Dynamic Lateral Balance of Humanoid Robots on Unstable Surfaces

Aditya Sripada, Abhishek Warrier, Arpit Kapoor, Harshit Gaur

Abstract: This work presents a control algorithm to achieve dynamic lateral balance in humanoid robots on unstable and geodynamic surfaces such as a seesaw or a suspension bridge. The proposed method generates a stable pose at any given point using the feedback from an Inertial Measurement Unit (IMU) in realtime which enables the robot to balance on dynamic surfaces. The algorithm provides a robust control of the Center of Gravity (CoG) of the robot and constricts the CoG from moving away from the most stable pose of the robot in the double support mode thus achieving stability at any given point in time. The method implemented is relatively less heavy in the terms of computation and easy to implement. The proposed system works on any humanoid platform irrespective of size and weight, given the robot has the required degrees of freedom needed to achieve balance.

Keywords—Dynamic Stabilisation, Lateral Balance, Humanoid Robotics, Bipedal Robots.

I. INTRODUCTION

Dynamic balance of humanoid robots is one of the most basic yet challenging problem in robotics. It is extremely important for a legged robot to be able to traverse though uneven and unstable surfaces. There is a great deal of research performed on push recovery, dynamic stabilisation while walking and balance in transverse plane but lateral balance in the coronal plane is very important for the robot to be efficient.

To achieve lateral balance, the robot's legs are manipulated in such a way that the Centre of Gravity of the robot is always restored to its original position in the double support mode when the humanoid is standing on a flat surface and is most stable. It is achieved by generating the pose that provides balance by calculating the angles of the pose for each individual motor in the legs and actuating them.

Aditya Sripada was a final year undergraduate student with the Department of Electrical and Electronics Engineering, SRM University, Chennai
(Email: adityassripada@gmail.com)
Abhishek Warrier is a third year undergraduate student with the Department of Computer Science and Engineering, SRM University, Chennai
(E-mail: warrier.abhishek@gmail.com)
Arpit Kapoor is a third year undergraduate student with the Department of Computer Science and Engineering, SRM University, Chennai
(E-mail: arpitkapur.dps@gmail.com)
Harshit Gaur is a third year undergraduate student with the Department of Mechanical Engineering, SRM University, Chennai
(E-mail: hgaur014@gmail.com)

II. RELEVANT WORK

Nagarajan et. al [1] proposed a Universal balancing controller capable of stabilizing planar bipedal robot in dynamic environments, such as seesaw and bongoboards, as well as static environments like curved or flat floors. As the dimensions of the state spaces belonging to these different dynamic systems differ, the controller is derived as a single output feedback instead of full state feedback. The robustness of the controller against disturbances and parameter uncertainties is analyzed. Also, pre-eminence and universality of the derived controller over similar LQR and H ∞ controllers is validated.

Desai et. al [2] introduced a virtual-model based controller to investigate lateral balancing of humanoid robot on challenging surface. They propose that humans prefer intuitive task- space control for lateral balancing on simple as well as challenging surfaces. With simulation of a planar model, the subsequent balancing behavior against human lateral balancing on flat ground and on a challenging surface are compared. It is found that after tweaking the controller to respond to disturbances, it could mirror human behavior on a seesaw.

Anderson et. al [3] proposed a model-based policy-mixing controller to perform dynamic tasks with a humanoid robot. The controller is augmented with automated model adaptation techniques to perform these tasks on a Sarcos Primus humanoid robot. The model-based controller exposes modeling errors that are not apparent when performing slower or less difficult tasks. Experiments are conducted to demonstrate the performance of this augmented controller in balancing Sarcos Primus humanoid on an unstable seesaw platform in the presence of unexpected disturbances and significant model errors.

III. THE HUMANOID ROBOT

The experiment uses a 20 DOF Humanoid as shown in Fig.1.

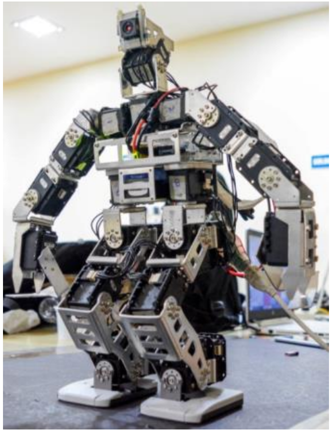

Fig.1. Humanoid Robot used for the experiment

The technical specifications of the humanoid are as shown in the Table I. A Hard Kernel Odroid XU4 is used as the main computer. Twenty Robotis MX28 Robot Actuators are used for actuation.

TABLE I. Robot Specifications

Robot	**Specification**
Height	45 CM
Weight	2.5 KG
Degrees of Freedom	22
Actuators	20 X MX28T Dynamixel
Sensor	9DOF IMU
Main Computer	Odroid XU4
Operating System	Linux
Battery	3S 2200mAh 20C Lipo

Human body is arguably the most complex system and replicating all the motions a human body can perform in a humanoid is nearly impossible. Hence it is very hard to provide a robot with all kinds of degrees of freedom as in a human body. Human body has approximately two hundred and forty degrees of freedom but due to structural constraints, the humanoid used for this experiment has only twenty degrees of freedom. The robot's links and joints are as shown in the Fig.2. The robot has four limbs, a head and a torso. Each leg has 6 actuators and the hands have 3 each. The motion limits of the joints are as shown in the Table. II. The axes of the motor rotations are as represented in Fig. 2.

Number	**Joint Name**	**Axis**	θ_{Min}	θ_{Max}
Right Arm				
1	Shoulder Pitch	Z_1	$-4\pi/3$	$4\pi/3$
3	Shoulder Roll	Z_2	$-\pi/2$	$\pi/2$
5	Elbow	Z_3	0	$5\pi/6$
Left Arm				
2	Shoulder Pitch	Z_1	$-4\pi/3$	$4\pi/3$
4	Shoulder Roll	Z_2	$-\pi/2$	$\pi/2$
6	Elbow	Z_3	$-5\pi/6$	0
Right Leg				
7	Hip Yaw	Z_1	$-5\pi/6$	$\pi/4$
11	Hip Roll	Z_2	0	$\pi/3$
9	Hip Pitch	Z_3	$-\pi/2$	$\pi/6$
13	Knee	Z_4	0	$3\pi/4$
17	Ankle Pitch	Z_5	$-\pi/3$	$\pi/3$
15	Ankle Roll	Z_6	$-\pi/6$	$\pi/3$
Left Leg				
8	Hip Yaw	Z_1	$-\pi/4$	$5\pi/6$
12	Hip Roll	Z_2	$-\pi/3$	0
10	Hip Pitch	Z_3	$-\pi/6$	$\pi/2$
14	Knee	Z_4	$-3\pi/4$	0
18	Ankle Pitch	Z_5	$-\pi/3$	$\pi/3$
16	Ankle Roll	Z_6	$-\pi/6$	$\pi/3$
Head				
19	Head Yaw	Z_1	$-5\pi/6$	$5\pi/6$
20	Head Pitch	Z_2	$-\pi/3$	$\pi/6$

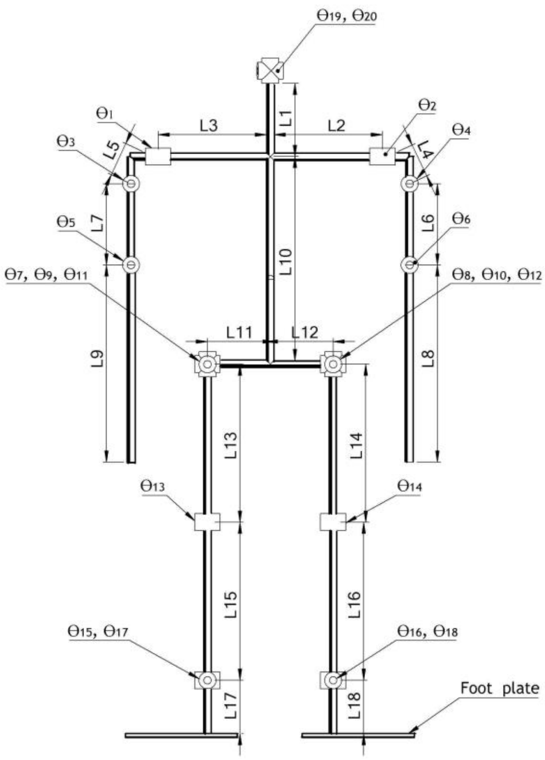

Fig.2 Robot joint locations

IV. CONTROL ALGORITHM

A. IMU Data Processing:

A 9DoF IMU is placed at the centre of the gravity of the robot. Raw data obtained from the IMU is fused using a Kalman Filter and accurate predictions for the roll, pitch and yaw of the robot are obtained. As the robot attempts to stabilize only in a single plane and also due to the placement of the IMU, only the pitch values are required. But this data is still prone to noise. So, to further smoothen the data, a mean filter is implemented by first storing the IMU data in a buffer and considering the mean of the buffer values as the final reading.

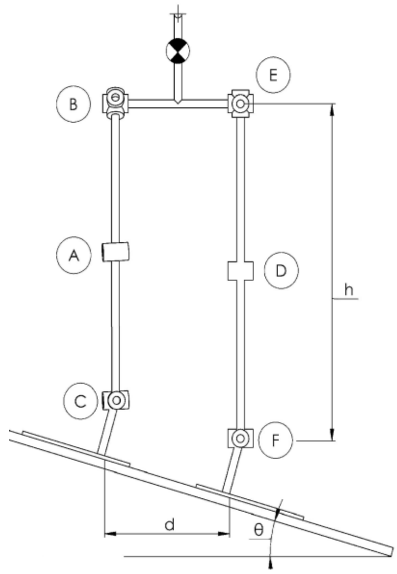

Fig.3 Robot pose while on an unstable surface

B. Pose Generation:

The stable pose is created by actuating the motors with axes parallel to the frontal axis and to the transverse axis in the robot's legs.

Motion with respect to the transverse axis:

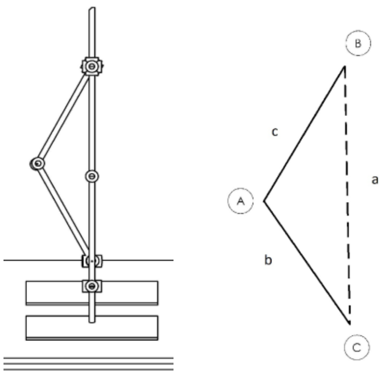

Fig.4 Side View of the robot depicting the positions of the motors with axes parallel to the transverse axis

$$a = h - d \sin\theta \tag{1}$$

Every time the robot goes out of balance, the IMU data is used to generate a stable pose. The angles are calculated by using Cosine Rule of Triangles and by using Eq (1). The angles are calculated as shown in the Eq (2), Eq (3) and Eq (4) to drive the motors with axes parallel to the transverse axis.

$$A = \cos^{-1}\left[\frac{b^2+c^2-h^2-d^2\sin^2\theta+2hd\sin\theta}{2bc}\right] \tag{2}$$

$$B = \cos^{-1}\left[\frac{c^2-b^2+h^2+d^2\sin^2\theta-2hd\sin\theta}{2(h-dsin\theta)c}\right] \tag{3}$$

$$C = \cos^{-1}\left[\frac{b^2-c^2+h^2+d^2\sin^2\theta-2hd\sin\theta}{2(h-dsin\theta)b}\right] \tag{4}$$

Where,

b is the length of the link joining the ankle and the knee (m)
c is the length of the link joining the knee and the hip (m)
h is the total height of the robot's leg (m)
d is the distance between the centers of the legs (m)
θ is the Pitch of the center of gravity of the robot

Motion with respect to the Frontal Axis:

The motors with axes parallel to the frontal axis are actuated with the angle of inclination of the plane to maintain the torso of the robot's body parallel with the ground.

C. Motor Actuation

In such a dynamic system, the actuation of motors is very crucial. For controlling the motors, Pypot which is a high-level python library for interfacing with Dynamixel Motors is used. Using Pypot, the speed and goal position of multiple motors can be controlled simultaneously.

Initially, the calculated joint angles were directly written to the motors and the speed of the motors was kept constant. But this led to a few problems. When the speed was set too low, the motors would fail to reach the designated goal position in time causing the robot to fall. To the contrary, when the speed was set too high, although the robot was responsive, the motions were quite abrupt and sudden, which destabilized the robot.

A solution was to dynamically control the speed of each motor which is calculated as given in Eq (5).

$$\omega = \omega_0 + \omega_0\left(\frac{|\varphi_{new}-\varphi_{prev}|}{180}\right) \tag{5}$$

Where ω = Final Speed
ω_0 = Base Speed
φ_{new} = New Joint Angle
φ_{prev} = Current Joint Angle

Thus, the speed of each motor is proportional to the angular displacement it needs to cover.

Once the IMU data is obtained and the pose is calculated, the robot needs to dynamically decide which leg to manipulate. This is done so by the algorithm given below in Algorithm 1.

Algorithm 1 Deciding which leg to move

```
theta = imu_pitch
Calculate angleA. angleB, angleC using theta
if theta > 0:
        Align footplates to slope
        Initialize right leg
        Write angleA. angleB, angleC to left leg

else if theta <0:
        Align footplates to slope
        Initialize left leg
        Write angleA. angleB, angleC to right leg
end if
```

V. RESULTS

The proposed algorithm was applied and tested on the robot specified in the section 3 of this paper on a base platform actuated by an MX 64T Dynamixel motor to achieve the better control. The robot was tested on this platform oscillating from 16.69 degrees to -16.69 degrees and dynamic balance in the coronal plane was observed.

The robot can be seen exhibiting lateral balance in the transverse plane on an inclined plane, inclined at 16.69 degrees with respect to the ground from its front & side view as shown in Fig.5. The robot can be seen exhibiting lateral balance in the transverse plane on an inclined plane, inclined at 6.74 degrees with respect to the ground from its front & side view as shown in Fig.6. The robot can be seen exhibiting lateral balance in the transverse plane on an inclined plane, inclined at -6.74 degrees with respect to the ground from its front & side view as shown in Fig.7. The robot can be seen exhibiting lateral balance in the transverse plane on an inclined plane, inclined at -16.69

degrees with respect to the ground from its front & side view as shown in Fig.8.

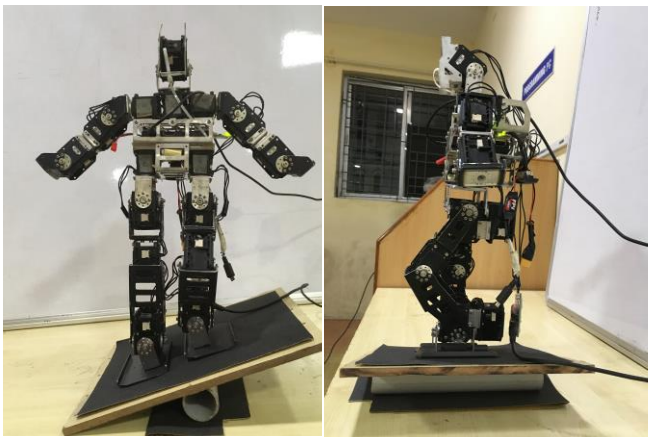

Fig.5 Front and Side views of the Robot balancing on a plane with an inclination of 16.69 degrees

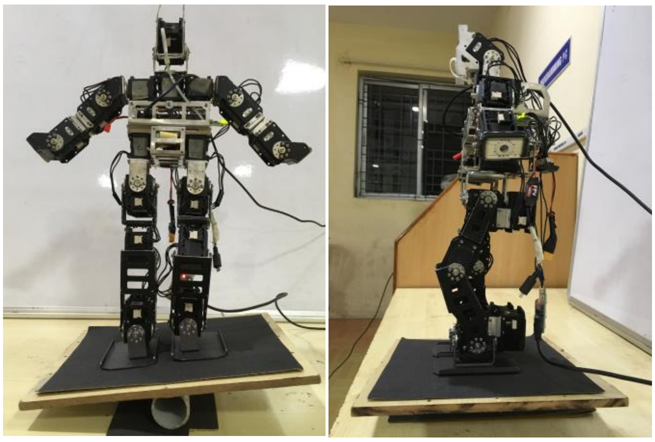

Fig.6 Front and Side views of the Robot balancing on a plane with an inclination of 6.74 degrees

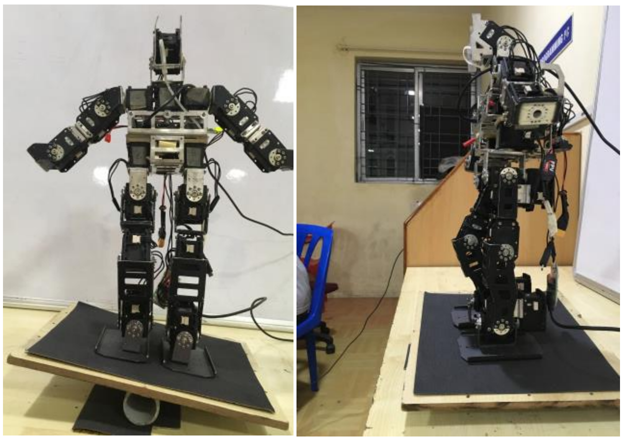

Fig.7 Front and Side views of the Robot balancing on a plane with an inclination of -6.74 degrees

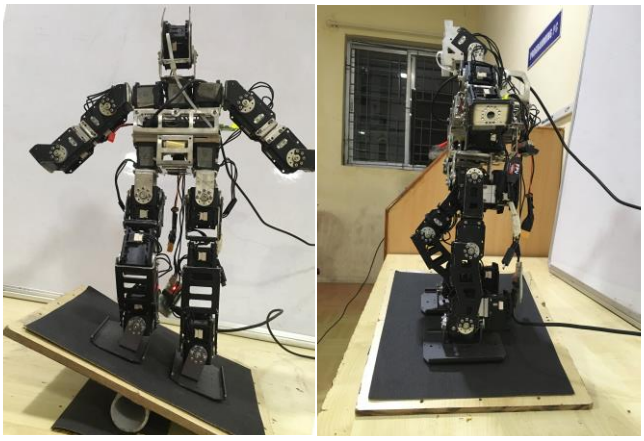

Fig.8 Front and Side views of the Robot balancing on a plane with an inclination of -16.69 degrees

VI. FUTURE WORK

The research work proposed in this paper has a huge scope of improvement and advancement. We aim to achieve balance in 3 dimensions as a combination of both coronal and sagittal plane and to make the control algorithm more robust. The response of the whole system can also be improved by incorporating a PID Controller. Our results prove that the proposed algorithm is efficient in achieving dynamic balance in the transverse plane, but we aim to fine tune it and make it more responsive to transient changes in the surfaces.

REFERENCES:

[1] U. Nagarajan and K. Yamane, "Universal balancing controller for robust lateral stabilization of bipedal robots in dynamic, unstable environments," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 6698-6705.

[2] R. Desai, H. Geyer and J. K. Hodgins, "Virtual model control for dynamic lateral balance," *2014 IEEE-RAS International Conference on Humanoid Robots*, Madrid, 2014, pp. 856-861.

[3] S. O. Anderson and J. K. Hodgins, "Adaptive torque-based control of a humanoid robot on an unstable platform," *2010 10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, 2010, pp. 511-517.