2016 IEEE-RAS 16th International Conference on
Humanoid Robots (Humanoids)
Cancun, Mexico, Nov 15-17, 2016

Dynamic Behaviors on the NAO Robot With Closed-Loop Whole Body Operational Space Control

Donghyun Kim, Steven Jens Jorgensen, Peter Stone, and Luis Sentis

Abstract— **Exploiting full-body dynamics in feedback control can enhance the balancing capability of a legged system using various techniques such as Whole-Body Control (WBC) or Centroidal Momentum control. However, motion control of the NAO robot based on full-body dynamics has not been extensively studied due to its limited computation power, limited sensors, and restricted access to its low-level controllers.**

Whole-Body Operational Space Control (WBOSC) is a promising WBC approach for NAO, since its closed form solution provides computational efficiency. But, users need to provide the velocity map (Jacobian) between operational space and configuration space to add the balancing control task. Thus, in this paper, we formulate the Jacobians incorporating the Capture Point (CP) technique [1] and the Centroidal Angular Momentum (CAM) [2], [3], and demonstrate the enhancement of balancing capability in a physics-based simulation.

While WBOSC reduces the computational load, implementing WBC in the real system with limited sensing capability and built-in joint position control is challenging. We show that the combination of a virtual model as an interface to the real robot and an Extended Kalman-filter based orientation estimator results in a stable implementation of a closed-loop WBOSC. We demonstrate the validity of our approach by performing a dynamic kicking motion on the physical NAO robot.

Overall, the contributions of this paper are: (1) to extend WBOSC by adding CAM and CP control tasks, and (2) to implement WBOSC in a restricted physical system by utilizing a virtual model and an orientation estimator.

I. Introduction

The capabilities of Whole-Body Controllers (WBC), which exploit full body dynamic models to accomplish dynamic tasks, are increasingly used in humanoid robots [4]. More recently, prominent demonstrations of WBC were shown by teams such as MIT [5] and IHMC [6] during the DARPA Robotics Challenge (DRC). While impressive, the ATLAS robot was limited to fairly slow whole body motions. Understandably, testing fast moving behaviors in large humanoid robots has the risk of damaging the expensive equipment.

In the Standard Platform League of RoboCup, NAOs, which are small commercial humanoid robots, play soccer while making dynamic motions such as passing, kicking, and even physical charging. However, to the best of our knowledge, there have not been previous demonstrations of closed-loop whole-body dynamic control on NAO robots. The current state of the art uses inverse kinematics [7], [8] or joint-space control [9], [10].

There are various reasons why there have been a lack of studies of WBC on low-end commercial robots such as the NAO. First, most WBCs rely on computationally expensive optimization techniques [11], [12]. Since the NAO robot has limited CPU power, implementing such an algorithm on a feedback-based real-time control loop becomes difficult. Second, dynamic motion control normally requires high-quality hardware and system requirements. For instance, the performance of IMUs (Inertial Measurement Unit) used in most DRC humanoid robots is superior to the NAO robot. Especially, the IMU of the NAO we used does not provide an angular velocity in yaw direction. (Although we note that the latest version of NAO equips the IMU with full sensing capability.) Its lack of torque sensing is another serious drawback. Third, some commercial robots like the NAO do not allow access to the embedded controllers. NAO has a built-in joint position control as the only option to control its joints. The closed loop bandwidth performance of whole-body controllers greatly depends on tuning the embedded controllers [13] and this is not possible to do in the NAO.

To handle some of these limitations, our approach extends and implements a version of the Whole-Body Operational Space Control (WBOSC) framework [14]. Since WBOSC has a closed-form solution, it is well suited for controlling a robot with low computational power. To add the task in the closed form solution, users need to find the Jacobian between the task space and the configuration space. To enhance the balancing capability, we formulate Capture Point (CP) and Centroidal Angular Momentum (CAM) based control into the WBOSC framework. We show in a physics-based simulation that while CP balances the robot, balancing capability is enhanced with the incorporation of the CAM task.

Although we circumvent the low-computation power problem by utilizing WBOSC, overcoming limited sensing capability is another challenging issue. Considering the computed command of WBOSC is torque, the lack of torque sensor in the NAO and its built-in joint position controller are implementation bottlenecks. To bypass the limitation, we put a virtual robot model between the actual robot and WBOSC. By using whole-body dynamics, the desired torque command from WBOSC are converted into desired joint accelerations, and through integration, the desired joint state position and velocity can be obtained. We also focus on creating a clean velocity signal to achieve feedback control stability. Then, the computed joint position is sent to NAO's built-in joint position controllers. By doing so, we synchronize the virtual robot model's joint and the actual robot's joint position.

To synchronize the remaining body orientation configuration, we designed an Extended Kalman-filter orientation estimator and used it for closed-loop balance control. The estimator also uses the foot contact sensor to adjust the uncertainty covariance.

978-1-5090-4718-5/16/$31.00 ©2016 IEEE

In our experiments and simulations, NAO balances on a single foot while demonstrating dynamic kicking behaviors using closed-loop WBOSC. Overall, our contributions are on incorporating new balance tasks on WBOSC, and porting methods suitable for high-end robots into low-end commercial robots.

II. Whole Body Operational Space Control (WBOSC)

WBOSC was described in [14], but a summary is provided here. A humanoid robot can be represented as a combination of its 6-dimensional floating base dynamics and its joint states. Concatenating both states into a single vector, the robot's state is represented as $q \in \mathbb{R}^{n_{dofs}}$ where n_{dofs} is the number of under-actuated and actuated degrees of freedom of the robot Since a humanoid cannot directly control its floating base dynamics, an under-actuation matrix, $U \in \mathbb{R}^{(n_{dofs}-6)\times n_{dofs}}$, is used to map the global state vector, q, to the subspace of actuated joints, q_{act}.

$$q_{act} = Uq. \tag{1}$$

Using the generalized state vector, q, the dynamics of the robot can now be represented as the following linear differential equation.

$$A\ddot{q} + b + g + J_s^T F_r = U^T \tau_{control}, \tag{2}$$

where A is the inertia matrix, b is the centrifugal and Coriolis forces, g is the gravitational forces, and $\tau_{control}$ is the output torque command on the actuated joints of the robot. When single (or dual) foot contact is considered, the contact can be described by the support Jacobian $J_s \in \mathbb{R}^3$ (or $J_s \in \mathbb{R}^6$) which maps the state vector velocity to the constrained foot (or feet) in Cartesian space. Since this is constrained, the acceleration of the foot must be 0. Substituting the constraint $J_s\ddot{q}+\dot{J}_s\dot{q} = \ddot{x}_{foot(or feet)} = 0$ and the corresponding reaction forces, F_r the dynamics become:

$$A\ddot{q} + N_s^T(b+g) + J_s^T\Lambda_s\dot{J}_s\dot{q} = (UN_s)^T\tau_{control}, \tag{3}$$

where $N_s = I - \overline{J_s}J_s$ is the null space projector of J_s under dynamically consistent inversion, $\overline{J_s} = A^{-1}J_s^T\Lambda_s$ and $\Lambda_s = (J_sA^{-1}J_s^T)^{-1}$.

An operational task, p_{task}, is defined by the constrained kinematic mapping

$$\dot{p}_{task} = J^*_{task}\dot{q}_{act}, \tag{4}$$

where $J^*_{task} = J_{task}\overline{UN}_s \in \mathbb{R}^{n_{task}\times n_{acts}}$ is the contact consistent task Jacobian and $J_{task} \in \mathbb{R}^{n_{task}\times n_{dofs}}$ is the unconstrained task Jacobian, which users need to define whenever adding a new task. Note that n_{acts} and n_{task} are the number of actuated joints and the number of dimensions controlled in the task space respectively. The torque commands can be compactly represented as

$$\tau_{control} = J^{*T}_{task}F_{task}, \tag{5}$$

where F_{task} is the operational space impedance control law,

$$F_{task} = \Lambda^*_{task}u_{task} + \mu^*_{task} + p^*_{task}, \tag{6}$$

where Λ^*_{task}, μ^*_{task}, $p^*_t ask$, and u_{task} are the operational space inertia matrix, velocity-based forces, gravity-based forces, and desired acceleration forces respectively.

III. Formulation of Centroidal Momentum Task

The Centroidal Momentum of a robot is the aggregate momentum on the robot's CoM. The robot's Centroidal Momentum vector, h_G, is related to its joint velocity vector as $h_G = A_G\dot{q}$. Alternatively, this can be expressed as the product of the robot's Centroidal inertia, I_G, and its average velocity v_G:

$$h_G = I_G v_G = A_G\dot{q}. \tag{7}$$

Using an adjoint operator to change the reference coordinate, and defining the inertia of each link as I_i, I_G and A_G can be expressed as

$$I_G = \sum_i \mathrm{Ad}^*_{T_i^{-1}} I_i \mathrm{Ad}_{T_i^{-1}}, \tag{8}$$

$$A_G = \sum_i \mathrm{Ad}^*_{T_i^{-1}} I_i J_i, \tag{9}$$

where T_i is the $SE(3)$ of the local frame of each link seen from the center of mass frame, consisting of a rotational representation (R_i) and linear position (p_i),

$$T_i = \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix}, \tag{10}$$

and Ad_{T_i} and $Ad^*_{T_i}$ are the Adjoint and dual Adjoint mapping, defined as

$$\mathrm{Ad}_{T_i} = \begin{bmatrix} R_i & 0 \\ [p_i]^\times R_i & R_i \end{bmatrix}, \tag{11}$$

$$\mathrm{Ad}^*_{T_i} = \begin{bmatrix} R_i & 0 \\ [p_i]^\times R_i & R_i \end{bmatrix}^T, \tag{12}$$

where $[p]^\times$ is the function which changes a vector to a skew symmetric matrix. From the above equations, we can easily identify the Jacobian with the following equation,

$$J_{\mathrm{CM}} = I_G^{-1}A_G. \tag{13}$$

Eq (13) represents the velocity mapping between the Centroidal Momentum (CM) space (operational space) and configuration space. We newly incorporate this mapping for creating CM tasks in the WBOSC framework.

In this paper, we only utilize the angular momentum components, which we refer to as the Centroidal Angular Momentum (CAM), by taking the last three rows of the above Jacobian matrix and specify them as a lower priority operational task. In doing so, the controller reduces the average angular velocity while maintaining a higher priority Capture Point (CP) task, which controls the linear components of the robot's CM.

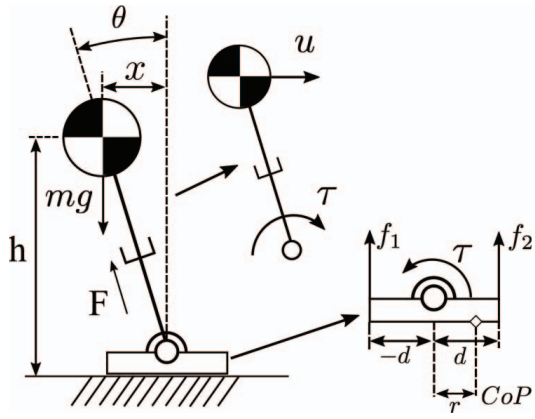


Fig. 1: **CoM and thin foot Model** A single supported legged system can be modeled as a flat foot connected with rotational joint (ankle) and prismatic joint (leg).

IV. Formulation of Capture Point Task

In [1], Capture point (CP) is defined as follows: For a biped in state x, a Capture point, p, is a point on the ground where if a biped either covers p with its stance foot or steps onto p and then maintains its Center of Pressure (CoP) on p, then there is a safe feasible trajectory that the robot will end in a Captured State. Therefore, the first objective of the balance controller is to move the CoP to the CP of a given state and then add feedback to move the CP to the desired CP location. [15] and [16] explain how to design a biped's balance controller by manipulating the CoP (or ZMP). However, it is not trivial to define the Jacobian for a CoP control task since there is no direct velocity relationship between the CoP space and the configuration space. Additionally, the CoP can only be changed by moving the CoM. Therefore, we propose the simplified model (Fig. 1) to design a CP controller in the WBOSC framework. Intuitively speaking, the following process finds the relationship between the CoM acceleration (WBOSC input) and the CoP dynamics, which will be used to manipulate the CP.

The dynamics of the robot's CoM are

$$mg = F\cos\theta, \tag{14}$$

$$m\ddot{x} = \frac{\tau}{h} + F\sin\theta. \tag{15}$$

Dividing the two equations and substituting $\tan\theta = \frac{x}{h}$ results into

$$\frac{m\ddot{x} - \frac{\tau}{h}}{mg} = \frac{x}{h}, \tag{16}$$

$$\frac{g}{h}x + \frac{\tau}{mh} = \ddot{x}. \tag{17}$$

Ignoring horizontal forces on the foot, the force/moment balance equations on the supporting foot are

$$f_1 + f_2 = mg, \tag{18}$$

$$-f_1 d + \tau + f_2 d = 0, \tag{19}$$

$$-f_1(d+r) + f_2(d-r) = 0. \tag{20}$$

From those equations, we can identify the relationship between torque and CoP position (r).

$$\frac{(f_2 - f_1)d}{f_1 + f_2} = r, \tag{21}$$

$$-\frac{\tau}{mg} = r. \tag{22}$$

Plugging Eq (22) into Eq (17) results in a linear ODE

$$\ddot{x} = \frac{g}{h}(x - r). \tag{23}$$

This equation implies that the ankle torque does not change the linear inverted pendulum dynamics above and influences only the CoP position. From the original CP formulation, $x_{\mathrm{CP}} = x + \sqrt{\frac{h}{g}}\dot{x}$. The CP dynamics are defined as

$$\dot{x}_{\mathrm{CP}} = \sqrt{\frac{g}{h}}(x_{\mathrm{CP}} - r). \tag{24}$$

Then, a suitable control law of the CoP to let the current CP converge to desired CP is

$$r = x_{\mathrm{CP}} + K(x_{\mathrm{CP}} - x^d_{\mathrm{CP}}), \tag{25}$$

where x^d_{CP} is the desired CP location and K is the gain. Inserting the above equation (25) into Eq. (23) and considering an operational task associated with the robot's center of mass, $\dot{x} = J_{\mathrm{CoM}}\dot{q}$, the desired CoM acceleration input to control the CP in WBOSC is

$$\ddot{x} = u = \frac{g}{h}(x - x_{\mathrm{CP}}) + K'(x^d_{\mathrm{CP}} - x_{\mathrm{CP}}) \tag{26}$$

$$= \frac{g}{h}(x - x - \sqrt{\frac{h}{g}}\dot{x}) + K'(x^d_{\mathrm{CP}} - x_{\mathrm{CP}}) \tag{27}$$

$$= -\sqrt{\frac{g}{h}}\dot{x} + K'(x^d_{\mathrm{CP}} - x_{\mathrm{CP}}), \tag{28}$$

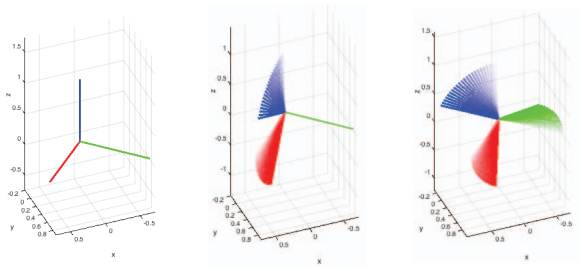
where $K' = K\frac{g}{h}$ and J_{CoM} is the Jacobian of the robot's Center of Mass. y directional capture point control is done in the same way.

V. Orientation Estimator

There have been several studies to use kinematic information to estimate the orientation of a floating body. Since the MEMS-based IMU suffers from significant drift, estimation techniques to avoid inherent noise and bias are required to control legged systems. Our approach is based on [17], [18], but a key difference is that we fix the error related to intrinsic and extrinsic rotation. In addition to the ordinary Kalman filter, we utilize the foot force sensor data to manipulate the covariance of the state transition model.

To avoid the confusion related with the quaternion convention, we clarify the definition of quaternion used in our estimator. Rotation is represented by a rotation axis (v) and with a rotation amount (θ) about the axis. In this paper, we construct a quaternion as

$$\begin{aligned} q =& [\ q_w,\ q_x,\ q_y,\ q_z\ ] \\ =& [\ \cos(\frac{\theta}{2}),\ \frac{v}{|v|}\sin(\frac{\theta}{2})\ ]. \end{aligned} \tag{29}$$



(a) q_1 (b) $q_1 \otimes q_2$ (c) $q_2 \otimes q_1$

Fig. 2: **Intrinsic and Extrinsic Rotation.** The figures shows the difference between intrinsic (b) and extrinsic (c) rotation. Here, q_1 is the 45^o rotation about z axis and q_2 is the 63^o rotation about y axis.

The rotation matrix according to the same rotation is defined by

$$R[q] = \begin{pmatrix} 1-2q_y^2-2q_z^2 & 2q_xq_y-2q_zq_w & 2q_xq_z+2q_yq_w \\ 2q_xq_y+2q_zq_z & 1-2q_x^2-2q_z^2 & 2q_yq_z-2q_xq_w \\ 2q_xq_z-2q_yq_w & 2q_yq_z+2q_xq_w & 1-2q_x^2-2q_y^2 \end{pmatrix}. \tag{30}$$

With the definition, we obtain the relationship $q \otimes Q[v] \otimes q^* = Q[R[q]v]$, where $\otimes$ is the Hamiltonian multiplication and $Q[x]$ represents the mapping from a three dimensional vector to a pure imaginary quaternion.

Due to limited computational power and poor signal quality in the accelerometer data, our estimator has a smaller number of states than [17], [18].

In previous estimator designs, there is confusion regarding intrinsic and extrinsic rotations. Additionally, further confusion comes from competing quaternion conventions. To clarify our approach, we present our convention in Fig. 2. Since the Kalman filter update process occurs in the global frame, this update must be done in an extrinsic manner. However, the IMU data from the robot should be added to the estimated orientation in an intrinsic manner since the IMU data is the angular velocity from the local frame (body orientation). Therefore, the Kalman filter prediction rules are

$$\hat{q}_{k+1}^- = \hat{q}_k^+ \otimes \exp([\hat{w}_k]^\times \Delta t), \tag{31}$$

$$\hat{z}_{k+1}^- = \hat{z}_k^+, \tag{32}$$

$$\hat{b}_{w,k+1} = \hat{b}_{w,k}^+ \tag{33}$$

and the observations are

$$s_{1,k} = \hat{q}_k^- \otimes (\hat{z}_k^-)^{-1}, \tag{34}$$

$$s_{2,k} = \hat{z}_k^-, \tag{35}$$

where $s_{1,k}$ is the orientation difference between the IMU and the stance foot, and $s_{2,k}$ is the global orientation of the stance foot.

For the orientation estimator, we use the Extended Kalman Filter, which requires linearized update matrices and they are

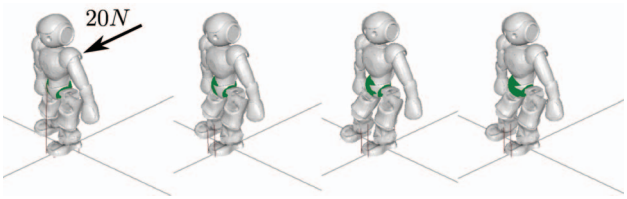


Fig. 3: **Balance Control with Capture Point Task.** NAO's balance is disturbed by a constant 20N push force for 0.1 s. By manipulating the capture point with whole-body motion, NAO recovers and continues to balance on its left foot

defined as

$$F_k = \begin{pmatrix} I & 0 & R[\hat{q}_k]\Delta t \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}, \tag{36}$$

$$H_k = \begin{pmatrix} I & -R[\hat{q}_k \otimes \hat{z}_k^{-1}] & 0 \\ 0 & I & 0 \end{pmatrix}. \tag{37}$$

Detailed derivations are presented in the Appendix.

The covariance matrix in prediction is given by the equation, $Q_k = F_k L_c Q_c L_c^T F_k^T \Delta t$, where $L_c = diag\{R[\hat{q}_k], I, I\}$, and the update matrix is $R_k = R_c/\Delta t$.

Note that the orientation observation in Eq (35) is assumed to be constant since the stance foot in the global frame is expected to be flat on the ground. If more than one of the four force sensors' signal in the foot is zero, this implies that the foot is not completely flat on the ground. To handle this scenario, the estimator adjusts the uncertainty covariances for the z term in both the prediction (Q_c) and update (R_c) process to be very high. This results to the Kalman-filter relying more on the IMU data over the kinematics.

VI. Dynamic Simulation

In the simulation, we assume that the NAO robot is a fully torque controllable humanoid. This assumption provides an environment to test highly dynamic motions and verify that our task extensions to WBOSC function as intended. Namely, the CP task balances the robot and the addition of CAM task further enhances the balancing capability.

A. Balance Control Test

To test the balance controller in the simulation, we induce a disturbance by pushing the NAO robot on its torso with a constant force for 0.1 seconds while NAO balances on its left foot.

First, we test a CP-based balance controller which only uses the CP criteria. This test is performed to verify the proposed control law in Eq (28). As seen in Fig. 3, we push the torso with $20N$ of force for 0.1 seconds. The simulation showed that the NAO robot successfully maintains its balance on its left foot.

Second, we test how much the CAM task helps with maintaining balance. In this test, we add an additional $15N$ of constant force perpendicular to the first force. As before, these two forces are induced for 0.1 seconds. Fig. 4 shows

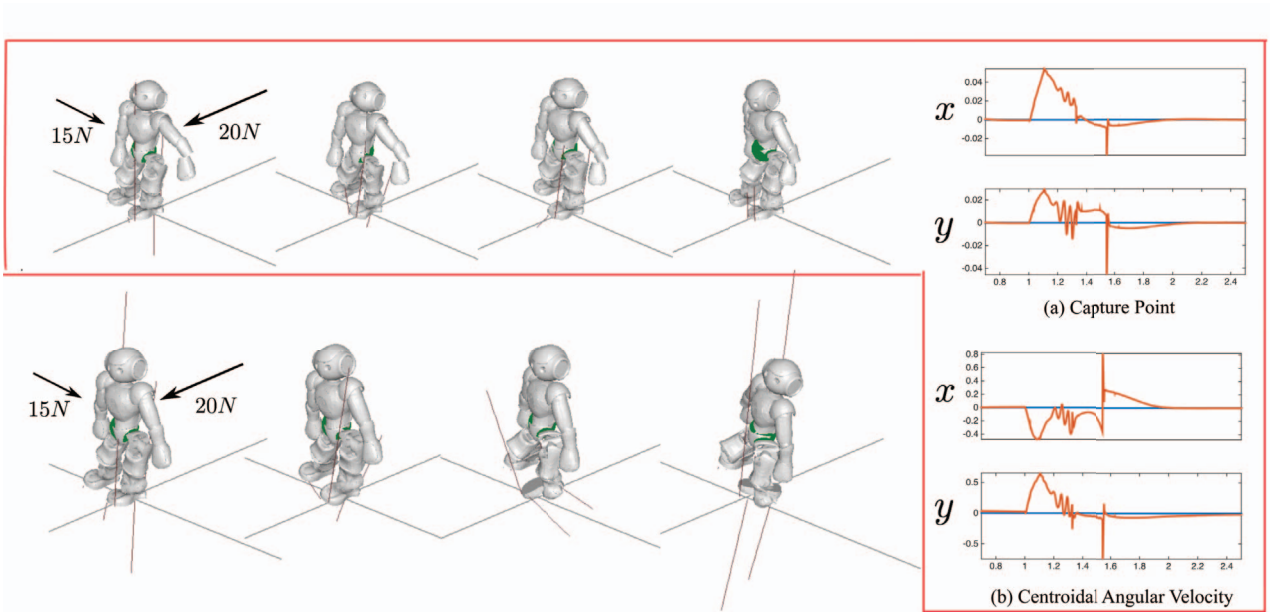


Fig. 4: **Comparison of Balance Controllers.** In these two trials, we push the NAO with 15N and 20N on the lateral and sagittal direction respectively for 0.1 seconds. The first row is the combined CP + CAM balance controller and the second row only uses a CP-based balance controller. The images boxed in red shows the tracking performance of the actual (orange line) CP and CAM trajectory to the desired (blue line) trajectories for the CP + CAM balance control test. The simulation shows that under these forces, the CP + CAM balance controller successfully balances the robot, but fails with only a CP-based controller.

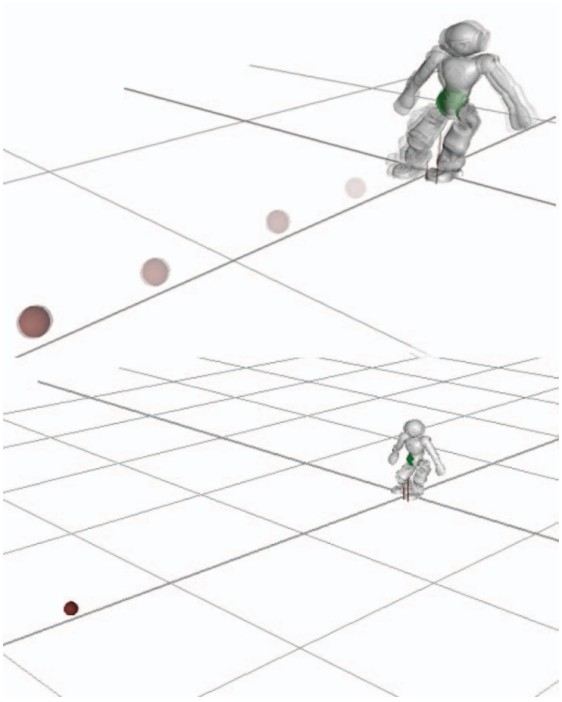


Fig. 5: **Kick Motion.** NAO kicks the 70g weighted ball. The ball travels around $2.5m$ in the simulation.

two tests being performed: (1) with CP + CAM Control, and (2) with CP Control only. Note that when performing the combined CP + CAM balance controller, we place the CP criteria as a higher priority task than the CAM task. This is because the CAM task acts as a supplement, which attempts to minimize the centroidal angular velocity of the robot.

As Fig. 4 shows, when the combined CP + CAM Controller is used, the NAO robot uses its limbs more actively and successfully balances. Otherwise, using only a CP-based controller under these forces fail. Thus, the inclusion of CAM control contributes significantly to balance control.

B. Dynamic Kick

In performing a dynamic kicking motion, we use the complete balance controller including both CP and CAM tasks. Initially the robot starts from a double support posture. To begin the kicking motion, the CP is first moved to the left foot using WBOSC and the constraint setup is also changed from dual to single contact. The NAO performs a wind up motion using foot pose control and makes a dynamic kick by swinging the right foot forward by $27cm$ in $0.1sec$. The resulting motion is presented in Fig 5.

VII. Hardware Experiments

A. WBOSC Challenges in the NAO

Implementation of WBOSC on the NAO robot is challenging since it is not designed for real-time feedback control. There are four major difficulties. The NAO platform has (1) low computational power, (2) low-quality sensing capability,

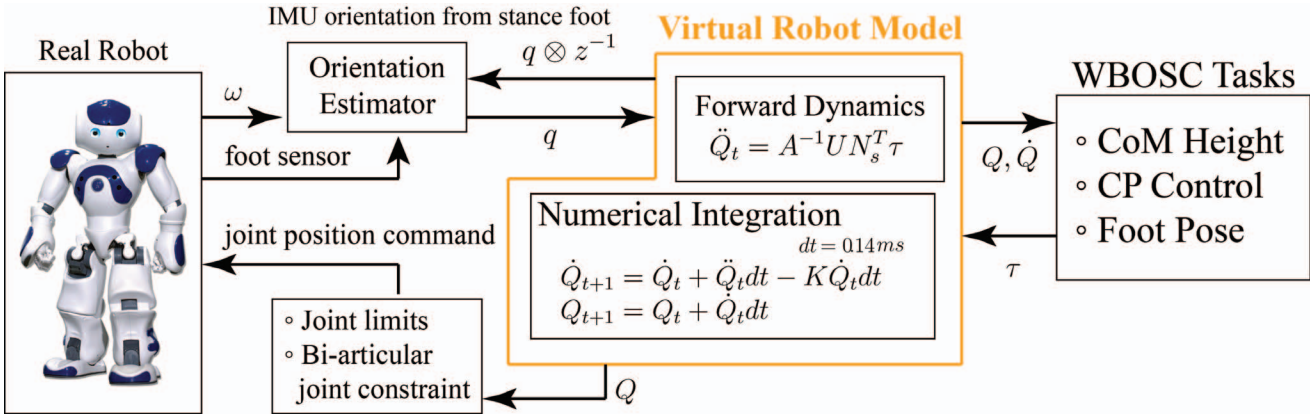


Fig. 6: **Total Control Scheme.** The virtual robot model plays an important role in the feedback control. In the figure, ω is the angular velocity data from the IMU, q is the orientation of the IMU, z is the orientation of a stance foot, Q is the full state of the dynamic model including virtual joints, and τ is the torque command from the WBOSC. In the virtual robot model, the forward dynamics convert the torque command to joint accelerations, and through integrations using dt as the time step, Q is delivered to the real robot after checking hardware constraints. Note that dt is $\frac{1}{7}$ms since it runs 70 times in one system control loop from the Real robot to the WBOSC.

(3) limited embedded controller access, and (4) particular hardware deficiencies.

First the CPU in the NAO robot is a dual core Intel Atom(TM) @ 1.60Ghz. This is a low-end computer compared to other humanoid robots that utilize computationally expensive whole-body controllers. The low computational power limits the types of estimation techniques that can be utilized. For example high dimensional Kalman filters are not possible.

Second, NAO's low sensing capability gives low quality velocity data. It is delayed, biased, and noisy. Moreover, the slow control servo rate (10ms) significantly reduce the phase margin and limit the bandwidth of the feedback controller. Furthermore, the quality of the velocity data is critical with stabilizing a feedback system since velocity feedback can be used to damp the system dynamics and provide passivity [19].

Third, in a cascaded control structure, having the ability to manipulate the joint-level controller is crucial to tuning the WBOSC feedback controller. Since NAO only provides a joint position controller that accepts only two parameters (desired joint position and stiffness), this tuning flexibility is no longer available. Additionally other types of low-level controllers such as a joint-level torque controller are not possible. It is important to remember that WBOSC returns joint torque outputs to accomplish the specified tasks. Thus, we must convert joint torque commands to appropriate joint position commands.

Finally, while it is possible to imitate a torque controller by reverse-engineering the joint-position controllers and identifying the appropriate inverse function to create a mapping between desired torque outputs and joint position commands, this is not possible due to the limited information on the embedded controllers and NAO's hardware deficiencies. NAO's spur gears in the drive train are not appropriate for torque control due to inherent friction, stiction, and backlash.

B. Implementing WBOSC on the NAO

To address the above hardware limitations, we provide the following methodologies to lower computational burden, obtain clean velocity data, and send the appropriate joint position commands from the WBOSC torque commands.

To lower computational burden, we first note that using our WBOSC framework is computationally faster due to its projection-based methods and closed-form solution. Secondly, on the NAO system itself, while the WBOSC is computed on the main control loop, the mass matrix is computed on a different thread and is only occasionally updated. Note that while the configuration-dependent mass matrix is slightly behind temporally to the true mass matrix, our empirical tests have shown that this difference is small and negligible even with a 10ms system control loop.

To obtain clean velocity data and send appropriate joint position commands, we create a virtual robot model which simulates the joint accelerations the NAO robot will experience if the links exert the desired torque commands from the WBOSC. The virtual robot model takes the joint accelerations and through integration, gives the joint velocities and positions, which are sent to the joint position controllers and the orientations are sent to the orientation estimator. Clean velocity data is obtained from the joint velocity data in the virtual robot model and is used to stabilize the system dynamics.

C. WBOSC Control Scheme for the NAO Robot

The virtual robot model consist of forward dynamics and numerical integration (Fig. 6). Note that this virtual model is what the WBOSC controls. The virtual robot model is synchronized with the actual robot via two methods. First commanding the desired joint positions of the virtual robot model to the the built-in joint position controllers synchronize the joint states. Second, the orientation estimator estimates the real robot's body orientation by combining the

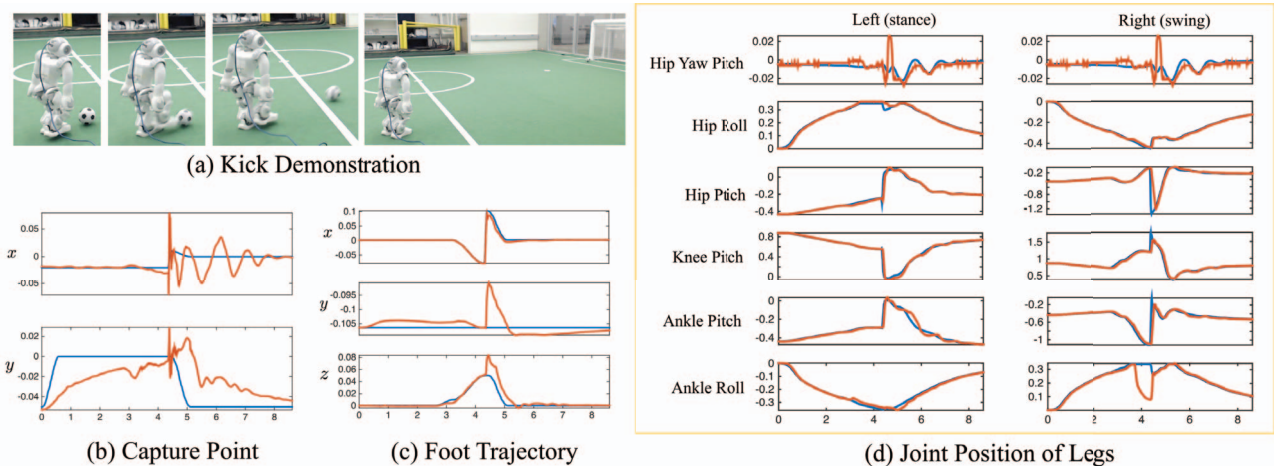


Fig. 7: **Experiment Result.** x, y, and z mean saggittal, lateral, and vertical direction, respectively. In (a) the NAO performs a dynamic kick from the half-way line of the field in which the ball traveled 5m away. In (b) and (c), the blue lines are the desired CP and foot trajectories and the orange lines are the measured trajectories. In (d), the joint position data shows that the real robot's configuration (orange) is well matched with the virtual model configuration (blue)

measured angular velocity data and its kinematics. Finally, using another Kalman-filter, we also estimate the robot's yaw velocity. In this way, WBOSC performs closed-loop control with the actual robot.

The numerical integration process requires small step sizes to avoid truncation errors. With a 10 ms system control loop rate, the numeric integrator is not stable on its own. Since this process is not computationally expensive, we run the integration step 70 times in one system control loop to further reduce the step size. By doing so, we enhance the accuracy of numeric integration.

To enhance the computational efficiency, we do not include gravity and Coriolis terms in both the WBOSC and forward dynamics. Specifically, these terms are canceled out when performing inverse dynamics in the WBOSC and forward dynamics in the virtual robot model. We also remove the neck joints and elbow joints in the virtual model since they are not actively used in the kicking demonstration. Instead, the joint position commands for these joints are fixed and WBOSC sees them as fixed masses.

D. Dynamic Kick Demonstration

To test the proposed controller's performance, we demonstrate a dynamic kicking behavior in the NAO robot. The WBOSC tasks involved in the kicking motion, in order of priority from high to low, are to: (i) maintain the CoM height, (ii) perform CP control, and (iii) do pose control on the swing foot. The desired CoM height is set to the starting height of the NAO robot after it performs its boot up routine, and the CP and foot trajectories are designed with a sinusoidal function. The ball used is the 2016 black and white competition ball used in SPL, which is 45g.

The result is presented in Fig. 7. The ball traveled 5m away. Empirical tests show that a similar performance is obtained in multiple trials. We note that due to the slow update rate of the NAO robot (10 ms), we cannot use high stiffness gains in the balance controller. The lack of high-stiffness control causes the lateral direction of the capture point to have noticeable error. However, the controller still manages to follow the desired trajectories. Overall, Fig. 7 shows that we have successful performed a dynamic motion using approach described in Sections VII-B and VII-C.

VIII. Conclusion and Future Work

In this paper, we formulate Centroidal Moment Control and Capture Point Control in WBOSC framework. We verify that we can control CAM and CP tasks in simulation by performing disturbance rejection tests, and we also demonstrate that we can perform a dynamic kicking motion both in simulation and hardware.

To utilize WBOSC for performing dynamic behaviors in a low-end commercial robot such as the NAO, we provide the following methodology. We first create an Extended Kalman-filter based orientation estimator which includes the robot's kinematic models, foot sensor data, and IMU data. Second, to be compatible with NAO's embedded joint-position controllers, we create a virtual robot model that acts as an interface between the WBOSC and the real robot.

In the future, we are gearing toward fast locomotion with the techniques presented in this paper. Fast walking is one of the most desirable capability in soccer competition, and we expect that we can achieve much higher speed locomotion provided large stable region thanks to the current study.

IX. Acknowledgements

The work was partially supported by an Office of the Naval Research (ONR) and a NASA Space Technology Research Fellowship (NSTRF) under the grant number NNX15AQ42H.

Appendix

In this section, we describe how the matrices in the orientation estimator are derived. First we derive the prediction

process for the IMU orientation.

$$q = \delta q \otimes \bar{q} \tag{38}$$

$$b_\omega = \bar{b}_\omega + \delta b_\omega \tag{39}$$

where q is the actual orientation of IMU, $\bar{q}$ is the estimated value, δq is the orientation adjustment, and similarly, b_ω is the bias, $\bar{b}_\omega$ is the estimated value, and δb_ω is the bias adjustment. Since the adjustment is done in the global coordinate system, we apply an explicit rotation.

$$\dot{q} = q \otimes \frac{1}{2}\begin{bmatrix}\omega + w_\omega + b_\omega \\ 0\end{bmatrix} \tag{40}$$

$$= q \otimes \frac{1}{2}\begin{bmatrix}\tilde{\omega} + b_\omega \\ 0\end{bmatrix} \tag{41}$$

$$\dot{\bar{q}} = \bar{q} \otimes \frac{1}{2}\begin{bmatrix}\omega + b_\omega \\ 0,\end{bmatrix} \tag{42}$$

where ω is the angular velocity measurement, and w_ω is the angular velocity noise. Next,

$$\dot{q} = \frac{d}{dt}(\delta q \otimes \bar{q}) \tag{43}$$

$$q \otimes \frac{1}{2}\begin{bmatrix}\tilde{\omega} + b_\omega \\ 0\end{bmatrix} = \dot{\delta q} \otimes \bar{q} + \delta q \otimes \dot{\bar{q}} \otimes \frac{1}{2}\begin{bmatrix}\omega + \bar{b}_\omega \\ 0\end{bmatrix} \tag{44}$$

$$q \otimes \frac{1}{2}\begin{bmatrix}\tilde{\omega} + b_\omega \\ 0\end{bmatrix} \otimes \bar{q}^{-1} = \dot{\delta q} + \delta q \otimes \dot{\bar{q}} \otimes \frac{1}{2}\begin{bmatrix}\omega + \bar{b}_\omega \\ 0\end{bmatrix} \otimes \bar{q}^{-1} \tag{45}$$

Since $q \otimes \begin{bmatrix}v \\ 0\end{bmatrix} \otimes q^{-1} = R[q]v$, we simplify the equation to

$$\dot{\delta q} = -\delta q \otimes \left(\frac{1}{2}R[\bar{q}](\omega + \bar{b}_\omega)\right) + q \otimes \bar{q}^{-1} \otimes \left(\frac{1}{2}R[\bar{q}](\tilde{\omega} + b_\omega)\right) \tag{46}$$

$$= -\delta q \otimes \left(\frac{1}{2}R[\bar{q}](\omega + \bar{b}_\omega)\right) + \delta q \otimes \left(\frac{1}{2}R[\bar{q}](\tilde{\omega} + b_\omega)\right) \tag{47}$$

$$= \delta q \otimes \frac{1}{2}R[\bar{q}](\delta b_\omega + w_\omega) \tag{48}$$

$$= \frac{1}{2}\begin{bmatrix}[R[\bar{q}](\delta b_\omega + w_\omega)]^\times & R[\bar{q}](\delta b_\omega + w_\omega) \\ -R[\bar{q}](\delta b_\omega + w_\omega) & 0\end{bmatrix}\begin{bmatrix}\frac{1}{2}\delta\phi \\ 1\end{bmatrix}, \tag{49}$$

where $\delta q = [\frac{1}{2}\delta\phi, \quad 1]^T$. From Eq (48) to Eq (49) we apply the matrix transformation from Hamiltonian multiplication. Since the multiplication of small numbers is small enough to ignore,

$$\dot{\delta q} = R[\bar{q}](\delta b_\omega + w_\omega). \tag{50}$$

Eq (50) is substituted into the F_c matrix in the linearized system written as $\dot{\delta x} = F_c \delta x + L_c w$, where $\delta x = [\delta q, \delta z, \delta b]^T$ Therefore,

$$F_c = \begin{bmatrix}0 & 0 & R[\bar{q}] \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix}, \tag{51}$$

and $F_k = e^{F_c \Delta t}$. As for the orientation difference between the IMU and the stance foot, it is same as the process described in [18].

REFERENCES

[1] J. E. Pratt and R. Tedrake, "Velocity-Based Stability Margins for Fast Bipedal Walking," in *Fast Motions in Biomechanics and Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 299–324.

[2] S.-H. Lee and A. Goswami, "Reaction Mass Pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots." *ICRA*, pp. 4667–4672, 2007.

[3] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[4] F. L. Moro, M. Gienger, A. Goswami, N. G. Tsagarakis, and D. G. Caldwell, "An attractor-based whole-body motion control (wbmc) system for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 42–49.

[5] S. Kuindersma, R. Deits, M. Fallon, and A. Valenzuela, "Optimization-based locomotion planning, estimation, and control design for the ATLAS humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2015.

[6] S. Bertrand, J. Pratt, *et al.*, "Momentum-based control framework: Application to the humanoid robots atlas and valkyrie," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Worshop Slides*, 2014.

[7] J. J. Alcaraz-Jiménez, D. H. Pérez, and H. M. Barberá, "Robust feedback control of ZMP-based gait for the humanoid robot Nao." *I. J. Robotic Res. ()*, vol. 32, no. 9-10, pp. 1074–1088, 2013.

[8] J. J. Alcaraz-Jiménez and M. Missura, "Lateral disturbance rejection for the nao robot," *RoboCup 2012: Robot ...*, vol. 7500, no. Chapter 1, pp. 1–12, 2013.

[9] I. Becht, M. de Jonge, and R. Pronk, "A Dynamic Kick for the Nao Robot," *Project Report*, 2012.

[10] A. D. Ames, E. A. Cousineau, and M. J. Powell, "Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics," in *the 15th ACM international conference*. New York, New York, USA: ACM Press, 2012, p. 135.

[11] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation." *I. J. Robotic Res. ()*, vol. 33, no. 7, pp. 1006–1028, 2014.

[12] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, "Summary of Team IHMC's virtual robotics challenge entry," *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2013)*, pp. 307–314, 2013.

[13] Y. Zhao, N. Paine, K. S. Kim, and L. Sentis, "Stability and performance limits of latency-prone distributed feedback controllers," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 11, pp. 7151–7162, 2015, In Press.

[14] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Stanford, 2007.

[15] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*. IEEE, Sept. 2011, pp. 4420–4427.

[16] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator," in *Robotics and Automation (ICRA)*. Kobe: IEEE, 2009, pp. 1966–1971.

[17] M. Blosch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU." *Robotics Science and Systems 2012*, 2012.

[18] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. IEEE, 2014, pp. 952–958.

[19] C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "On the Passivity-Based Impedance Control of Flexible Joint Robots," *Robotics, IEEE Transactions on*, vol. 24, no. 2, pp. 416–429, 2008.