Archetecture for High Degree of Freedom Complex Control Systems

A Thesis

Submitted to the Faculty

of

Drexel University

by

Daniel Marc Lofaro

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Electrical and Computer Engineering Engineering

June 2013

Table of Contents

List of Tables

List of Figures

0.1 Introduction

0.1.1 Platform

The Hubo2 Plus series robot is a 130 *cm* (4' 3") tall, 42 *kg* (93 *lb*) full-size humanoid commonly refereed to as Hubo. The Hubo series was designed and constructed by Prof Jun-Ho Oh at the Hubo Lab in the Korean Advanced Institute of Science and Technology (KAIST) in Daejeon, South Korea [**?**]. Hubo has 2 arms, 2 legs and a head making it anthropomorphic to a human. It contains 6 degrees of freedom (DOF) in each leg, 6 in each arm, 5 in each hand, 3 in the neck, and 1 in the waist; all totaling 38 DOF. All joints of the major joints are high gain PID position controlled with the exception of the fingers. The fingers are open-loop PWM controlled. The sensing capability consists of a three axis force-torque (FT) sensor on each leg between the end of the ankle and the foot as well as between the arm where it connects to the hand. Additionally it has an inertial measurement unit (IMU) at the center of mass and accelerometers on each foot. The reference commands for all of the joints are sent from the primary control computer (x86) to the individual motor controllers via two Controller Area Network (CAN) buses. This is the same communications bus found in most modern motor vehicles. There are currently eight Hubo's functioning in the United States as of December 2012. Four reside at Drexel University and one at Georgia Tech, Purdue, Ohio State and MIT. Jaemi Hubo is the oldest of the Hubos in America and has been at the Drexel Autonomous Systems Lab[1] (DASL) since 2008 [1]. Fig. 1 shows the major dimensions of Hubo.

A full-scale safe testing environment designed for experiments with Jaemi Hubo was created using DASL's Systems Integrated Sensor Test Rig (SISTR) [2]. 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

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

testing platform for humanoids [3].

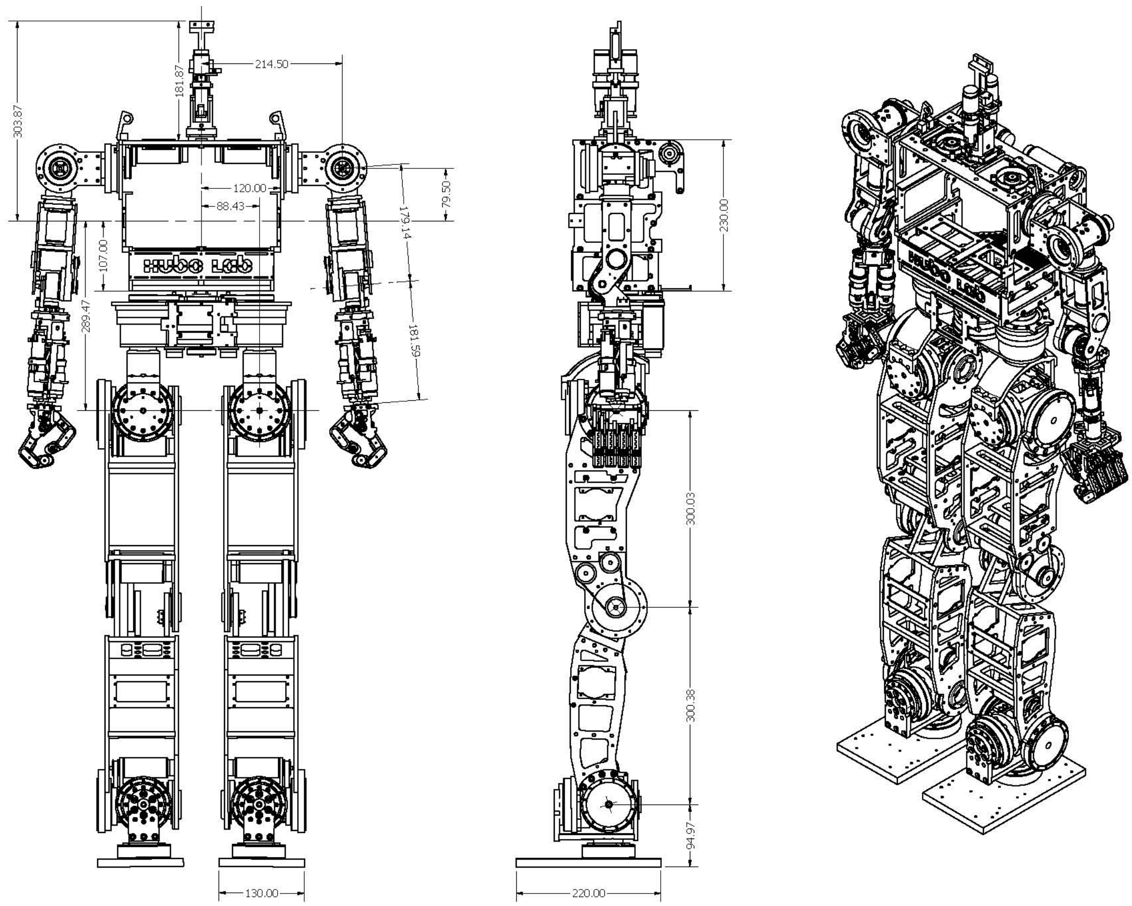

Figure 1: Hubo2 Plus platform: 38 DOF, 130 cm tall full-size humanoid robot weighing 37 kg.

0.1.2 Background

Kinematic Planning

Kinematic planning focuses on creating and testing valid trajectories for series kinematic manipulators. The focus of this research is on high degree of freedom (DOF), high-gain, position controlled mechanisms. The works are chosen as it pertains to end-effector velocity control. Throwing and hitting are examples of end-effector velocity control. The goal is to have the end-effector moving at a specific rate in a specific direction. In most cases it demands whole-body coordination to achieve a desired end-effector velocity. Whole-body coordination is different for planted robots and un-planted robots.

Planted robots are robots where the base is attached to the ground or the base is significantly more massive then the manipulator. Planted robots do not have to worry about balance consternates.

Un-planted robots are robots that have an manipulator that is not significantly ligher then the base. In addition the robot is not physically attached to the ground. This results in the robot needing to satisfy balance constraints. In the static case if the robot satisfies the zero moment point (ZMP) criteria it will remain stable [4]. When the manipulator moves quickly, as in the case of pitching or throwing, such upper-body motions if not coordinated with the lower-body, can cause the humanoid to lose balance.

End-Effector Velocity Control

End-effector velocity control (EEVC) is the act of moving your manipulator at a given speed through space at a given velocity. EEVC is being looked at as the mass of the end-effector does not change. Thus by controlling the velocity we also control the inertia. In addition I will be exploring EEVC as it pertains to manipulating objects. Through my research I have found that end-effector velocity control can be broken up into four major categories: *Time and location sensitive*, *location sensitive*, *time sensitive*, and *time and location insensitive.*

Time and Location Sensitive: If your velocity control is time and location sensitive it means that your end effector needs to have a given velocity at a specific time in a specific location or the task fales. Hitting a baseball with a bat is an example of *time and location sensitive* EEVC. If the bat has the correct velocity but not at the correct time it will not hit the ball or the ball will not go in the desired place. The same goes for if it does not have the correct location but does have the correct velocity. It is important to note that the manipulator only has instantaneous control over the object at the instant of contact. Other examples include playing the piano, hitting a tennis ball with a racquet, a moving soccer ball with a foot or any other task that requires to *hit* a *moving* object.

Location Sensitive: If your velocity control is location sensitive it means that it only matters that the velocity occurs at a given location. The time it takes to reach that velocity will not effect the results. Hitting a nail with a hammer is a prime example of location sensitive EEVC. The nail is not moving but it does need to be hit in a given location with a given velocity. The vector of the velocity is determined by the required angle the nail needs to be hit at. In this example the nail is not time dependent and can be hit any time. Hitting it a $t = N$ or $t = N + 1$ will not effect the results. It is important to note that the manipulator only has instantaneous control over the object at the instant of contact. Other examples of location sensitive end-effector velocity control are hitting a golf ball with a club, hitting a pool ball with the cue, and other activities that require a given location and direction of manipulation but are not time dependent.

Time Sensitive: If the location were the end-effector achieves a given velocity is not required to complete the task but the time when it happens is required it is considered *time sensitive* EEVC. This means that the end-effector can move in any region it desired as long as the end effector achieves a given velocity at a given time. The end-effecter's velocity can be dependent on the location achieved but the location is an independent variable and the velocity is the dependent variable. It is important to note that the manipulator control over the object during the entirety of the motion. This typically means that the manipulator is holding the object until the release stage. An example of this is throwing a baseball to first base to get someone out. Throwing the ball side arm, over arm, or even underarm does not matter as long at it is released at the correct time with the correct velocity to get it ball to the first-baseman to get the runner out. Other example of time sensitive EEVC are any other instance where an object is thrown within a given time.

Time and Location Insensitive: If the location and the time of when the end-effector achieves a given velocity does not matter it is considered time and location insensitive. The end-effecter's velocity can be dependent on the location achieved but the location is an independent variable and the velocity is the dependent variable. In this case the manipulator has control over the object until the release stage. Examples of this would be pitching a baseball, bowling, throwing a grenade or horseshoes etc. Throwing is an example of when the end-effector's velocity holds a higher priority over the position.

Mechanisms with only a single degree of freedom are restricted to throwing in a plane. 2-DOF mechanisms are able to throw in R^3 space with the correct kinematic structure. Such a mechanism can choose its release point or its end-effector velocity but not both. Mechanisms containing 3 or more DOF with the correct kinematic structure are able to throw in R^3 and choose both the release point and the end-effector velocity simultaneously.

In recent work Mori et al. [5] has show his ability to control the translational velocity, angular velocity and direction in a 2-dimension plane independently with a single DOF mechanism. The only input is torque to the manipulator. The concept consists is to map the input torque that will change only one of the kinimatic variables and not the other two. This map is done over a given space and thus you can independently chose your translational and angular velocity as well as direction as long as it is in the valid search space. The manipulator and a search space example can be seen in Fig. 2.

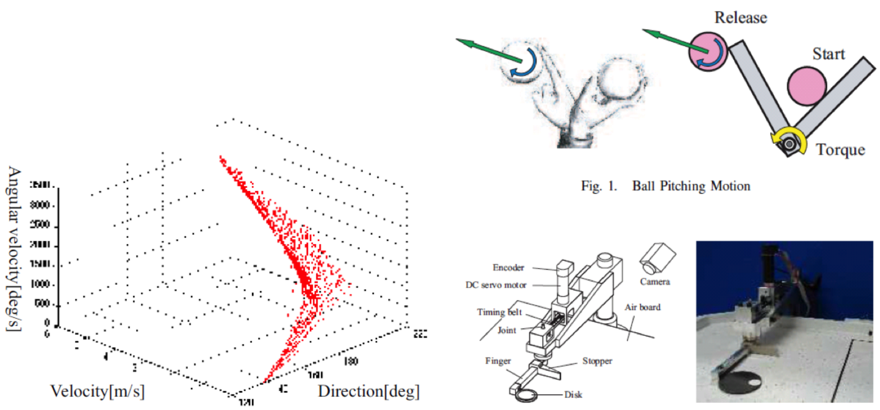

Figure 2: Map of the input torque that will change only one of the kinimatic variables and not the other two. This map is done over a given space and thus you can independently chose your translational and angular velocity as well as direction as long as it is in the valid search space.

Low degree of freedom throwing machines/robots are common. Typical throwing robots have between one and three degrees of freedom (DOF) [5–9]. All of these mechanisms are limited to throwing in a plane. Sentoo et al. [10] achieved an end-effector velocity of 6.0 m/s and can throw in R^3 space. This is done via the use of a planted robot arm made by Barret Technology Inc consisting of 4-DOF with a 360^o rotation base yaw actuator.

These low degree of freedom throwing robots are either physically attached/planted to the mechanical ground or have a base that is significantly more massive then the arm.

Haddadin et al.[11] used their 7-DOF arm and a 6-DOF force torque sensor with standard feedback methods to dribble a basket ball. In addition Zhikun et al. [12] used reinforcement learning to teach their 7-DOF planted robot arm to play ping-pong. Likewise Schaal et al. [13] taught their high degree of freedom (30-DOF) humanoid to hit a tennis ball using an on-line special statistical learning methods. Visual feedback was used in the basketball throwing robot by Hu et al. [14] achieving accuracy of 99%. All of the latter robots were fixed to the ground to guarantee stability.

Kim et al. [15, 16] 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 [17].

Balancing: Zero-Moment-Point (ZMP)

The past years of research in humanoids robotics has resulted in a stability criteria that must be followed for bipedal robots to stay stable. This is known as the Zero Moment Point criteria commonly referred to as ZMP [18]. ZMP is ubiquitous in the humanoid robotics community. The ZMP criteria states that a system is statically stable (balanced) if there is no moment acting on the connection between the end effectors touching the ground and the ground. This means that if the center of mass is over the support polygon there will be no moment. The support polygon is defined by the are formed by connecting the out most portions of the end effectors (typically feet) that are touching the ground and/or walls, rails etc. If the zero moment point, the location of the center of mass (COM) projected in the direction of gravity, is located within this support polygon then the system is considered statically stable. Fig. 3 gives an example of the zero moment point on a bipedal robot in a single support phase and a double support phase.

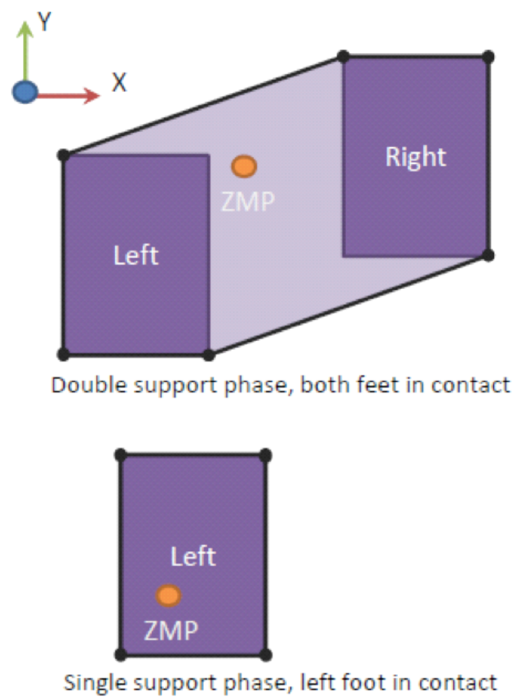

Figure 3: Example of the zero moment point on a bipedal robot in a single support phase (bottom) and a double support phase (top). If the zero moment point, the location of the center of mass (COM) projected in the direction of gravity, is located within this support polygon then the system is considered statically stable.

Single Support Phase: The single support phase of a bipedal robot is when a single foot is touching the ground. This creates a smaller support polygon.

Double Support Phase: The double support phase of a bipedal robot is when two feed of a bipedal robot are on the ground. This creates a larger support polygon. In addition there is a stable path that the ZMP can move from above one foot to the other. This allows the robot to guarantee stability while walking (static walking).

0.1.3 Vertical Leap

Bibliography

[1] D. M. Lofaro, R. Ellenberg, P. Oh, and J.-H. 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] 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 Int’l Conference on*, pp. 587 –592, Dec 2010.

[3] 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 Int’l Conference on*, pp. 175 –180, Dec 2009.

[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 Int’l Conference on*, pp. 46 –51, Dec 2010.

[5] W. Mori, J. Ueda, and T. Ogasawara, “1-dof dynamic pitching robot that independently controls velocity, angular velocity, and direction of a ball: Contact models and motion planning,” in *Robotics and Automation, 2009. ICRA ’09. IEEE Int’l Conference on*, pp. 1655 –1661, May 2009.

[6] N. Kato, K. Matsuda, and T. Nakamura, “Adaptive control for a throwing motion of a 2 dof robot,” in *Advanced Motion Control, 1996. AMC ’96-MIE. Proceedings., 1996 4th Int’l Workshop on*, vol. 1, pp. 203 –207 vol.1, Mar 1996.

[7] K. M. Lynch and M. T. Mason, “Dynamic nonprehensile manipulation: Controllability, planning, and experiments,” *Int’l Journal of Robotics Research*, vol. 18, pp. 64–92, 1997.

[8] T. Nakamura, “Search guided by skill in motion planning using dynamic programming,” in *Advanced Motion Control, 1996. AMC ’96-MIE. Proceedings., 1996 4th Int’l Workshop on*, vol. 2, pp. 711 –716 vol.2, Mar 1996.

[9] A. Sato, O. Sato, N. Takahashi, and M. Kono, “Trajectory for saving energy of a direct-drive manipulator in throwing motion,” *Artificial Life and Robotics*, vol. 11, pp. 61–66, 2007.

[10] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed throwing motion based on kinetic chain approach," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ Int'l Conference on*, pp. 3206 –3211, Sept 2008.

[11] S. Haddadin, K. Krieger, M. Kunze, and A. Albu-Schaffer, "Exploiting potential energy storage for cyclic manipulation: An analysis for elastic dribbling with an anthropomorphic robot," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ Int'l Conference on*, pp. 1789 –1796, Sept 2011.

[12] Z. Wang, C. H. Lampert, K. Mulling, B. Scholkopf, and J. Peters, "Learning anticipation policies for robot table tennis," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ Int'l Conference on*, pp. 332 –337, Dec 2011.

[13] S. Schaal, S. Vijayakumar, S. D'Souza, A. Ijspeert, and J. Nakanishi, "Real-time statistical learning for robotics and human augmentation," in *Int'l Symposium of Robotics Research (ISRR01)*, pp. 117–124, Springer, 2001.

[14] 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 Int'l Conference on*, pp. 2511 –2512, Oct 2010.

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

[16] J. H. and Kim, "Optimization of throwing motion planning for whole-body humanoid mechanism: Sidearm and maximum distance," *Mechanism and Machine Theory*, vol. 46, no. 4, pp. 438 – 453, 2011.

[17] M. Vukobratovic, "How to control artificial anthropomorphic systems," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, pp. 497 –507, sept. 1973.

[18] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *I. J. Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.