Humanoid Throws Inaugural Pitch at Major League Baseball Game: Challenges, Approach, Implementation and Lessons Learned

Daniel M. Lofaro
Electrical and Computer Engineering
Drexel University
Philadelphia, PA, USA
dan@danLofaro.com

Paul Oh
Mechanical Engineering
Drexel University
Philadelphia, PA, USA
paul@coe.drexel.edu

Abstract— **Three different approaches of having a full-size humanoid throw the first pitch at a Major League Baseball game are tested and implemented. The approaches include kinematic mapping using a motion capture system to capture a human's throwing motion then mapping that to a full-size humanoid. The second method is a fully automated approach that uses the sparse reachable map to provide viable full body throwing trajectories to provide the end effector with the desired velocity. The third approach borrows from the animation industry. The key-frames of the desired trajectory are constructed by hand. The time between each key-frame is defined by the user. Interpolation methods are used to smoothly move between key frames while limiting the jerk. Each method is analyzed and tested in simulation and on physical hardware. The full-size humanoid used is the Hubo series robot. Based on the latter tests one method was chosen to successfully throw the ceremonial first pitch at a Major League Baseball game in April 2012.**

1. Introduction

In early February 2012 the director of the Philadelphia Science Festival asked the Drexel Autonomous Systems Lab (DASL)[1] if they could have their full-size humanoid Jaemi Hubo throw the ceremonial first pitch at the second annual *Science Night at the Ballpark*. On April 28th, 2012 Hubo successfully threw the first pitch at the Philadelphia Phillies vs. Chicago Cubs game in front, see Fig. 1. There were 45,196 fans at the game and thousands more were watching it on telivision acording to USA Today.

Hubo was the first full-size humanoid to throw the inaugural pitch at a Major League Baseball game. This task poses challenges in the area of fully-body locomotion, coordination and stabilization that must be addressed. This paper describes how the latter was done via the analyses/tests of three different approaches and the resulting final design. Section 2 gives a brief introduction to work already done in the field as well as states the requirements for the pitch. Section 3 describes the three different methods tested where: Section 3.1 discusses the balancing methods and criteria used. Section 3.2 describes the human-robot kinematic mapping approach that uses a motion capture system to capture a human's throwing motion then mapping that to a full-size humanoid. Section 3.3 describes a fully automated approach that uses the sparse reachable map (SRM) to provide viable full body throwing trajectories with the desired end effector velocity[1]. Section 3.4 describes the final method explored which is based on key-frame trajectories. Section 4 compares the tests and analyses of each of the methods. Section 5 describes the finial design in detail and the modifications needed to make the robot's pitch reliable. Section 6 gives final thoughts and possible improvements for future years.

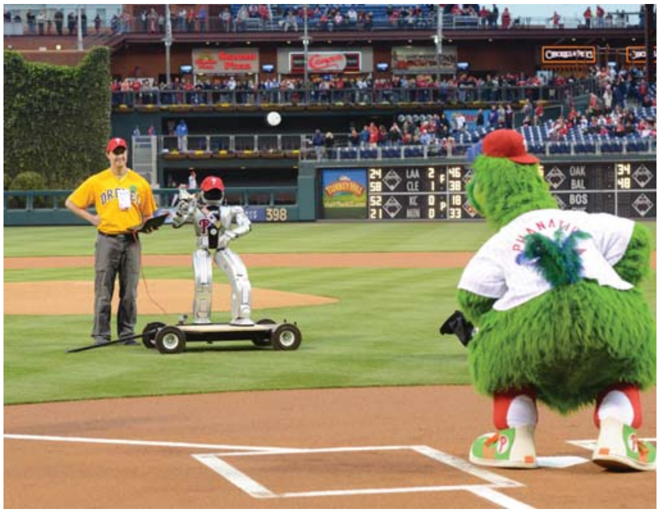

Fig. 1. Hubo successfully throwing the first pitch at the second annual Philadelphia Science Festival event Science Night at the Ball Park on April 28th, 2012. The game was between the Philadelphia Phillies and the Chicago Cubs and played at the Major League Baseball stadium Citizens Bank Park. The Phillies won 5-2. Video of the pitch can be found at http://danlofaro.com/urai2012/#pitch

2. Background and Objectives

The objective is to have our humanoid Jaemi Hubo throw a regulation Major League Baseball ball from the pitchers mound across home plate, a distance of 60.5 feet (18.4 m). The regulation Major League Baseball ball has a circumference between $9-9\frac{1}{4}$ inches (229-235 mm) and weights $5-5\frac{1}{4}$ ounces (142-149 g)[2] When children are asked to throw the first pitch it is typical for them to stand halfway between

*This project was supported by a Partnerships for International Research and Education (PIRE) #0730206, and Major Research Infrastructure Recovery and Reinvestment (MIRR) #CNS-0960061 sponsored by the the U.S. National Science Foundation (NSF).

[1] Drexel Autonomous Systems Lab: http://dasl.mem.drexel.edu

[2] MLB Official Rules: http://mlb.com

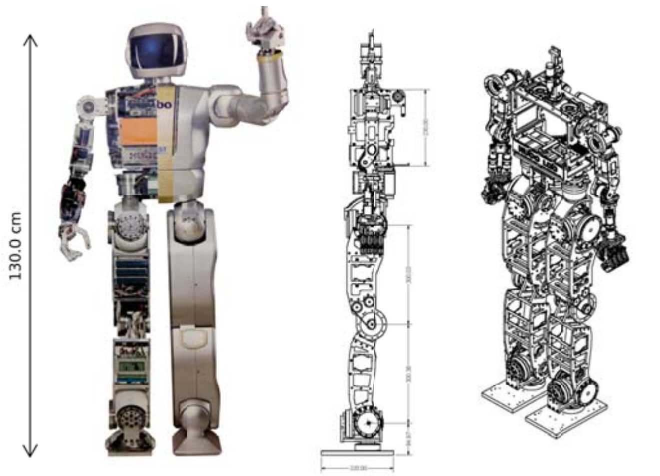

Fig. 3. Jaemi Hubo is a 130 cm tall 37 kg 40-DOF highly articulated, high-gain position controlled, full-size humanoid.

the pitchers mound and home plate. The robot used to throw the pitch, Jaemi Hubo, stands 130 cm tall, the average height of a ten year old child. To properly fit the stature of the robot the pitch will be given from half the regulation distance. Using the well known projectile motion formulation it is determined that the robot must have an end effector velocity of 9.47 $\frac{m}{s}$ at 45^o when it releases the ball in order for it to cross the plate.

Along with the technical challenges the primary objective sought to anthropomorphize robotic pitching. As such Jaemi Hubo was outfitted with a uniform shirt and cap. In addition Jaemi gestures the crowd by waving its hand when entering and exiting the field. Thus, beyond simply pitching, the challenge was to engage the audience hence the *Science Night at the Ballpark*.

Completing the technical objective with a humanoid consists of two major parts: 1) end-effector velocity control and 2) balance/stabilization. There are many examples of throwing/pitching machines made by commercial companies such as Louisville Slugger^{TM}, Jugs^{TM}, and Atec^{TM} to name a few. The PhillieBot[3] made by the GRASP Lab at the University of Pennsylvania was the robot that threw the first pitch at the first annual Philadelphia Science Festival *Science Night at the Ballpark* in 2011. The PhillieBot consisted of a ≤ 7 DOF arm with a pneumatic wrist actuator to increase end-effector velocity at the release point. The arm was attached to a wheeled mobile platform. All of these robots/devices are well planted to the ground/base to ensure stability. None of the latter robots is are anthropomorphic thus limiting their audience engagement.

Kim et al. [2] takes the research to the next level with finding optimal overhand and sidearm throwing motions for a high degree of freedom humanoid computer model. The model consists of 55-DOF and is not fixed to mechanical ground or a massive base. Motor torques are then calculated to create both sidearm and overhand throws that continuously satisfies the zero-moment-point stability criteria [3].

The highly articulated 40-DOF full-size humanoid Jaemi Hubo (Fig. 3) is the platform focused on in this work. Jaemi Hubo is a high-gain, position-controlled biped humanoid weighing 37 kg and standing 130 cm tall. It is designed and made by Dr. Jun-Ho Oh director of the Hubo Lab at the Korean Advanced Institute of Science and Technology (KAIST). Jaemi has been located at the Drexel Autonomous Systems Lab (DASL) at Drexel University since the Fall of 2008. DASL has extensive experience with the Jaemi Hubo KHR-4 platform in key areas needed to complete this work. Balancing was explored when developing a real-time zero moment point (ZMP) preview control system for stable walking [4]. A full-scale safe testing environment designed for experiments with Jaemi Hubo was created using DASL's Systems Integrated Sensor Test Rig (SISTR) [5]. Additionally all algorithms are able to be tested on miniature and virtual versions of Jaemi Hubo prior to testing on the full-size humanoid through the creation of a surrogate testing platform for humanoids [6].

3. Methodology

3.1 Balance and Stability

Each of the methods used have to be stable through the motion in order for the system to be stable (i.e. not to fall down). The well known zero-moment-point (ZMP) criteria is what each method must adhere to in order to stay statically stable[7]. To handle perturbation an active balance controller was added. The active balance controller is applied on top of the pre-defined trajectories. Hubo is modeled as a single inverted pendulum with the center of mass (COM) located at length L from the ankle. The compliance of the robot is composed of a spring K and a damper C, see Fig. 4. An IMU located at the COM gives the measured orientation.

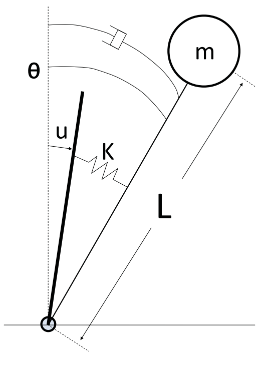

Fig. 4. Hubo modeled as a single inverted pendulum with COM located a distance L from

The dynamic equation of the simplified model is assumed to be the same in both the sagittal and coronal plane.

$$mL^2\ddot{\theta} + C\dot{\theta} - K\theta = Ku \tag{1}$$

This can be linearized and made into the transfer function:

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{\frac{K}{mL^2}}{s^2 + \frac{C}{mL^2}s + \frac{K-mgL}{mL^2}} \tag{2}$$

[3] PhillieBot Video: http://youtu.be/ShId-vZ-ZEY

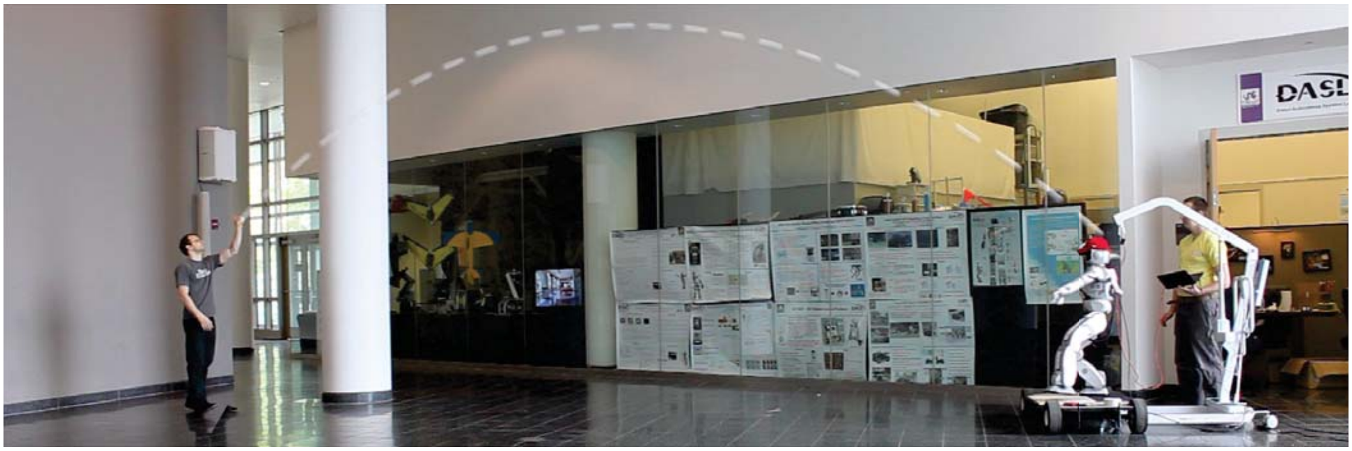

Fig. 2. Frame overlay of the Hubo throwing overhand a distance of 10 m (32.8 feet) with a release angle of 40^o and a tip speed of 10 $\frac{m}{s}$. Captured at 20 fps with a shutter speed of 1/30 sec. Each of the white dashes of in the image is the actual baseball as picked up by the video camera.

Prior work on the model and controller for the Hubo by Cho et. al. calculated K=753 $\frac{Nm}{rad}$ and C=18 $\frac{Nm}{sec}$ using the free vibration response method[8].

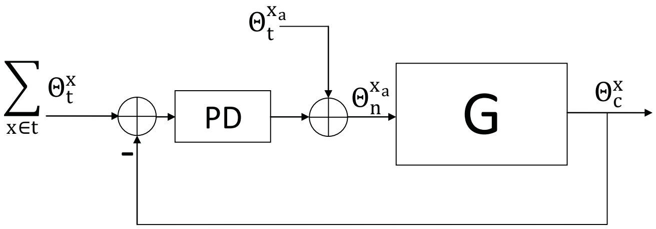

Fig. 5. Block diagram of the balance controller used to balance Hubo in this work.

The control law is as follows

$$\theta_n^{x_a} = \theta_t^{x_a} + \left(K_p^x + sK_d^x\right)\left(\sum_{x\in t}\theta_t^x - \theta_c^x\right) \tag{3}$$

Where θ_t is the desired trajectory of the lower body (pitch or roll), x denotes pitch or roll and x_a denotes pitch or roll on the ankle. θ_c is the orientation of the center of mass in the global frame. θ_n is the resulting trajectory. K_p and K_d are the proportional and derivative gains. The resulting control allows for a stable stance even with perturbations from upper body motions.

3.2 Human to Humanoid Kinematic Mapping

Motion capture (MoCap) systems can be used to generate human-like motions and map those motion to humanoids[9], [10]. Fig. 6 shows the Hubo's kinematic structure (left) and the human (MoCap) kinematic structure(left). The human has 3-DOF at each joint while the humanoid has limited DOF at each corresponding joint. Some of the challenges in mapping between the human kinematic structure (from MoCap) to a humanoid's kinematic structure are:

- The difference in the total degree of freedom (DOF).
- The difference in the kinematics descriptions.
- The different Kinematic constraints.

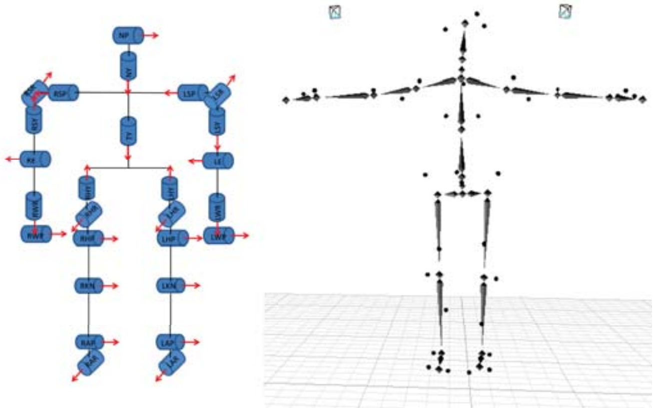

Fig. 6. Left: Jaemi Hubo joint order and orientation using right hand rule. Right: Motion capture model of human figure

Our approach is to: a) Chose a set MoCap model. b) Preform motions where the pitch motions are decoupled (roll and yaw stays constant), avoids singularities and robot joint position limitations. c) Combine joint values for near by joints (reduce the model to the same DOF as the robot). d) Some tests require the addition of static offsets to joints to ensure the zero-moment-point (ZMP) criteria is satisfied as stated in Section 3.1

To test this method we used a human subject to throw a ball using upper and lower body movements. All motions were in the sagittal plane to keep pitch joints decoupled. To avoid the robot's joint limit of ±180^o an underhand throwing motion was used. Fig. 7 shows the human throwing the ball and the robot throwing the ball to the mapped motion of the human.

To ensure balance throughout the motion the balance controller as described in Section 3.1 was applied and the static ZMP criteria was checked for the entire trajectory. The human subject threw the ball approximately eight feet (244 cm). The mapping of the latter motion caused the robot to throw the ball approximately five feet (152 cm). The discrepancy comes from the proportional difference in limb length from the human to the robot.

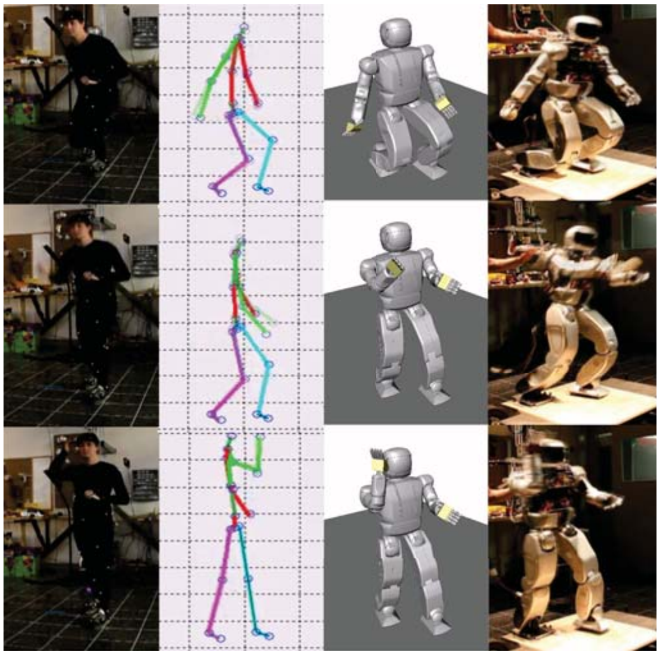

Fig. 7. (Left to Right): (1) Human throwing underhand in sagittal plane while being recorded via a motion capture system. (2) Recorded trajectory mapped to high degree of freedom model. (3) High degree of freedom model mapped to lower degree of freedom OpenHUBO. (4) Resulting trajectory and balancing algorithm run on Hubo.

3.3 Throwing Using Sparse Reachable Map

A Sparse Reachable Map (SRM) is used to create a collision free trajectories while having the end-effector reach a desired velocity as discribed in Lofaro et. al.[1]. The SRM has been shown to be a viable method for trajectory generation for high degree of freedom, high-gain position controlled robots. This remains true when operating without full knowledge of the reachable area as long as a good collision model of the robot is available. The end-effector velocity (magnitude and direction) is specified as well as a duration of this velocity. The SRM is created by making a sparse map of the reachable end-effector positions in free space and the corresponding poses in joint space by using random sampling in joint space and forward kinematics. The desired trajectory in free space is placed within the sparse map with the first point of the trajectory being a known pose from the original sparse map.

$$L_d(0) \in SRM \tag{4}$$

$L_d(0)$ is known both in joint space and in free space. The Jacobian Transpose Controller method of inverse kinematics as described by Wolovich et al.[11] is then used to find the subsequent joint space values for the free space points in the trajectory.

$$q_1 = q_0 + \dot{q}_0 = q_0 + kJ^T e|_{x_0}^{x_1} \tag{5}$$

Where q_0 and x_0 is the current pose and corresponding end-effector position respectively. q_1 is the next pose for the next desired end-effector position x_1. Each desired end-effector position x must be within a euclidean distance d (user defined) from any point in the SRM.

$$min\left(|x - SRM|\right) < d \tag{6}$$

If one of the points in x fails this criteria a new random point is chosen for $L_d(0)$ and the process is repeated.

Each pose in the trajectory is checked against the collision model to guarantee no self-collisions. The collision model is based on the OpenRAVE model of the Hubo platform called OpenHUBO, see Fig 8.

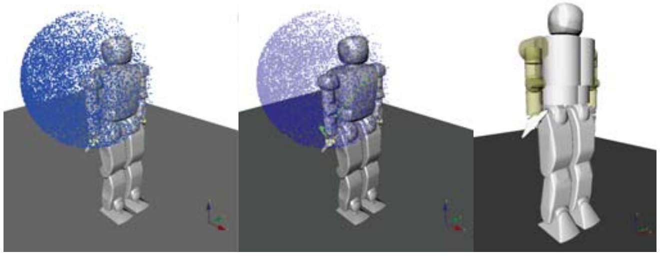

Fig. 8. OpenRAVE model of Hubo KHR-4. Left: Model with SRM of right arm. Center: SRM (blue) with setup and velocity phase trajectories (green) Right: Collision Geometry[1]

The commanded trajectory produces the desired velocity of 4.9 m/s at 60^o. This was then tested on the OpenHUBO and on the Jaemi Hubo platform, Fig 9 and Fig 9 respectively.

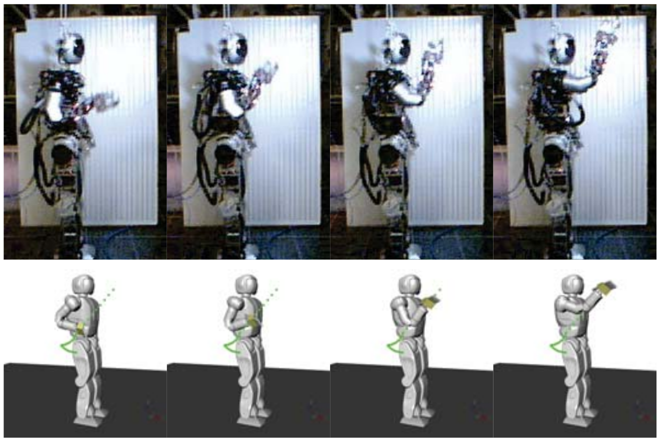

Fig. 9. (Top Row) Jaemi Hubo and (Bottom Row) OpenHUBO running the throwing trajectory immediately after the setup phase is completed. x_0 is top left. Frames are read left to right and have a Δt of 0.15 sec[1]

To ensure balance throughout the motion the balance controller as described in Section 3.1 was applied and the static ZMP criteria was checked for the entire trajectory. This method worked as desired. In approximately 10% of the tests one or more joints would over torque and shutdown. This is due to the system not taking the robots power limitations into account.

3.4 Key-Frame Motion

Key-frame motion profiles for humanoids borrows from the animation industries' long used techniques. When making an animation the master artist/cartoonist will create

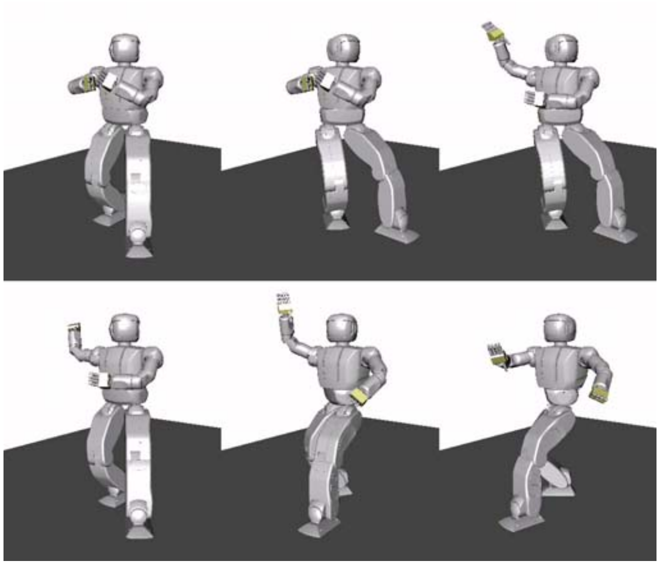

Fig. 10. OpenHUBO using key-frame based method for throwing trajectory creation. Frames are read from top left to bottom right. Video of the above trajectory can be found at http://danlofaro.com/urai2012/#keyframe

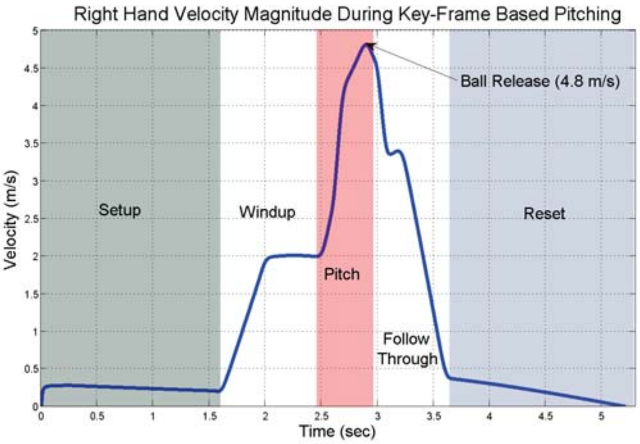

Fig. 11. Velocity vs. Time graph showing the magnitude of the end-effector's velocity for the key-frame based throwing motion. The six different stages of pitching are also shown. Setup: move from the current position to th throw stance. Windup: end effector starts to accelerate from the throw stance and move into position for the start of the pitch state. Pitch: end effector accelerates to release velocity. Ball Release: the ball leaves the hand at maximum velocity (4.8 $\frac{m}{s}$) at an elevation of 40^o from the ground. Follow Through: reducing velocity of end effector and all joints. Reset: moves to a ready state for anther throw if needed.

Table 1. Comparison between the three different methods of end effector (EEF) velocity control

	EEF Velocity ($\frac{m}{s}$)	Joint Failure (%)	Throw Overhand (y/n)	Stable (y/n)
MoCap	4.0	0	n	y
SRM	4.9	10	y	y
Key-Frame	4.8	0	y	y

the character in the most important (or key) poses. The apprentice will draw all of the frames between the key poses. We borrowed this technique when we: posed the robot in the desired pose, record the values in joint space, and make a smooth motion between poses. In place of the apprentice, forth order interpolation methods were used to make smooth trajectories between poses. Forth order interpolation was used in order to limit the jerk on each of the joints. The resulting trajectory is a smooth well defined motion as seen in Fig. 10.

To ensure stability throughout the motion the balance controller as described in Section 3.1 was applied and the static ZMP criteria was checked for the entire trajectory. The resulting end effector velocity was 4.8 $\frac{m}{s}$ at the release point. Fig. 11 shows the plot of the magnitude of the end effector's velocity. It should be noted that at the instance of release the velocity vector is at an elevation of 40^o from the ground.

4. Method Comparison

All three methods described are stable during the throwing motion and can successfully throw a baseball. Table 1 shows the end-effector (EEF) velocity, joint failure rate, overhand or underhand throwing and stability of the three different methods of end-effector velocity control used; human-robot joint mapping (MoCap), SRM based and key-frame.

The SRM technique worked well however the large jerk on each of the joints created large torques caused the motor controller to over torque/current and shutdown 10% of the time. The pitch needs to work 100% of the time thus this method is not well suited for the event. Both the human-robot joint mapping via MoCap and key-frame based methods consistently worked and stayed stable. A secondary objective is to have the throwing motion be overhand like a standard Major League Baseball pitcher. Due to the constraints placed on the joint mapping of the MoCap method in Section 3.2 an over arm throw would be impossible to preform due to the robot's $\pm 180^o$ joint limitations. The key-frame method was chosen as the method to throw the ball.

5. Final Design

The final goal is to have an end-effector velocity of 9.47 $\frac{m}{s}$ at 45^o. The key-frame method was tested to throw at 4.8 $\frac{m}{s}$. To increase the end-effector velocity the upper body motion was kept unchanged but the lower body added a stepping motion with its legs. The stepping motion consists of lifting the left foot up, pushing forward with the right and move the left forward 10 cm. Stepping with your non-dominant foot, and pushing with the dominant, when throwing overhand, is common practice to increase the distance you can throw a ball. Jaemi Hubo throws with its right hand and steps with its left. This increased the end-effector velocity from 4.8 $\frac{m}{s}$ to 7.1 $\frac{m}{s}$. Fig. 12 shows the stepping motion of the robot.

The addition of pushing off with the right foot and stepping forward introduced two problems. 1) The ZMP criteria is not satisfied throughout the motion and 2) the right foot would slip when pushing its body forward. To avoid slip *hook and loop* was paced on the bottom of the right foot (dominant) and on the throwing platform. This did not permanently attach the robot to the platform but it did allow for more friction between the foot and the ground. This allowed the balancing controller to function adequately

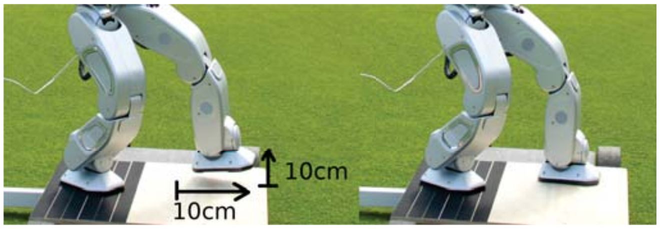

Fig. 12. (Left) Hubo stepping 10 cm up and forwards while pushing with dominant foot. (Right) Hubo lands with non-dominant foot. As a result the end effectors velocity is increased by 2.3 $\frac{m}{s}$.

for the short step and maintain stability. The platform was added to ensure a more consistent ground for the robot to balance on than the baseball field can inherently provide.

An additional 2.5 $\frac{m}{s}$ was needed to give a proper throw. Borrowing from the GRASP Lab and their high powered pneumatic wrist on their PhillieBot, a spring loaded mechanism was added to Hubo's wrist, see Fig. 13. The addition of this mechanism allowed the robot to achieve an end-effector velocity magnitude of 10 $\frac{m}{s}$. Fig. 2 shows a frame overlay of the the Hubo throwing a regulation baseball 10 m (32.8 feet). Fig. 1 shows the same throw at Citizens Bank Park on April 28th, 2012.

6. Conclusion

The creation of a reliable humanoid throwing method was completed. This allowed the Hubo to complete its goal of becoming the first full-size humanoid to throw the first pitch at a Major League Baseball game. Anthropomorphizing the robotic pitch was done via outfitting Jaemi with a uniform shirt and cap. Hubo also gestured to the crowd via waving during its entrance and exit of the field to further engage the audience at the *Science Night at the Ballpark*. Three methods of creating a viable pitch were explored and tested: (1) human-robot mapping via motion capture, (2) sparse reachable map based trajectory generation and (3) a key-frame based method. Though the key-frame based method was the simplest it also proved to be the most viable. The balancing controller ensured stability of the system throughout the throw. The addition of a high powered wrist and having the robot step forward when it throws created conditions for a successful and reliable pitch. The latter methods answered the challenges posed in the area of full-body locomotion, coordination and stabilization.

Prior full-size humanoid throwing methods such as the work by Kim et al. [2] were only shown in simulation. This paper's set of experiments and demonstrations shows the realization of these same tasks in the physical world. Throughout all of the experiments the system was manually aimed prior to each pitch. Each of the robot's pitches are accurate however a key factor in off-target or *wild* pitches is improper aiming during setup. The addition of feedback via a targeting system similar to that of Hu et al. [12] would remove the need for manual targeting of the robot and will be incorporated in the next revision of the pitching Hubo.

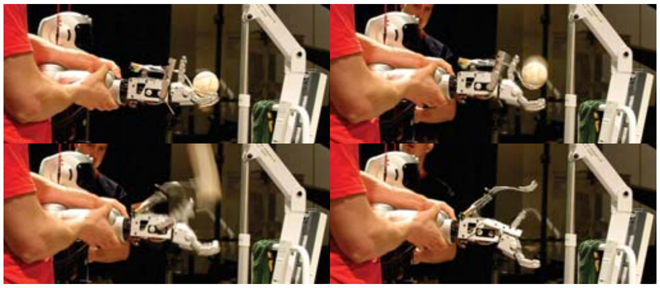

Fig. 13. Spring loaded mechanism test launching the baseball. Top-Left: Pre-launch. Top-Right/Bottom-Left: Launch. Bottom-Right: Pos-launch. The mechanism added 3.0 $\frac{m}{s}$ to the end-effector velocity at its release point.

This work is a step towards full-size humanoids performing fast and accurate full body tasks. Though one might perceive robots to be *better* than people (faster, more accurate, etc.) the reality is that the field of humanoid robotics is still in its infant state and Hubo is only the size of a ten year old child. Having Hubo complete this task shows that full-size humanoids can perform some tasks at a ten year old's level. More information and media about this event is available on this papers home page http://danlofaro.com/urai2012/

References

[1] D. Lofaro, R. Ellenberg, P. Oh, and J. Oh, "Humanoid throwing: Design of collision-free trajectories with sparse reachable maps," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012.

[2] J. Kim, "Motion planning of optimal throw for whole-body humanoid," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, dec. 2010, pp. 21 –26.

[3] M. Vukobratovic, "How to control artificial anthropomorphic systems," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 5, pp. 497 –507, sept. 1973.

[4] Y. Jun, R. Ellenberg, and P. Oh, "Realization of miniature humanoid for obstacle avoidance with real-time zmp preview control used for full-sized humanoid," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, dec. 2010, pp. 46 –51.

[5] R. Ellenberg, R. Sherbert, P. Oh, A. Alspach, R. Gross, and J. Oh, "A common interface for humanoid simulation and hardware," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, dec. 2010, pp. 587 –592.

[6] R. Ellenberg, D. Grunberg, P. Oh, and Y. Kim, "Using miniature humanoids as surrogate research platforms," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, dec. 2009, pp. 175 –180.

[7] M. Vukobratovic and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, no. 12, pp. 1 – 37, 1972.

[8] B.-K. Cho, S.-S. Park, and J. ho Oh, "Controllers for running in the humanoid robot, hubo," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, dec. 2009, pp. 385 –390.

[9] Q. Huang, Z. Peng, W. Zhang, L. Zhang, and K. Li, "Design of humanoid complicated dynamic motion based on human motion capture," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 3536 – 3541.

[10] S.Pollard, J. Hondgins, M.J.Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *In Proc. of IEEE International Conference on Robotics and Automation*, 2002.

[11] W. A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," in *Decision and Control, 1984. The 23rd IEEE Conference on*, vol. 23, dec. 1984, pp. 1359 –1363.

[12] J.-S. Hu, M.-C. Chien, Y.-J. Chang, S.-H. Su, and C.-Y. Kai, "A ball-throwing robot with visual feedback," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 2511 –2512.