<td>( G_1 )</td>
<td>35.4 mm</td>
<td>( G_2 )</td>
<td>200.2 mm</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>94.4 mm</td>
<td>( H_2 )</td>
<td>25.0 mm</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>90.0°</td>
<td>( L_2 )</td>
<td>126.4 mm</td>
</tr>
</tbody>
</table>

The PCB was designed by Altium designer software, including the main dsPIC33 microcontroller as in Figure 6.

![Figure 6. The main controlling board of prosthetic leg](image)
ones (e.g. resistive sensors) or must be the absolute encoders. The former type will be gradually eroded, while the encoders are relatively hard to assembly into prosthetic knee without bloating the knee’s volume. Besides, the encoders with an adequate precision are relatively expensive, and it is hard enough to measure the angle of knee joint because of the complicated structure of “remotion knee”. So, we should use a motion tracking sensor system to track the thigh, calf and foot’s angles, then convert it to the angle of hip, knee, ankle joint angles. After that, we transform this relationship into the position of the screw nuts. In order to ensure the accuracy of the converting process, we utilise 3D simulation (SolidWorks) before manufacturing using 3D printing technology.

Figure 8 shows how the data of leg’s movement was collected (angles of the thigh, the calf and the foot) by a motion tracking sensor system (Figure 7) using InvenSense MPU6050 sensors and Arduino.

There are a few stages of investigating and transforming the data to a dictionary have been using in leg’s controller. Firstly, data was collected at least three times each type of topographic. Next, Microsoft Excel was used to illustrate data as a line graph showing the relationship of three angles of leg in comparison with the axes (thigh, calf, foot).

The obtained measurements of leg’s movement can be represented by the angles’ relation graph (Figure 9). The angles in this graph were adjusted to make the angles of thigh, calf and foot have the same value of -90 degree as the patient state’s standstill. The data analysis bases on the different states of value in Figure 9 rather than the human gait. As seen in Figure 9, we can divide the data into 4 phases based on the value of thigh’s angle and its velocity as well as calf’s angle velocity. These 4 stages can be then easily converted to a continuous function, much more easy to implement to the microcontroller (MCU) than electromyography method [5, 9], and still gives the leg a relatively trusty trajectory.

This relationship can then be transformed to the relationships of \( \theta_1 \) and \( \theta_2 \) (in Figure 2) with the thigh’s angle and calf’s angle. Next, these correlations can be converted to the positions of screw nuts, and the two motors will do the rest of tasks to provide the prosthetic leg with an adequate movement.

3.3. Initial transfer function of the system

At the initial state of the prosthetic leg, when the human state is a standstill, we collect the data and estimate the initial values for the systems:

Above-knee transfer function:

\[
G_{V_1}(z) = \frac{Y(z)}{U(z)} = \frac{-0.06068z + 0.072}{z^2 - 0.3656z - 0.4994} \quad (25)
\]

Ankle transfer function:

\[
G_{V_2}(z) = \frac{Y(z)}{U(z)} = \frac{-0.1318z + 0.1463}{z^2 - 0.5266z - 0.2994} \quad (26)
\]

These transfer functions play a role as initial values of transfer functions of the prosthetic leg, and the parameters will be continuously updated by using recursive least squares filter in 2.3.

3.4. The performance of the self-tuning PI controller

Figure 10 illustrates the performance of the prosthetic leg with sample time of 50ms. There are some disorders in the first 5s of the operation process because of the leg initializing.

With the average time instant as 1s, the performance of the above-knee is adequate with the average static error of only over 10%. The first state of disabled human’s legs is standing, so the errors tend to be larger than 0.
The performance of ankle has a quite large average static error (27%) but the error is considerably stable. The performance is probably acceptable because the static error must be accepted to maintain the steady movement of the prosthetic leg, which is mainly used to walk rather than flaunt.

As we have mentioned, the static error comes from the way we implement the algorithms into MCU, with some anti-wind-up and stabilised filter to preserve the prosthetic leg’s stabilisation. Besides, the adaptive self-tuning PI controller is still not optimised, and the rounding of MCU in calculation also plays some part in producing the static error.

![Figure 10. The performance of the prosthetic leg](image)

4. Conclusion

In this paper, a sophisticated and flexible active prosthetic leg model was design and built. The difference between leg’s movement and on leg’s trajectory is to stay to an acceptable degree. Besides, the control theories are successfully adapted to the prosthetic leg, revealing as system modelling and self-tuning adaptive PID controller.

Future research will focus on the developing of a switch mechanism for battery used and non-battery used operation mode. Another perspective relates to expanding the capability of the prosthetic leg to move on undulating surface and stairs. Finally, many functions (e.g. accelerating speed, medical tracking sensor…) can be integrated to the prototype to enhance the comfort and safety of amputees.

REFERENCES


[16] [Microchip Technology Inc., 2012] dsPIC33FJ32MC202204 and dsPIC33FJ16MC304 datasheet..