Design and Control of Mechanical Structure of Lego Robot

**Abstract** - Gait parameters of humanoid robot are optimized by introducing three energy consumption indexes, calculation formulas of driving torque for each joint of humanoid robot are derived based on Lagrange dynamics equation, and mathematic models for gait optimization are established. The minimum consumed energy gait, which similar with human motion, are used to teach the fuzzy logic network, after supervised learning, the FLN can quickly generate the humanoid robot gait parameters. A new approach for real-time gait planning of humanoid robot during walking is proposed based on FNN, ZMP criteria, B-spline interpolation and inverse displacement analysis model. Physical interactivity is a major challenge in humanoid robotics. To allow robots to operate in human environments there is a pressing need for the development of control architectures that provide the advanced capabilities and interactive skills needed to effectively interact with the environment and/or the human partner while performing useful manipulation and locomotion tasks.

Keywords - Lego Robot, Design, Control, Mechanical Structure.

I. INTRODUCTION

With advances in science and technology, the interest to study the human walking has developed the demand for building the Bipedal robots. The development of Bipedal walking robot involves research in heterogeneous areas. This Project describes the attempt in building the Bipedal walking robot. Design of Bipedal robot involves equal amount of mechanical and electronics considerations. There are many factors which are to be considered are cost, actuator, size, weight and controlling of actuators. All these factors have been considered and designed. The robot has six degrees of freedom, with three degrees of freedom per leg. Each leg has Hip, Knee and Ankle. The hip and knee Joints are actuated in vertical plane (Pitch) and the ankle joints are actuated in horizontal plane (Roll).

II. ALGORITHM

1. The Base Positions

   The base positions and the final output position are assumed to be sets of integer motor positions (motor 1 is at position 30, motor 2 is at position 100, etc.) and it is assumed that the motor position values are continuous (that is, if moving the motor controlling the head from position 20 to 21 the head doesn't suddenly move from facing the far right to facing the far left).

2. Fuzzy Progression

   The algorithm works by varying smoothly between base positions using fuzzy logic. For example, take a position for stopping a fall forward and a position for stopping a fall to the right. If instead of a fall just to the right or forward we needed to counter a fall forward and to the right the position needed would be a combination of the two base positions. You want to step forward (as with the falling forward counter) and to the right (as with the falling to the right counter) as in figure 2. The degree to which you would step to the right and forward varies [6]. Say you are falling mostly to the right but also slightly forward. Then the position needed would be mostly the falling right counter but also would have elements of the falling forward counter. There is a progression here – from falling forward to falling to the side – which my algorithm takes advantage of. Using fuzzy logic you can smoothly vary between the different base positions to get the final result.

3. Symmetry

   The algorithm can and in this case has been simplified because the robot is reasonably symmetrical, both in terms of weight distribution and in terms of positions the robot can take (e.g. the robot can move its leg forward and backward the same amount). The main advantage of this is in terms of reusability. For example, you only need to define fuzzy rules in terms of roll and pitch, rather than in terms of falling left, right, forward and back, because you can generalize. Instead of having to define a separate rule for falling left and falling right you can use a general rule for roll and just change the base position the rule is associated with. This essentially quarters the number of fuzzy rules you need to define [7].

   Furthermore if both the pitch and roll inputs have the same range (e.g. they both could be any value between 1.0 to -1.0) and a large pitch is the same value as a large roll then you do not need membership functions for both a large pitch and a large roll – you just need a membership function for a large value.

4. Calculating the final position

   The final position to take for countering a fall is found first by computing the weights for the various fuzzy rules.
Then each position in each base position is multiplied by the weight that corresponds to that base position (for example, the pitch weight could be associated with the falling forward base position). This is done for all weights and the resulting position sets are then added to get a single position set. For example, the head’s motor position in each base position would be multiplied by the corresponding weight. The resulting position values for the head would then be all added together to get the final result. Finally this position set needs to be normalized as in my experience the fuzzy weights rarely sum to 1.0 This is done by dividing each motor position by the sum of the weights to get the final result.

This is equivalent to the following equation:

\[
fp_i = \sum_{j=0}^{n} w_j \cdot \text{position}_j
\]

### III. CONTROLLING OF BIPEDAL ROBOT

Generally any robot has a combination of motors and sensors, which are controlled by microcontrollers. There are wide varieties of motors, sensors and microcontrollers available. In this project low cost and high speed microcontroller and actuators are used. There are 15 D.O.F, each D.O.F has one RC servomotor and it is controlled by LPC2148 ARM7 TDMI-S microcontroller. The robot has the capability to work in closed loop with the help of sensory inputs. The robot is controlled and actuated using a pre-defined sequences and it implements an open loop control and thus does not use sensors. This system is controlled using LPC2148 ARM7 TDMI-S microcontroller. The LPC2148 microcontrollers are based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty.

**Servo Motors**

While dealing with humanoid robot, position control has to be very accurate and should be in a closed loop system. For the following requirement DC servo motors suits the best for this application.

Servos are controlled by sending them a pulse of variable width. The control wire is used to send this pulse. The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction. It is important to note that different servos will have different constraints on their rotation but they all have a neutral position, and that position is always around 1.5 milliseconds (ms).

The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse width Modulation. The servo expects to see a pulse every 20 ms. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

When these servos are commanded to move they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.
When a pulse is sent to a servo that is less than 1.5 ms the servo rotates to a position and holds its output shaft some number of degrees counter clockwise from the neutral point. When the pulse is wider than 1.5 ms the opposite occurs. The minimal width and the maximum width of pulse that will command the servo to turn to a valid position are functions of each servo. Different brands, and even different servos of the same brand, will have different maximum and minimums. Generally the minimum pulse will be about 1 ms wide and the maximum pulse will be 2 ms wide.

Another parameter that varies from servo to servo is the turn rate. This is the time it takes from the servo to change from one position to another. The worst case turning time is when the servo is holding at the minimum rotation and it is commanded to go to maximum rotation. This can take several seconds on very high torque servos.

IV. MECHANICAL DESIGN OF BIPEDAL ROBOT

It is mainly divided into three parts
- Determining the Mechanical constraints.
- Prototype
- Specification and fabrication of model

1. Determining the Mechanical constraints

There are various design considerations when designing a Bipedal robot. Among them, the major factors that have to be considered are Robot’s size selection, Degrees of freedom (D.O.F) selection, Stability.

a) Robot Size Selection

Robot size plays a major role. Based on this the Cost of the Project, Materials required for fabrication and the no of Actuators required can be determined. In this project miniature size of the robot is preferred so a height of 300mm is decided which includes mounting of the control circuits, but the actual size of the robot is 230mm without controlling circuits.

b) Degrees of Freedom (D.O.F)

Human leg has got Six Degrees of freedom (Hip – 3 D.O.F, Knee – 1 D.O.F, Ankle – 2 D.O.F), but implementing all the Six D.O.F is difficult due to increase in cost of the project and controlling of the actuators which become complex, so in this project reduced degrees of freedom is aimed so 3 D.O.F per leg has been finalized. Similarly its defined for hand and neck joints.

c) Stability

With Biped mechanism, only two points will be in contact with the ground surface. In order to achieve effective balance, actuator will be made to rotate in sequence and the robot structure will try to balance. If the balancing is not proper, in order to maintain the Centre of Mass, dead weight would be placed in inverted pendulum configuration with 1 D.O.F. This dead weight will be shifted from one side to the other according to the balance requirement. But in this project no such configuration is used.

2. Prototype

A prototype model has been made using cardboard in order to see how the joints will be formed.

3. Specification and fabrication of model

Degrees of Freedom - (Hip (6 to 8 DOF), Knee (6-10DOF) and Ankle (8-12))

Dimensions:
- Height – 40cm, Width – Around15-18cm
- Leg Length – Greater than 24cm
- Foot pad: Length – 8cm, Width – 6cm
- Width – 32 mm

Before Fabrication weight of the robot is roughly estimated
- Estimated Bracket weight: 50gms – 65gms
- Servo motor: 70 grams
- Total estimated weight for a link (Servomotor + Servomotor Bracket) = 140gms
- Circuits & Batteries: 800grm- 1kg approx
- Total weight of the robot = 3Kg approx.

4. Algorithm for servo motor control

All the Six motors are controlled and actuated simultaneously while maintaining the previous positional values. Initially, the first motor will be serviced with on-time pulse period and during the off-time pulse period of
the motor, second motor will be serviced with on-time pulse period. This type of actuation is continued till all the six motors are serviced. Positional values loaded in the Look-up table and are retrieved and pulses are sent to the motors accordingly. It is shown in the figure below with various ON and OFF time periods. No special algorithms are used for balancing the bipedal robot. In the future we hope to add sensor-based active balancing for making a robot closed loop system. Similarly the pulse pattern will come for hands and neck part servo motor. Fig below is the time pulse for biped part of the humanoid robot.

Another parameter that varies from servo to servo is the turn rate. This is the time it takes from the servo to change from one position to another. The worst case turning time is when the servo is holding at the minimum rotation and it is commanded to go to maximum rotation. This can take several seconds on very high torque servos.

V. WALKING APPLICATIONS

Bipedal Robots are the fundamental block of any advanced walking robots. By making the Bipedal robots fully autonomous, it can be used in environment where human cannot enter. Based on the analysis and study, the output of this type of robots can be used for developing artificial limbs for the physically challenged person.

VI. CONCLUSION

An extensive Literature Survey conducted for the project gave profound insight on the requirements for building the robot. Based on the Literature survey, the inputs for designing the robot have been decided and Software model has been created. After creating the software model it is fabricated and tested.

REFERENCES


Fig.6. The time pulse for biped part of the humanoid robot