A Closed Loop Locomotion System for the Aldebaran Nao

Alex Baucom

***Abstract*— This paper presents two approaches to bipedal locomotion for use on the Aldebaran Nao as part of the RoboCup competition. The first is an approach that uses the Zero Moment Point and preview control to adjust the center of mass motion to stabilize the walk based on future inputs. The second approach uses real-time adaptation based on observations of the position and velocity of the center of mass to adjust the step size and gait timing to maintain stability and compensate for any external disturbances quickly. Finally, the two methods are compared and one is selected to be used for the Penn RoboCup team.**

I. INTRODUCTION

The UPennalizers is a team of students from Penn that participate in the RoboCup Standard Platform League (SPL). This standardization means that all teams use the same robot - the Aldebaran Nao - as their development platform and the competition involves writing the best software and algorithms for the robots to play soccer. The idea behind this is that the standardization of the platform will allow teams to work together and share their results with the community as we develop new and innovative algorithms for all the different features needed to play soccer effectively.

One such algorithm that is needed is a walking algorithm. In the past, the UPennalizers have relied upon the work of a post-doc who wrote a lot of the locomotion code that is currently in use. However, since he left recently and did not provide much in the way of documentation or details about his work, we have been forced to start from scratch and re-develop the locomotion algorithm for our robots.

Thankfully, the cooperative nature of the SPL means that we do not have to start from square one since there are a lot of papers out about various walking algorithms that have been implemented on the Nao. So, we decided to do a thorough review of the different approaches and find one that would be the best to implement for our team.

Although many papers and reports were reviewed during this research, this paper will detail only the two best algorithms. Additionally, it will discuss the strengths and weaknesses of each, explain why we chose to use the one that we did, and explore possible future improvements to the locomotion system once the new walk is implemented.

II. BACKGROUND

Almost all walking algorithms have some sort of simplifying assumptions involved. Trying to control all the joints directly would be far too complicated and so we need some sort of abstraction or template [6] system that we understand and can control. For simple walking, the model most often used is the Linear Inverted Pendulum (LIPM), pictured in Figure 1.

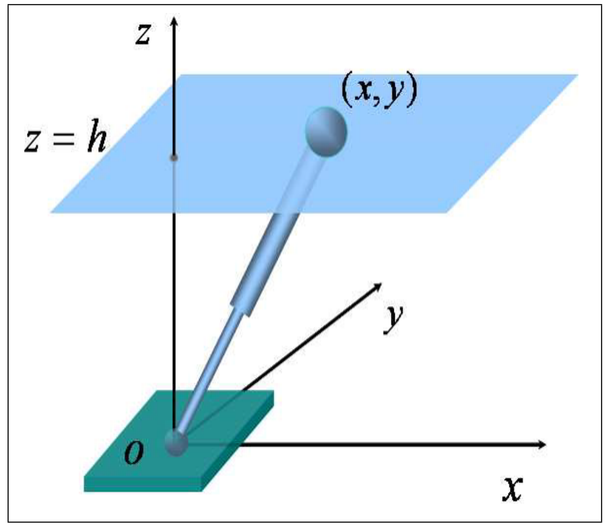

Fig. 1. Linear Inverted Pendulum diagram

This model assumes that all of the mass is concentrated at a point at the end of the pendulum with a fixed height. The fixed height assumption is not completely accurate, but this simplification allows us to use easily understandable controllers and then use feedback to compensate for the modeling errors.

In addition to a simplifying model, we would like to be able to talk about dynamic stability in locomotion. Since walking (and an inverted pendulum) are inherently unstable systems and we want to guarantee stability while in motion the notion of static stability does not work for us. Instead, we can use a concept called a Zero Moment Point (ZMP) [1].

The ZMP is a point on the ground plane where all the forces and moments, both applied and dynamic (gravity, inertia, ankle torque, etc.), cancel out, resulting in zero moment, as the name implies. If this point lies within the support polygon of the robot, then the robot will be dynamically stable. The support polygon is the convex hull of the robot's footprint on the table or, in other words, the minimal 'bounding box' that contains all points in contact with the ground plane. With this background out of the way, we can now talk about two bipedal locomotion approaches.

III. ZMP WALK WITH PREVIEW CONTROL

The first of these approaches was proposed by Kajita [2] in 2003 and adapted for the Nao by various groups in the following decade [3] [4]. Each group has a slightly different take on it but the basic idea is the same: for a prescribed set of footsteps (essentially constraints on the ZMP), how do we control the center of mass (CoM) to maintain stability?

If we take the equations of motion for the LIPM, where z_c is the fixed height of the CoM and τ is the applied ankle torque,

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x \tag{1}$$

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{m z_c} \tau_y \tag{2}$$

and then use the equations for the ZMP

$$p_x = -\frac{\tau_y}{mg} \tag{3}$$

$$p_y = \frac{\tau_x}{mg} \tag{4}$$

we can solve for the ZMP as a function of the position and acceleration of the CoM:

$$p_y = y - \frac{z_c}{g} \ddot{y} \tag{5}$$

$$p_x = x - \frac{z_c}{g} \ddot{x}. \tag{6}$$

However, we really need the inverse of this - the motion of the CoM as a function of the ZMP placements. Kajita poses this as a servo control problem and shows that in fact, as seen in Figure 2, the ideal system must begin moving the CoM before the input ZMP motion has occurred.

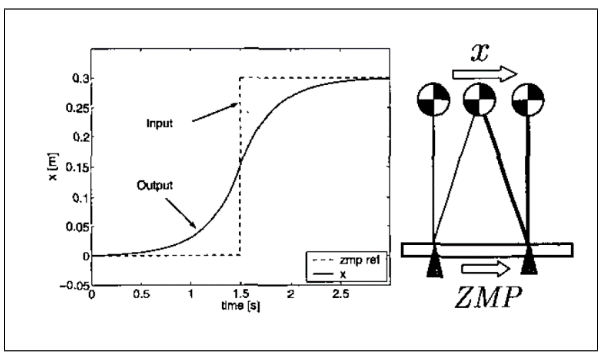

Fig. 2. Ideal system which much begin moving before input is received

Therefore, he presents the method of preview control which looks a certain number of time steps into the future to take into account the future inputs to the system. After solving the problem with an optimization of a performance index he obtains the following input equation:

$$u(k) = -G_i \sum_{i=0}^{k} e(k) - G_x x(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j). \tag{7}$$

This input consists of an integral term, a linear term, and a previewing term which takes into account N_L future time steps. Using this input does a very good job of keeping the system stable, as seen in Figures 3 and 4.

Another group [4] shows that the same system can be solved using a quadratic program as shown by Weiber [8] and they achieve very similar stability results.

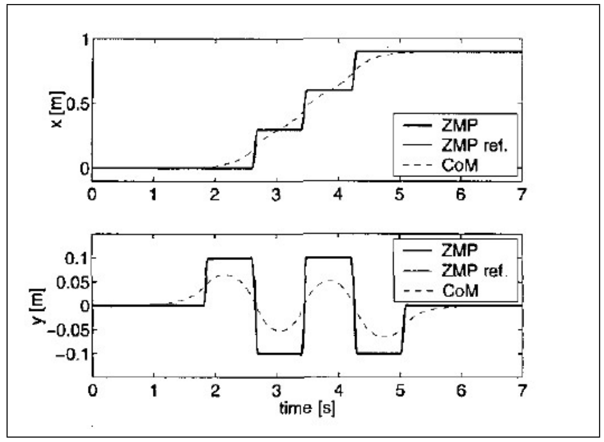

Fig. 3. System stability using preview control with adequate previewing period

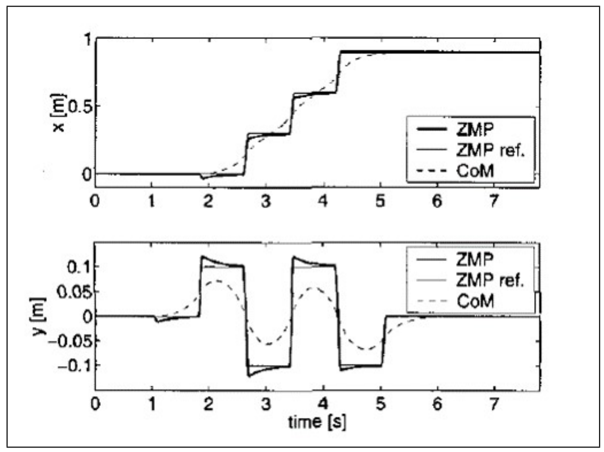

Fig. 4. Slight system instability can be seen when the previewing period is shortened

IV. BHUMAN WALK

One of the leading teams in the league, BHuman, regularly publishes papers detailing their work and so we decided to take a look at their walk since it is fairly recent recent and specifically designed for RoboCup [5].

BHuman also uses the LIPM to model their robot, but they use kinematic equations instead of kinetic equations as Kajita does. This means that each single support phase of the robot can be expressed by

$$x(t) = x_0 * cosh(kt) + \dot{x}_0 * \frac{1}{k} * sinh(kt) \tag{8}$$

$$\dot{x}(t) = x_0 * k * sinh(kt) + \dot{x}_0 * cosh(kt) \tag{9}$$

where $k = \sqrt{g/h}$. To fully parameterize these equations we must solve for x_0, $\dot{x}_0$, and t and we would like to chose values for these that maintain stability of the CoM.

Instead of prescribing footsteps and shifting the CoM to maintain stability, BHuman takes the opposite approach. They prescribe a CoM motion and make adjustments to the steps to remain stable. In order to do this, they have two main parameters to adjust - the phase timing and the location of the origin of the inverted pendulum.

The phase timing can be solved by looking at the gait in the y (side to side) direction only. If we define $t = 0$ at the inflection point where $\dot{y} = 0$, then the beginning time of each phase will be $t_b < 0$ and the ending time of each phase will be $t_e > 0$. Using bar notation to denote the next single support phase, we can write the following equalities for the y direction:

$$(x(t_e))_y - (\bar{x}(\bar{t}_b))_y = \bar{r}_y + \bar{s}_y - r_y \tag{10}$$

$$(\dot{x}(t_e))_y = (\dot{\bar{x}}(\bar{t}_b))_y \tag{11}$$

where $\bar{r}_y + \bar{s}_y - r_y$ is the distance between pendulum origins as shown in Figure 5. These equations are ensuring that both the position and velocity of the CoM are equal when the robot transitions which leg is supporting the body.

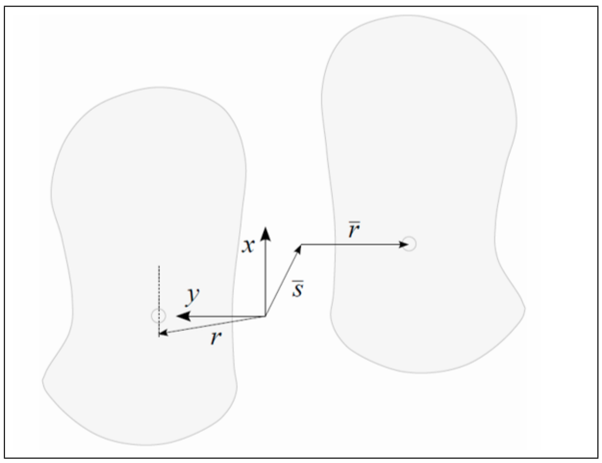

Fig. 5.

This system can be solved using an iterative approach by guessing t_e and then solving for $\bar{t_b}$:

$$\bar{t_b} = \frac{1}{\bar{k}} * arcsin\left(\frac{(x_0)_y * k * sinh(kt_e)}{(\bar{x}_0)_y * \bar{k}}\right) \tag{12}$$

Repeating this in conjunction with equations (10) and (11) will allow us to find t_e and t_b.

In the x (front to back) direction, we can adjust the origin of the LIPM. Equations that are similar to (10) and (11) can be derived in the x direction and everything can be combined into a linear system of equations

$$r + x_0 cosh(kt) + \dot{x}_0 \frac{sinh(kt)}{k} = x_t \tag{13}$$

$$x_0 k sinh(kt) + \dot{x}_0 cosh(kt) = \dot{x}_t \tag{14}$$

$$x_0 k sinh(kt_e) + \dot{x}_0 cosh(kt_e) - \bar{x}_0 \bar{k} sinh(\bar{k}\bar{t}_b) - \dot{\bar{x}}_0 cosh(\bar{k}\bar{t}_b) = 0 \tag{15}$$

$$r + x_0 cosh(kt_e) + \dot{x}_0 \frac{sinh(kt_e)}{k} - \bar{x}_0 cosh(\bar{k}\bar{t}_b) - \dot{\bar{x}}_0 \frac{sinh(\bar{k}\bar{t}_b)}{\bar{k}} - \bar{s} = \bar{r} \tag{16}$$

These can be solved to give a complete set of pendulum parameters (r, x_0, $\dot{x}_0$) for a pre-defined step size, $\bar{s}$. Additionally, if the r that is computed is out of bounds of an optimal region, we can instead adjust the step size to bring r back into the optimal region.

Since we are not using the ZMP here, these LIPM equations do not guarantee stability. To maintain stability, we instead need to observe the CoM and adjust the equations to keep the motion stable. To do the observation we can use a simple 4D Kalman filter to estimate the position and velocity of the CoM in each direction. For the update step of the Kalman filter, we can use forward kinematics and the joint angle measurements to directly calculate the position of the CoM.

Once we have a filtered estimate of the position and velocity of the CoM, we can use all of the same equations that were just defined to adjust the pendulum equations. This adjustment will use the observed position and velocity to recalculate r, x_0, $\dot{x}_0$, and t. Adjusting the equations in this way will cause the entire gait to absorb any error incurred by changing the step timing and pendulum origin location slightly. This will be much more stable than simply trying to force the CoM back to a pre-defined trajectory because the exact trajectory and footstep placement is not crucial for this walking motion.

BHuman presents their results when the robot is walking in place and pushed from the side. In Figure 6 it can be seen that both the phase and amplitude of the gait changes to adapt to this disturbance which is exactly what we want.

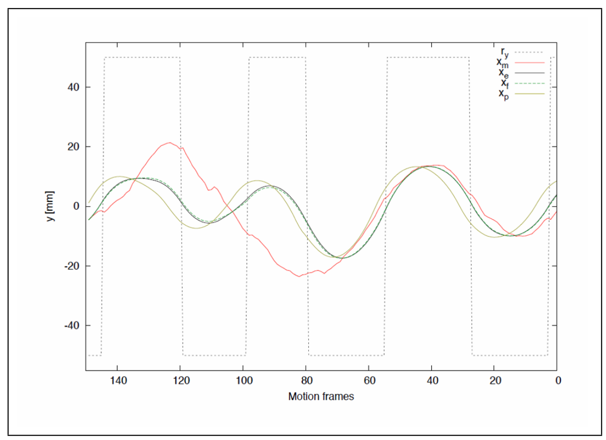

Fig. 6.

V. COMPARISON AND SELECTION

Both approaches presented can clearly achieve a stable gait and would work on the Nao robot. However, there are a few differences between them.

First, and probably most importantly, there is a clear difference in walking speed. Kajita's paper was not specific to the Nao but [3] implemented Kajita's work on the NAO and achieved speeds of 7-10 cm/s. By contrast, BHuman's walk can reach speeds of 30 cm/s. Since many leading teams in the league have walks at about 20-30 cm/s, this is definitely a major consideration.

Another difference is the implementation complexity. While it is not completely crucial, it would be ideal to be able to have a system which is as simple as possible to implement and understand. Those of us who have been working on this project will leave in a few years and we would like to avoid the problem of the new team members having to scrap and rewrite the code (again). For that reason we would like the code to be simple, modular, and easy to document so that future students can understand how the system works and build upon it. While the major complexity differences were not discussed in detail this paper, reading each of the original papers will reveal that BHuman's approach is simpler and easier to implement.

Due to both the faster walking speed and easier implementation, we have decided to pursue a system that is designed very similarly to BHuman's. We believe that this system will work well and will give our team the best starting platform to develop the walk further in the future.

VI. CONCLUSIONS AND FUTURE WORK

This paper has detailed two approaches to bipedal locomotion that could be used on the Aldebaran NAO for RoboCup and explained why the BHuman approach was selected for implementation on our robots.

At the time of writing, the RoboCup team is working to finish up the last stages of the implementation and hoping to begin testing soon to verify the stability of the walk and the integrity of the system. In the future, we hope to expand upon the work done by BHuman and add other feedback to the system to try to increase the stability even further.

Other teams have used the pressure switches in the Nao's foot [7] to provide feedback on the phase timing every half cycle which might be a useful feature for our new system. Additionally, since our team has recently migrated to the Nao V5 from the V4, the gyroscope and IMU sensors in the robot's chest are much better. We would like to experiment with adding in feedback from these sensors to help improve the position and velocity estimation of the COM that is currently only being handled by forward kinematics of the joint sensors.

REFERENCES

[1] Dekker, M. H. P. "Zero-moment point method for stable biped walking."Eindhoven University of Technology(2009).

[2] Kajita, Shuuji, et al. "Biped walking pattern generation by using preview control of zero-moment point." Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. Vol. 2. IEEE, 2003.

[3] Strom, Johannes, George Slavov, and Eric Chown. "Omnidirectional walking using zmp and preview control for the nao humanoid robot."RoboCup 2009: robot soccer world cup XIII. Springer Berlin Heidelberg, 2009. 378-389.

[4] Gouaillier, David, Cyrille Collette, and Chris Kilner. "Omnidirectional closed-loop walk for NAO."Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on. IEEE, 2010.

[5] Graf, Colin, and Thomas Rofer. "A center of mass observing 3D-LIPM gait for the RoboCup Standard Platform League humanoid."RoboCup 2011: Robot Soccer World Cup XV. Springer Berlin Heidelberg, 2011. 102-113.

[6] Full, Robert J., and Daniel E. Koditschek. "Templates and anchors: neuromechanical hypotheses of legged locomotion on land." Journal of Experimental Biology 202.23 (1999): 3325-3332.

[7] Hengst, Bernhard. "rUNSWift Walk2014 Report Robocup Standard Platform League." (2014).

[8] P.B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbation," IEEE - International Conference on Humanoids, pp. 137-142, 2006.