Team AcYut
Team Description Paper 2018

Vikram Nitin, Archit Jain, Sarvesh Srinivasan, Anuvind Bhat,
Dhaivata Pandya, Abhinav Ramachandran, Aditya Vasudevan,
Lakshmi Teja, and Vignesh Nagarajan

Centre for Robotics and Intelligent Systems
Birla Institute of Technology and Science, Pilani,
Rajasthan - 333031, India

Abstract. This paper describes the humanoid robots which we, Team AcYut, plan to use for the RoboCup Humanoid Teen-size Soccer competition. AcYut uses Preview Control of the Zero Moment Point (ZMP) for walking pattern generation with a closed-loop stabilization algorithm. It is equipped with an Xsens Inertial Measurement Unit (IMU), an Intel NUC on-board computer, and a Sony PS Eye Camera. Dynamixel MX-106 Servo motors are used for actuation. We use Histograms of Oriented Gradients (HOGs) for object-detection, and Monte-Carlo Localization to determine the relative position and orientation of the robot within the field. The optimal path to be followed is computed using a fast geometrical approach.

Keywords: AcYut, BITS Pilani, RoboCup, Humanoid

1 Introduction

AcYut is the name given to the series of Humanoid Robots developed by undergraduate students at the Birla Institute of Technology and Science, Pilani, under the guidance of Prof. BK Raut from the Department of Mechanical Engineering.

AcYut 1 in 2008 was India's first indigenously developed humanoid robot. Since then, the team has developed many more iterations of AcYut, with each successive version building on the strengths of those before it. The team's robots for this edition of the RoboCup are AcYut 7 and AcYut 8.

Team AcYut has been a regular participant in the RoboCup Soccer competitions, with our best result being a 3rd place finish. Aside from this, the team has also won accolades at competitions such as the Iran Open RoboCup (2nd place finish in 2013), the RoboGames (Gold medals in 2010 and 2011) and the FIRA 2010 (world record for the most weight lifted by a large humanoid).

1.1 Summary of Progress

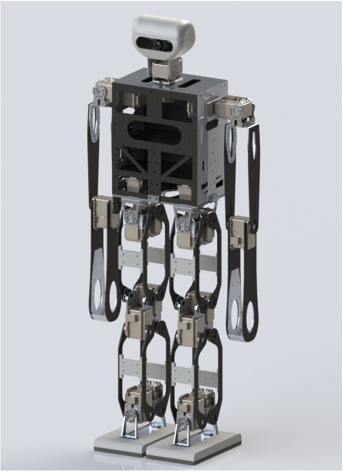
We have upgraded our electronics setup by adding a circuit to detect if our Lithium Polymer battery drops below a certain critical voltage. A signal is sent to the NUC which instantly brings the robot to a safe state and then relaxes the joints. Apart from this, switches have been added to the setup to control the robot.

We have moved from open-loop walk to closed-loop walk, by using a combination of Adaptive Step Timing and a PID controller on the orientation of the hip plate of the robot. This enables the robot to recover from pushes and collisions. For ball and obstacle detection, we now use Histograms of Oriented Gradients (HOGs).

2 Mechanical Design

2.1 AcYut 8

AcYut 8 has 18 degrees of freedom distributed as follows: 5 in each leg, 3 in each arm and 2 in the head. A combination of Dynamixel MX-106 and Dynamixel MX-64 actuators is used.



Since last year's competition, the length of the arms has increased to aid in get-up. AcYut 8 is fully capable of getting up from a fallen position.

All leg brackets are made of Carbon Fibre Reinforced Polymer (CFRP), and the torso is a hybrid of CFRP and sheet metal. The torso is designed for easy access of interior components (the NUC, the LiPo battery and the IMU), and its plates slide in and out to facilitate this.

AcYut 8 is 91 cm tall and weighs 6.3 kg.

2.2 AcYut 7

AcYut 7 has the same degrees of freedom as AcYut 8, namely, 5 in each leg, 3 in each arm and 2 in the head. A combination of Dynamixel MX-106 and Dynamixel MX-64 actuators is used.

AcYut 7 too is capable of getting up from a fallen position. The entire skeleton is made out of Aluminium 6061-T6, including all leg brackets and the torso. The robot is 99 cm tall and weighs 7.8 kg.

3 Electronics

The following is a list of electronic components used in both AcYut 8 and AcYut 7.

1. Processor : Intel NUC D34010WYKH
2. IMU : Xsens 9 DoF IMU
3. Camera : Sony Playstation Eye
4. Battery : 14.8 V Lithium Polymer

Communication to the servo motors takes place through the USB2Dynamixel adapter. A Wifi communications chip is used to interface with the Game Controller.

A circuit has been added to take appropriate action if the battery voltage drops below a certain threshold. This circuit operates in three regions : Dangerous-low (0-11.8), Safe and Dangerous-high (16.2+). To interface with the NUC, the circuit is connected to the GPIO pins on-board an Arduino UNO, which sends a signal to the NUC through a serial port.

4 Walk

Preview Control of the Zero-Moment Point [1] is used to create the ideal Centre of Mass (CoM) trajectory. The coordinates of the foot placements are provided as inputs to the algorithm, along with the step duration and the height of the plane on which we constrain the COM to move along.

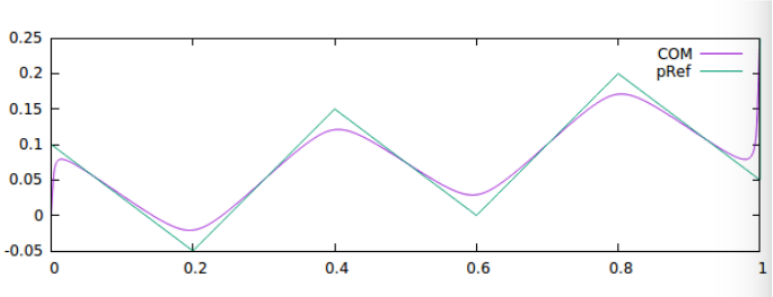


Fig. 1. An example of a trajectory generated for a diagonal walk of 6 steps. The X axis is the direction of motion of the robot.

The swing leg trajectory is computed from a linear interpolation between the previous and next touchdown positions.

Before the trajectory is given to Inverse Kinematics, another controller modifies it based on real time data from the Inertial Measurement Unit (IMU) - namely roll and pitch of the trunk. An offset to the trajectory is computed as a weighted sum of the robots inclination, its derivative, and its integral (a PID controller). The weights for this controller are precomputed using a genetic algorithm, by running numerous simulations.

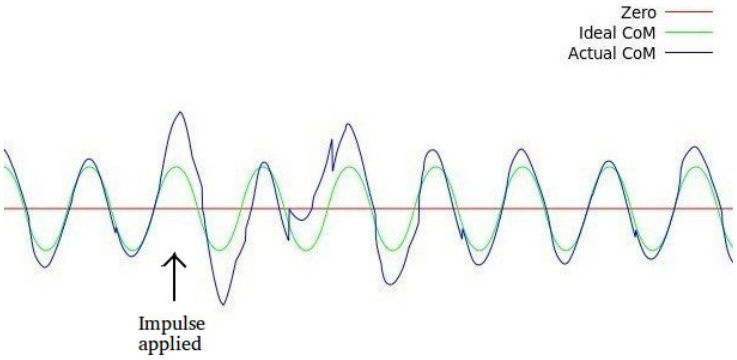


Fig. 2. A graph of the CoM trajectory showing our push-recovery algorithm in action.

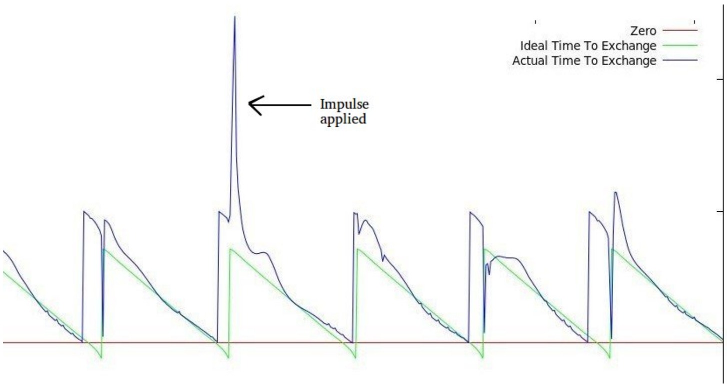


Fig. 3. A graph of the time to support exchange for the same push as in the above figure.

In addition to this, another controller dynamically varies the speed of the robot, causing it to vary its step time (hasten or delay) as required to maintain stability. This is done by varying the sleep periods between subsequent control cycles of the code. The magnitude of the delay is computed by comparing the time to step exchange predicted by Linear Inverted Pendulum Model (LIPM) equations using the theoretical (ZMP controller) CoM with that computed using the actual CoM of the bot (obtained from sensors).

Refer to [2] for more details.

5 Computer Vision

In the context of soccer, the robot needs to detect several things to make decisions about the play.

5.1 Object and Field Detection

For field detection, convex hulls of green color are chosen, ranked on the basis of area and merged if in close (relative to area) proximity, to merge the parts of the field separated by field lines. Potential corner points are also detected using the corners of the convex hulls.

Object detection, like the ball and the goalposts, is achieved using support vector machine classifiers trained on the Histogram of Oriented Gradients (HOGs) of the respective objects. Since the process is computationally expensive, once detected, the bounding rectangles are passed to a CamShift tracker, which searches for new positions of the objects only in their neighbourhoods from the previous frames. This allows real-time tracking of multiple objects with relative ease.

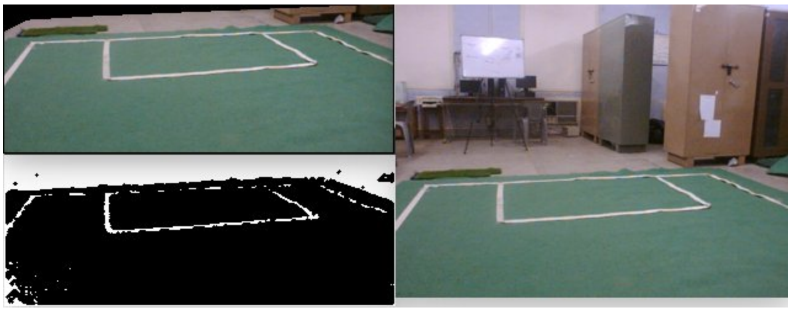


Fig. 4. Extraction of field from a frame

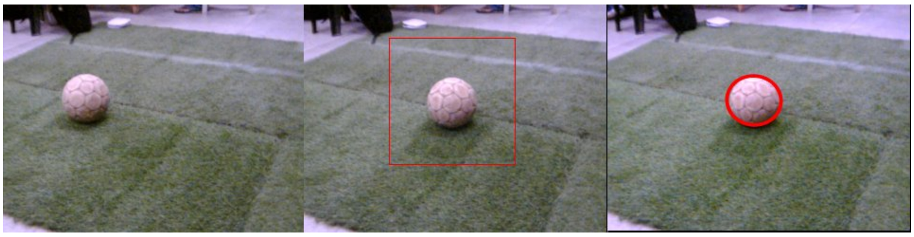


Fig. 5. Ball detection using HOG and handing over to CamShift

5.2 Localization

Once information about the goalposts, fieldline corners and ball is established, it is then passed to the localization module. First, the position of the robot with respect to the field is established. The locations of the goalposts and corners come into consideration while doing this. Inverse Perspective Mapping (IPM) is used to convert local coordinates (relative to the robot) into global coordinates (relative to the field). Monte Carlo Localization is then used to determine the position and orientation of our robot. The particle filter is updated at every frame based on the motion model (internal) and the model generated through observation.

Then, the location of the ball and other robots is calculated, also using IPM, and this information is passed to the behaviour module.

6 Behaviour Model

The behavior model employed is an XABSL-based architecture. We use XABSL to visualize our control flow and decision making as a finite state machine. Programs for each of the processes, such as walk, image processing, and localization are run on the pthreads generated for each module by the XABSL framework. Decision making is primarily controlled by flags which are implemented as mutex variables shared with each pthread.[4]

6.1 Path Planning

After features are obtained from localization, the behaviour controller sends the position information of the obstacles, the ball, and the goal to the path planner. Instead of a direct grid-mapping and graph planning approach (such as A*), we use a geometric online trajectory planner. Obstacles are modelled as circles for simplicity, and tangents and arcs are used to navigate around the obstacles while taking the least amount of time. Initial and final orientation of the robot is also taken care of the path planner by introducing virtual obstacles.

Although the planner assumes that the obstacles are static, the path planner can process new localization information quickly, and hence the path is updated at the end of every footstep. This ensures that the dynamic nature of other robots in the field is taken into account.

For further details, refer to our paper on this method.[3]

References

1. Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa (2003). "Biped walking pattern generation by using preview control of zero-moment point." IEEE International Conference on Robotics and Automation
2. Missura, M. (2016). Analytical and learned footstep control for robust bipedal walking (Doctoral dissertation, University Of Bonn, Bonn, Germany).
3. Kaustubh Nawade and V. Aditya (2015): Geometrical approach to On line Trajectory generation, Obstacle avoidance and Footstep planning for a Humanoid Robot. In the 15th IEEE-RAS International Conference on Humanoid Robotics (Humanoids 2015)
4. Philipp Allgeuer and Sven Behnke (2013): Hierarchical and State-based Architectures for Robot Behavior Planning and Control In: Proceedings of 8th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots. Atlanta, USA
5. Pratyush Kar and Archit Jain (2016): Effective localization of humanoid with fisheye lens using field line detection. In Intelligent Robot Systems (ACIRS), Asia-Pacific Conference
6. Jia, Hui-Xing, and Yu-Jin Zhang (2007). "Fast Human Detection by Boosting Histograms of Oriented Gradients." Fourth International Conference on Image and Graphics (ICIG 2007)