International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 2, No. 1, January 2013

Research Paper

ACHIEVING BIPED STABILITY BY PLACING PIEZORESISTIVE PRESSURE SENSORS ON THE SOLE OF A ROBOTIC FOOT

Akhil Cherian Verghese[1]* and Rohit John Varghese[2]

Corresponding Author:* ***Akhil Cherian Verghese,* ✉ *akhilcherian@gmail.com*

Simulating biped stability has long been an obstacle in the creation of practical humanoid robots. This study demonstrates, through experimentation, the feasibility of using piezoresistive pressure sensors, located on the sole of a robotic foot, as a means of maintaining balance while the robot is in stable equilibrium. Adjustments to this method may allow this approach to be extended to situations of dynamic equilibrium, particularly in cases where the phase of motion is determined by the angle of the foot.

Keywords: Biped stability, Biped robot, Biped balance, Humanoid, Pressure sensors, Robotic foot

INTRODUCTION

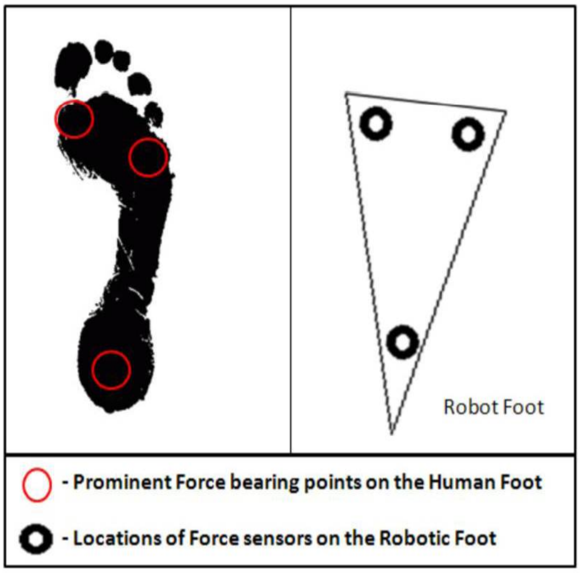
The commonly practiced methods to maintain biped balance involve the use of accelerometers, gyroscopes or image processing, all of which determine the degree of tilt of the robot. Some of these methods add considerably to the cost of the robot, while others are susceptible to mechanical vibrations. However, the most important drawback of these techniques is that, by definition, they respond to tilt and not force, and, as such, corrective action can only be taken when the robot begins to fall.

To expand on this shortcoming, let us consider the human body and how it maintains balance. The inner ear (the biological equivalent of accelerometers or gyroscopes) and visual stimuli (the biological equivalent of image processing) certainly play a role. However, humans also maintain their equilibrium by resisting external forces that they perceive, either through nerves in the skin, that experience said forces directly, or through pressure variations in their feet (Kavounoudias *et al.*, 1998; and Maurer *et al.*, 2001), that arise as a result of these forces. The first is extremely

[1] Physics and B.E. Civil Engineering, BITS Pilani.
[2] Mechanical Engineering, BITS PIlani.

difficult to recreate, but the second is quite practical, especially, when we consider the fact that there are only three major points of contact between the foot and the surface on which it rests, as shown in Figure 1.[1]

Figure 1: Position of Significant Points on the Foot

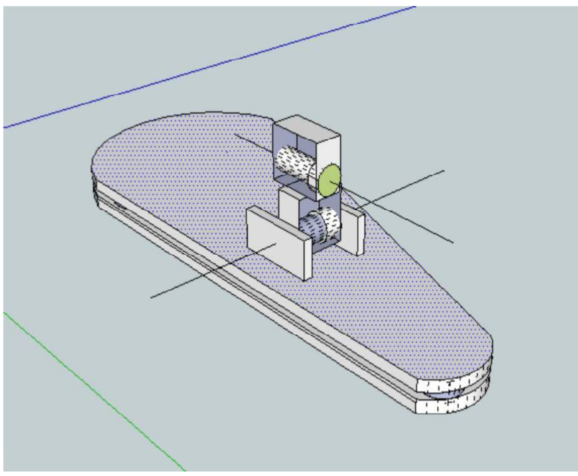


By using pressure sensors at these three points, we can accurately determine changes in torque about the robotic foot, which is used to calculate the location of the zero moment point on a two dimensional plane. This information can, through the use of appropriate equations, be used to determine the corrective action necessary to stabilize the biped (Goswami, 1999; Kalamdani *et al.*, 2006; and Kalamdani *et al.*, 2006), by returning the zero moment point to a certain area or 'safe zone' on the foot.

In our experiment, only one axis of tilt or force was considered, using two pressure sensors in a line. The robotic leg created comprised a base plate with two pressure sensors in a line and a glass rod with a weight at the upper end, acting as the shin (Figure 2). This structure could balance on a platform tilting along one axis by moving the shin, using a single axis motor, such that the zero moment point falls within the predetermined safe zone (Sardain and Bessonnet, 2004). The sensors and motors are connected through an Arduino[2] microcontroller, which sends and receives the signals.

In the algorithm and arduino code presented, two axes are considered, with four sensors, and the position of the zero moment point is calculated on a two dimensional plane. Corrective action is calculated for two motors, one along the x-axis, the other, along the y-axis, as represented in the diagram below. This approach is adopted due to the ease of dealing with two perpendicular axes while calculating the position of the Zero Moment Point (ZMP). The ZMP can, however, be plotted with a minimum of three points of contact. The general equation of its position is,

Figure 2: Design of the Robotic Foot



[1] Footprint taken from http://www.footprintdesigns.co.nz/

[2] http://www.arduino.cc/

$ZMP(x) = \Sigma(Xi * Fi)/\Sigma(Fi)$

$ZMP(y) = \Sigma(yi * Fi)/\Sigma(Fi)$

Where Fi is the force recorded on the i^{th} sensor and Xi and Yi are its coordinates on the foot.

ALGORITHM

- The force value is read from all four sensors.
- In most cases, as the sensors are not identical and may not be placed perfectly, a corrective constant is subtracted from one or more of the sensor values. This constant is determined by displaying the sensor values on the laptop screen, while the foot is on a perfectly flat surface, and ensuring that both values on each axis are equal.
- 256 sets of readings are taken at a time and the averages are used for further calculations.
- Using the following equation, the ZMP is calculated for each of the sensor pairs, to give its x and y coordinate, by using the following equation:

$$ZMP(x \text{ or } y) = \frac{F1(x \text{ or } y) - F2(x \text{ or } y)}{F1(x \text{ or } y) + F2(x \text{ or } y)} x \frac{L(x \text{ or } y)}{2}$$

Where $F1$ and $F2$ are the two pressure inputs and L is the physical distance between the two sensors.

- The location of the ZMP is then compared with three regions, separated by two concentric ellipses on the foot, the centermost region of maximum stability, where no corrective action is to be required; the intermediate region, where corrective action is taken at a rate that varies with the magnitude of the displacement of the ZMP; the outer region, where maximum corrective action is taken.

The Equations of the two ellipses are given below

$$\frac{ZMPx^2}{B1^2} + \frac{ZMPy^2}{L1^2} = 1$$

$$\frac{ZMPx^2}{B2^2} + \frac{ZMPy^2}{L2^2} = 1$$

Where $L1 < L2$ and $B1 < B2$ are, respectively, the semi major and minor axes of the two ellipses.

- The direction and speed of corrective action is decided by the position of the ZMP. The Inverse Tan of ($ZMPx/ZMPy$) is taken as Theta and corrective action in x and y directions are correspondingly proportionate to sin(Theta) and cos(Theta)
- The Individual sensor values, Location of the ZMP, Theta, and extent of corrective action are continuously displayed via a serial monitor.

Corrective Action in x

$$= \sin\left(Arc\tan\left(\frac{ZMPx}{ZMPy}\right)\right) \times Motor\ speed$$

Corrective Action in y

$$= \cos\left(Arc\tan\left(\frac{ZMPx}{ZMPy}\right)\right) \times Motor\ speed$$

EXPERIMENTATION

The experiment done to confirm our calculations was performed in the Center for Robotics and Intelligence Systems laboratory, BITS Pilani, India. The aim of the experiment was to create a foot that could balance on a platform tilting along one axis.

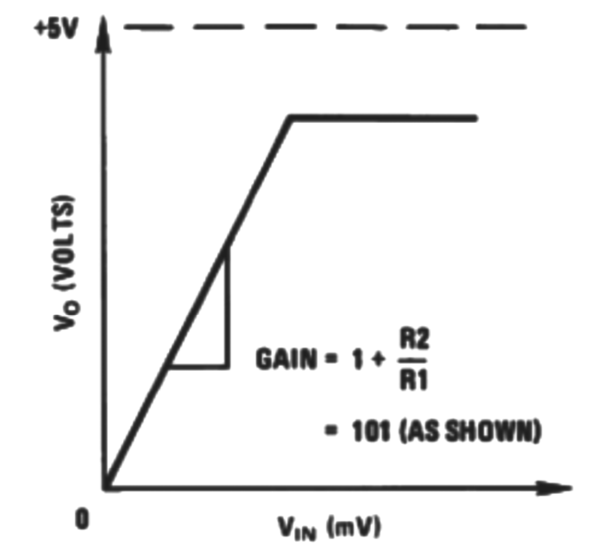
Apparatus

- Arduino Microcontroller[3]
- Flexiforce Sensors[4]
- LM 324 Op-Amp[5]
- Assorted Resistors
- Laptop

Procedure

The first step of the lab work was interfacing the 'Flexiforce' sensors with our Arduino circuit board. The difficulty being that, while the Flexiforce sensor is a piezoresistive sensor (value of resistance varies with force applied), the Arduino sensors only read a change in voltage (0-5 V). Therefore, a linear change in conductance (1/R), with force, had to be converted into a linear change in voltage. The term linear is used to highlight the fact that, while there are many methods to convert a varying resistance into a varying voltage, it is important to ensure that the voltage produced is directly proportional to the force in order to use it in calculations. This can be achieved by using an Op-Amp (LM 324) circuit. In this circuit, the voltage input (V_O) to the Arduino increases as V_{IN} $(1 + R1/R2)$, where $R1$ is a constant, adjusted according to desired sensitivity, and $R2$ is the sensor, as shown in Figure 3.[5] As $1/R2$ varies linearly with force, $V_{IN} * R1/R2$ varies linearly with force, and V_O can be used to recalculate the force. A V_{IN} value of approximately 2 volts was maintained, allowing V_O to vary from 2 to 4.

Figure 3: Graph of V_O vs. V_{IN}



After interfacing the pressure sensors with our microcontroller, the code to calculate the center of pressure with two sensors was implemented, using the formula $((Input_1 - Input_2)/(Input_1 + Input_2)) \times$ Constant. Theoretically, the constant used should be the length of the foot (if the physical location of the center of pressure is to be determined), but it can be multiplied by any constant to adjust the sensitivity. This value is also proportional to the displacement of the ZMP.

The area of the foot was then categorized into three zones, the safe zone, in which the motors take no corrective action; the intermediate zone, in which they take corrective action proportional to the location of the ZMP within the zone; and the critical outer zone, where they take maximum corrective action. The position of the ZMP is checked each time the shin is moved, as a feedback mechanism, and the speed of the corrective action reduces as the leg becomes more stable (Sardain and Bessonnet, 2004).

[3] Arduino library available at http://www.arduino.cc/

[4] Data sheet, along with product description are available at http://www.sparkfun.com/products/8685

[5] http://www.fairchildsemi.com/ds/LM/LM324.pdf (figures also taken from the datasheet).

A point to note here is that an object will topple when its ZMP moves outside the boundary of the smallest convex curve that encloses all its points of contact with the ground. (In the case of a biped balancing on one foot, that is approximately the outer boundary of the foot).

The condition when the ZMP reaches this boundary is known as the 'brink of instability'.

The final stage of the project was constructing a basic foot like structure and load, with which to demonstrate the functioning of the sensors and the program. Like the algorithm, the code given below uses four sensors (The input and *ZMP*, *ZMPx*, of the two missing sensors is taken as 0). Again, the *ZMP* can be located with a minimum of 3 points of contact, using the formula given in the introduction; however, using two perpendicular axes simplifies the process.

Arduino Code – Important Portion

```
void loop() // continuous process
{
/* The sensor values are inputted and averaged, 256 values at a time (code not included here). "totalA" and "totalB" are the sensor values from both ends of the y axis.
*/
/*Constant values of l1 and b1 and l2 and b2 are chosen to represent the semi major and semi minor axes of the inner and outer ellipse, respectively.
Within Ellipse1: No corrective action required.
Within Ellipse 2 (outside 1): Corrective action taken proportional to displacement
Outside Ellipse 2: Maximum corrective action taken*/
float temp1=(((ZMPy*ZMPy)/(l2*l2))+
((ZMPx*ZMPx)/(b2*b2)));
//outer ellipse
float temp2=(((ZMPy*ZMPy)/(l1*l1))+
((ZMPx*ZMPx)/(b1*b1)));
//inner ellipse
if (temp1>=1)
  {
  Serial.print("\n corrective
  action required");
  Serial.print("\t Decrement Motor
  x and y at ");
  if(ZMPx>0)
    Serial.print(speed2*sin
    (theta));
  else
    Serial.print(-speed2*sin
    (theta));
  //x motor corrective action
  Serial.print("\t & ");
  if(ZMPy>0)
    Serial.print(speed2*cos
    (theta));
  else
    Serial.print(-speed2*cos
    (theta));
  //y motor corrective action
  }
```

```
/*End of loop for outer region (maximum correction)*/
else if (temp2>=1)
   {
   Serial.print("\n corrective
   action required");
   Serial.print("\t Decrement Motor
   x and y at ");
   if(ZMPx>0)
      Serial.print((speed1+
      (speed2*temp1)-1) * sin
      (theta));
   else
      Serial.print(-(speed1+
      (speed2*temp1)-1) *sin
      (theta));
   //x motor corrective action
   Serial.print("\t & ");
   if(ZMPy>0)
      Serial.print((speed1+
      (speed2*temp1)-1) *cos
      (theta));
   else
      Serial.print(-(speed1+
      (speed2*temp1)-1) *cos
      (theta));
   //y motor corrective action
   }
/*End of loop for intermediate region (adjustable correction). This loop is second as the first one tests a stronger condition, which would be more urgent*/
```

Noise Cancellation

Various methods for reducing measurement error were tried, such as taking a moving average and applying various filters to the input values. However, most of the filters reduced the response time of the device considerably and lowered the effectiveness of the corrective action, due to the limitation of our onboard processing speed. Therefore, a simple average method was used, taken after every 256 inputs, since division by a power of 2 requires comparatively fewer clock cycles in the binary system. The various filters did, however, work well in theory and with a faster processor they could greatly improve the accuracy of response.

Visual Representation of ZMP

With the serial inputs provided, a visual representation of the pressure gradient was created. This comprised an arrow that pointed in the direction of the corrective action required and which grew proportionally with the amount of corrective action required. This was accomplished by setting its direction, using the sign of the displacement of the ZMP, and its length, using the magnitude of displacement of the ZMP. The arrow was then superimposed on a figure of the foot, with the aforementioned ellipses also displayed. A brief algorithm of this calculation is given below:

If $ZMPy < 0$

  Arrow Y is positive

Else

  Arrow Y is negative

Arrow Y Mag = $|K \times ZMPy|$, where K is a constant, adjusted in such a manner that the arrow cuts the first ellipse when temp2>0 and the second when temp1>0

If $ZMPx$ is also given, Arrow$X$ and Arrow$Y$ are considered as the x and y projections of the Final Arrow Vector.

Therefore,

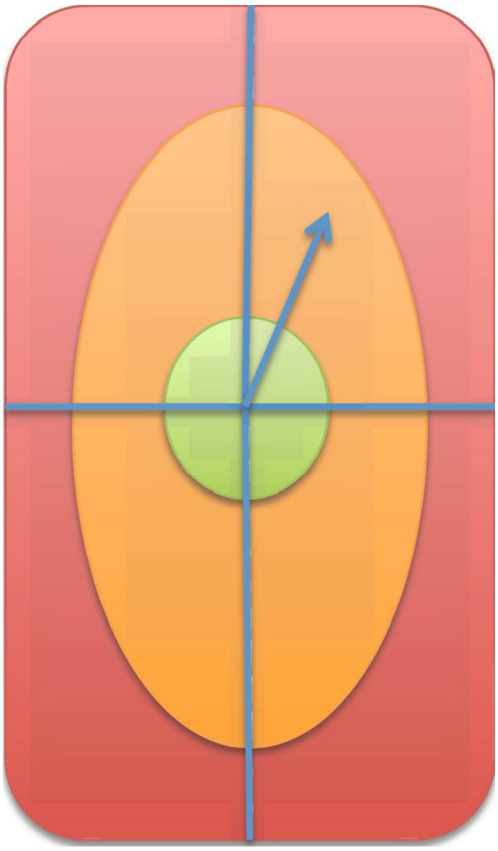
$$|\text{Arrow}| = ((\text{Arrow}X)^2 + (\text{Arrow}Y)^2)^{1/2}$$

And

$$\text{Angle (Arrow)} = \text{atan}(ZMPx/ZMPy)$$

This representation could be made more understandable by color-coding, as demonstrated in Figure 4.

Figure 4: Possible Format for Visual Representation



APPLICATIONS

The above experiment and algorithm can be used to help maintain balance in a variety of situations. The one demonstrated in the experiment is when the surface on which the leg rests is unsteady, but the same principle could be used when the robot is off balance because it is pushed or pulled, or even when its center of gravity is offset because it is carrying a heavy object. It could also be extended to cases where the robot is walking, and help facilitate real time corrections by plotting the zero moment point (Huang *et al.*, 2000; Grizzle *et al.*, 2001; Goswami and Kallem, 2004; and Sugihara and Nakamura, 2006). Although this method cannot be directly applied to situations involving dynamic equilibrium, it has applications even in these methods. Specifically, even for cases using advanced techniques like transverse linearization (Manchester *et al.*, 2009), this method could help in determining the angle of the foot, which is often used to determine the phase of the stride, and the tilt of the surface, improving these techniques over rough terrain.

This experiment uses the most fundamental principle of balance- the center of gravity must fall safely within the base of the foot and the closer it is to the edge, the more unstable the object. As such, this particular design can be used for balance in everyday equipment as well. A crane with a similar pressure sensing design could have a topple-warning system, which would help while carrying a large load and indicate the level of stability in an easily understandable manner. The same principle could apply to all equipment that carries or transports a considerable weight.

CONCLUSION

The robotic foot balanced well on a surface tilting at considerable speeds- over half a radian per second. However, with faster processing and more accurate sensors, greater angular velocities could be compensated for by this mechanism. In this experiment, the average input from approximately 256 cycles was used for calculation, but the number of cycles can be increased, for more accurate measurements, or reduced, for a lower response time.

The results of the experiment have shown that this approach can, with certain alterations, be applied to numerous situations pertaining to equipment stability and provide an affordable alternative to the current methods of maintaining biped balance.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. We would like to thank our instructor, Professor Arshad Javed, for his support and guidance during the project. Our thanks also to Rohit De Sa for all his advice and help throughout our lab work. Lastly, we are grateful to the authorities at BITS Pilani and Mr. Tulsi, for allowing us to use the CRIS laboratory to conduct our experiment.

REFERENCES

1. Goswami A (1999), "Postural Stability of Biped Robots and the Foot-Rotation Indicator Point", *Int. Jour. of Rob. Res.*, Vol. 18, No. 6, pp. 523-533.
2. Goswami A and Kallem V (2004), "Rate of Change of Angular Momentum and Balance Maintenance of Biped Robots", Proceedings of the 2004 IEEE International Conference on Robotics & Automation.
3. Grizzle J W, Abba G and Plestan F (2001), "Asymptotically Stable Walking for Biped Robots: Analysis via Systems with Impulse Effects", *IEEE Transactions on Automatic Control*, Vol. 46, No. 1.
4. Huang Q, Kaneko K, Yokoi K, Kajita S *et al.* (2000), "Balance Control of a Biped Robot Combining Off-Line Pattern with Real-Time Modification", Proceedings of the 2000 IEEE International Conference on Robotics & Automation.
5. Kalamdani A, Messom C and Siegel M (2006a), "Tactile Sensing by the Sole of the Foot Part I: Apparatus and Initial Experiments Toward Obtaining Dynamic Pressure Maps Useful for Stabilizing Standing, Walking and Running of Humanoid Robots", *IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications*.
6. Kalamdani A, Messom C and Siegel M (2006b), "Tactile Sensing by the Sole of the Foot Part II: Calibration and Real-time Processing", 3rd International Conference on Autonomous Robots and Agents.
7. Kavounoudias A, Roll R and Roll J P (1998), "The Plantar Sole is a 'Dynamometric Map' for Human Balance Control", *Cognitive Neuroscience Neuroreport*, Vol. 9, No. 14, pp. 3247-3252.
8. Manchester I R, Mettin U, Iida F and Tedrake R (2009), "Stable Dynamic Walking Over Rough Terrain: Theory and Experiment", in Proceedings of the

International Symposium on Robotics Research (ISRR).

9. Maurer C, Mergner T, Bolha B and Hlavacka F (2001), "Human Balance Control During Cutaneous Stimulation of the Plantar Soles", *Neurosci. Lett.*, Vol. 302, No. 1, pp. 45-48.

10. Sardain P and Bessonnet G (2004), "Forces Acting on a Biped Robot, Center of Pressure—Zero Moment Point", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 34, No. 5.

11. Sugihara T and Nakamura Y (2006), "High Mobility Control of Humanoid Robots Based on an Analogy of ZMP-COG Model and Carted Inverted Pendulum Model", *Robotics Society of Japan Anual Conference*, Vol. 24, No. 1, pp. 74-83.