Angular Center of Mass for Humanoid Robots

Yu-Ming Chen\textsuperscript{1,2}, Gabriel Nelson\textsuperscript{2,*}, Robert Griffin\textsuperscript{3}, Michael Posa\textsuperscript{1} and Jerry Pratt\textsuperscript{2,3}

Abstract—The center of mass (CoM) has been widely used in planning and control for humanoid locomotion, because it carries key information about the position of a robot. In contrast, an “angular center of mass” (ACoM), which provides an “average” orientation of a robot, is less well-known in the community, although the concept has been in the literature for about a decade. In this paper, we introduce the ACoM from a CoM perspective. We optimize for an ACoM on the humanoid robot Nadia, and demonstrate its application in walking with natural upper body motion on hardware.

Index Terms—Humanoid and Bipedal Locomotion, Natural Machine Motion, Whole-Body Motion Planning and Control

I. INTRODUCTION

Reduced-order models have been widely used in model-based planning and control of legged robots, because they capture the bulk of the robot’s dynamics despite being low-dimensional [1]–[7]. The models are often derived from physical intuition [1] or computed via optimization [7]. An example of a reduced-order model is the center of mass (CoM) along with its simple equations of motion. The CoM responds to the total net forces acting on the robot according to Newton’s 2nd Law, and gives us an overall location for the robot, which is essential for locomotion. But in order to perform more diverse tasks such as back-flips or aggressive turning, we must also consider the orientation of the robot.

To regulate the orientation, many people have been using the centroidal momentum model [8]–[11] and the single rigid body model [12]. The centroidal momentum model computes the dynamics of the momentum without predicting individual joint motion. Thus, it is a model at the velocity level and does not tell us the absolute orientation at any given time. On the other hand, the single rigid body model requires using a single SO(3) coordinate to represent the entire robot’s orientation, which raises a question – how do we pick this coordinate? For robots with a heavy torso and light limbs such as the MIT Mini Cheetah [13], one can pick the torso coordinates as a proxy for the single rigid body model. This, however, does not work well for robots with relatively heavy or long limbs, where mass is distributed throughout the rest of the system, such as IHMC Nadia [14] and Agility Robotics Cassie [15].

Unlike the center of mass, one cannot simply average each link’s orientation to get the “angular CoM”. To get the most representative angular coordinates, some have used angular excursion, which is an integral of the angular velocity about the CoM derived from centroidal momentum [16]. The downside of this approach is that it is path dependent. That is, the angular excursion, unlike the Center of Mass, can take on multiple values for the same robot state.

Researchers in the field of geometric mechanics have, for about a decade, devised a way to find the optimal coordinates “globally” instead of along a particular trajectory [17], [18]. Travers et al. [18] called these optimal coordinates the minimum perturbation coordinates. However, no literature has yet shown its application to a complex robot (except for Boston Dynamics’ patent [19]), and the minimum perturbation coordinates is less known in the community than angular momentum, partially due to the use of geometric mechanics terminology and the lack of clear motivation for the optimization objective [18]. In this paper, we will call the minimum perturbation coordinates the “angular CoM” (ACoM), since it is this property that we are looking for conceptually. We will introduce the ACoM with clear definitions of frames and vectors, and will use a simple example to motivate the objective of the problem. We then find the ACoM for the humanoid robot Nadia [14] and show an application of ACoM in walking with natural upper body motions (arm swing and spine rotation).

A. Contributions of this paper

1) introducing the ACoM with clear definitions of frames and notations, and motivating the problem objective from a CoM perspective.
2) improving the ACoM problem formulation in [18] by not assuming small angle approximations, and providing an algorithm that solves the problem quickly by exploiting its structure.
3) applying the ACoM to humanoid walking to induce natural upper body motions, including demonstration on the physical Nadia robot.

B. Related work and comparisons

Studies have shown that human walking on flat ground exhibits relatively low centroidal angular momentum (CAM) [21]. Hence, researchers have been minimizing the CAM to generate arm swing motion on a robot while assigning locomotion tasks to the legs. Erez and Todorov [22] showed that arm swing emerged from the optimal solution of an inverse dynamic controller, as a way to counteract the angular momentum generated by leg cycling. Miyata et al. [23] used relative angular momentum to balance the momentum contribution between the pelvis and the arms, when regulating the CAM to zero.

Although the CAM approach provides a natural way of generating upper body motions, it is a feedback controller based on mutually constrained velocities rather than positions, so the orientation of the robot could gradually drift away from the neutral target. To fix this issue, one has to add a competing control objective against the CAM control objective, so that the robot is servoed back to its desired orientation. In contrast, in our approach, a control law based on the ACoM provides a desired orientation for the robot, and does not necessarily need to have competing objectives. Additionally, a position measurement is a smoother signal than a velocity measurement, so input commands coming from the feedback control based on the ACoM will also theoretically be smoother than the one based on CAM. Finally, the ACoM tells us the orientation of the whole system, so we can servo this orientation instead of a user-chosen “privileged” base link, such as the pelvis. The benefit of using an average orientation (such as an ACoM) becomes more significant in fast or agile motions, particularly for robots with relatively heavy or long limbs [16].

II. OPTIMAL COORDINATES

A. Definition and motivation

Fig. 2 shows a comparison between the translation and rotation of a system. One can derive the CoM position by integration of the CoM velocity and some simple algebraic operations. However, this is not possible in the rotational case, because the conservation of angular momentum is a non-holonomic constraint [20]. This means that one cannot simply average the orientation of each body of the robot to get the “angular CoM” (ACoM) of the system, because it would not satisfy the momentum conservation law. Nevertheless, we would still like to find the most representative orientation of the entire system because it provides useful information for analysis and control.

Let \( q \) be the joint positions of the robot (excluding the floating base joint). Fig. 3 shows the frames we are interested in. Frame \( C \) is the ACoM frame that we want to find. \( W R_B \) is the rotation of the floating base relative to the world, which we already know by estimating the state with an IMU. \( B R_C \) is the rotation of the ACoM frame relative to the base frame, so it is a function of just the local joint positions \( q \). We note that \( x_{BC} \) is a function of the local joints \( q_1 \) and \( q_2 \).

Our goal is to find \( Q_{BC}(q) \) such that frame \( C \) behaves...
Eq. (3) tells us how joint motion affects base rotational motion. In the rotational case, the angular velocity of frame $C$ relative to the world is $0$ for all $q$ and $\dot{q}$.

B. Problem formulation

For readability, we will drop the subscript in $Q_{BC}$ such that $Q \triangleq Q_{BC}$, and let $Q_s$, $Q_x$, $Q_y$ and $Q_z$ be the elements of the quaternion $Q$. We will also use a prescript to denote the frame that a vector is expressed in. For example, $\omega_{WC}$ is the angular velocity of frame $C$ relative to the world frame $W$, expressed in frame $B$.

We can express $\omega_{WC}$ in terms of the angular velocity of the base relative to the world and the angular velocity of the ACoM frame relative to the base

$$\omega_{WC} = B\omega_{WB} + B\omega_{BC} = B\omega_{WB} + B R_C C \omega_{BC}. \quad (1)$$

Using conservation of centroidal angular momentum where there are no external moments on the system, we can derive the relationship between the base velocity $B\omega_{WB}$ and the local joint velocity $\dot{q}$. Here, we will assume the angular momentum is a constant zero. This is not a bad assumption for straight-line walking [21], and it is also what some researchers have been using as a control target to achieve arm swing and spine yaw rotation [22], [23]. Let $H_{CoM}$ be the centroidal angular momentum of the whole system. Thus we have

$$H_{CoM} = M_B B\omega_{WB} + M_q \dot{q} = M_B [B\omega_{WB} + B^{-1} M_q \dot{q}], \quad (2)$$

where $M_B$ and $M_q$ are parts of the centroidal momentum matrix [9], $B\omega_{WB}$ is the angular velocity of the base relative to the world expressed in the base frame, and $\dot{q}$ are the joint velocities. From Eq. (2) and the assumption $H_{CoM} = 0$, we derive

$$B\omega_{WB} = -M^{-1}_B M_q \dot{q} = A\dot{q}, \quad (3)$$

where $A \triangleq -M^{-1}_B M_q$ is a function of the local joint $q$ only. Eq. (3) tells us how joint motion affects base rotational motion.

In Eq. (1), $B R_C$ is only a function of $Q$. As for $C\omega_{BC}$, we take the time derivative of $Q(q)$ and map the derivative to angular velocity [24]

$$C\omega_{BC} = 2 E_Q \dot{Q} = 2 E_Q J_Q q, \quad (4)$$

where $J_Q$ is defined to be the Jacobian of $Q$ with respect to $q$, and

$$E_Q = \begin{bmatrix} -Q_x & Q_s & Q_z & -Q_y \\ -Q_y & -Q_z & Q_s & Q_x \\ -Q_z & Q_y & -Q_x & Q_s \end{bmatrix}.$$

Combining Eq. (1), (3) and (4), we derive

$$B\omega_{WC} = [A + 2 B R_C E_Q J_Q] \dot{q}, \quad (5)$$

where the terms inside the square brackets are only dependent on $q$ and $Q$.

Let the boldfaced $q$ be the random variable of the joint position with a uniform distribution over the feasible configuration (i.e. within the joint limits). According to Sec. II-A, our goal is to minimize Eq. (5) for all feasible configuration $q$ and for all $\dot{q}$. This is equivalent to minimizing the expectation of the norm of the matrix inside the square brackets over $q$

$$\min_Q E_q \| A + 2 B R_C E_Q J_Q \|_F$$

s.t. $\|q\|_2 = 1$, \hspace{1cm} (AO)

where $\| \cdot \|_F$ is the Frobenius norm, and we also added the unit norm constraint of the quaternion. We note that Eq. (AO) is a constrained nonlinear optimization problem, while the problem in [18] is an unconstrained least squares problem. This is the cost we incur by not making small angle approximations in our formulation.

C. Parameterization and optimization algorithm

One can parameterize $Q$ in a user-chosen way and solve the problem (AO) with algorithms like stochastic gradient descent. In our implementation, we choose to parameterize the latter three elements of the quaternion as a linear combination of some basis functions, and then recover the first element by the unit norm constraint:

$$Q_{xyz}(q; \Theta) = \Theta \phi(q), \quad Q_s = \sqrt{1 - Q^2_x - Q^2_y - Q^2_z}, \quad (6)$$

where $\Theta$ is a coefficient matrix of dimension $3 \times n_\phi$ and $\phi$ is a vector of basis functions with dimension $n_\phi$. Note that we choose to pick only the positive square root for $Q_s$, and also that $Q_s$ is a nonlinear function in $\Theta$. Given the parameterization in Eq. (6) and the following constraint on quaternion velocity

$$Q_s \dot{Q}_s + Q_x \dot{Q}_x + Q_y \dot{Q}_y + Q_z \dot{Q}_z = 0, \quad (7)$$

the time derivatives of the quaternion becomes

$$\dot{Q} = \begin{bmatrix} -Q_s^{-1} Q_{xyz}^T \dot{Q}_{xyz} \\ \dot{Q}_{xyz} \end{bmatrix} = \begin{bmatrix} -Q_s^{-1} Q_{xyz}^T \\ I_{3 \times 3} \end{bmatrix} \Theta J_Q \dot{q}, \quad (8)$$

Algorithm 1 ACoM optimization

Input: $N$ random poses $q_i$, $i = 1,...,N$

Output: $\Theta^*$

1: $\Theta \leftarrow 0$ (initialize to constant identity rotation)

2: repeat

3: Substitute $Q(q_i; \Theta)$ into $T_Q$ in Eq. (8) for $i = 1,...,N$

4: $\Theta \leftarrow$ Solve Eq. (8) with given $T_Q$

5: until convergence

6: return $\Theta$
where $J_\phi$ is defined as the Jacobian of the basis vector $\phi$, and the horizontal line inside a matrix separates the first element from the latter three elements. In our implementation, we also added Eq. (7) as a regularization term to the cost function, and instead of optimizing the expectation we simply pre-select $N$ number of random poses $q$ to optimize across. The final expression of the problem is

$$\min_\Theta \frac{1}{N} \sum_{i=1}^{N} \left\| \tilde{A} + T_Q \Theta J_\phi \right\|_F^2$$

(8)

with

$$\tilde{A} = \begin{bmatrix} 0_{1 \times n_u} \\ A \end{bmatrix}, \quad T_Q = 2 \begin{bmatrix} 1 & 0 \\ 0 & B R_C \end{bmatrix} \tilde{E}_Q \begin{bmatrix} -Q_x^{-1} Q_T \\ I_{3 \times 3} \end{bmatrix}$$

$$\tilde{E}_Q = \begin{bmatrix} -Q_x & Q_x & Q_y & Q_z \\ -Q_x & Q_x & Q_y & Q_z \\ -Q_y & -Q_z & Q_y & -Q_x \\ -Q_y & -Q_z & Q_y & -Q_x \end{bmatrix}.$$  

In Eq. (8), $\tilde{A}$ and $J_\phi$ are functions of $q$ and not $\Theta$, while $T_Q$ is nonlinear in $\Theta$.

One can solve Eq. (8) with many nonlinear solvers. In practice, we found that our simple algorithm in Alg. 1 works. The algorithm exploits the structure of the cost function by identifying that Eq. (8) is a least squares problem if $T_Q$ is given. In each iteration, we first substitute the current solution $\Theta$ into $T_Q$ to turn (8) into a least squares problem$^5$, and then solve the problem to get a new solution $\Theta$. We repeat the above steps until $\Theta$ converges.

III. CONTROLLER

Nadia (Fig. 3) is the target humanoid robot for our controllers. It has 31 degrees of freedom (DoF) – 6DoF legs, 7DoF arms, 1DoF grippers and a 3DoF spine. Fig. 5b shows the controller with ACoM-induced upper body motions. Fig. 5a shows the baseline controller with a fixed upper body pose, which we used for comparisons with our ACoM controller in experiments.

Each controller has a planner that generates desired trajectories. These are then converted into acceleration commands by the feedback controllers shown in the diagrams. With the acceleration commands, we use an inverse dynamics whole body controller (the rightmost block in each diagram in Fig. 5) to get the desired actuator commands for the robot [10]. We can roughly separate the controller into leg and upper body parts. The leg part takes care of tracking the desired path and heading of the robot, while the rest of the controller handles the upper body motion. The interaction and coordination between these controllers strongly influence the angular momentum of the system.

$^5$The Kronecker product identity $\text{vec}(XYZ) = Z^T \otimes X \text{vec}(Y)$ is useful in vectorizing the matrix $\Theta$ in preparation for solving the least squares.

[Diagram of controllers]

A. Legs

This part of the controller is the same between the baseline controller and the ACoM controller. We use Capture Point (CP) control for the locomotion task [26]–[29]. The footsteps are generated given the desired velocity commands from a higher level controller, and we augment the Linear Inverted Pendulum model with feedforward centroidal angular momentum (details in [29]). The planner outputs a reference Centroidal Moment Pivot trajectory that gets converted into a linear momentum rate command in the feedback controller, which is further sent to the inverse dynamics controller.

B. Upper body

Our short-term goal was getting more natural arm swing and spine yaw rotation by servoing just ACoM yaw$^5$. More complex motions have been left to future work. As such, we made temporary compromises and simplifications to only tackle certain parts of the whole-body control problem with the ACoM.

The ACoM can free up the pelvis to achieve high-level goals such as natural walking with natural pelvis motion and

$^5$In our experiments, we found that arm swing and spine yaw rotation were mostly induced by servoing the ACoM yaw angle to zero. Additionally, Miyata et al. [23] only used the yaw part of angular momentum to generate the arm swing.
stepping up/down on terrain. In Fig. 5b, we servo the ACoM frame (only the yaw axis) relative to the world frame, while both the pelvis and the upper body task reside in the null space of the ACoM task. In simulation, we have demonstrated straight-line walking with the controller in Fig. 5b. When moving to hardware, we made temporary compromises to mostly demonstrate arm swing and spine yaw rotation, so we did not take the full advantage of the ACoM approach. More specifically, on hardware, we servo the pelvis orientation relative to the world and regulate the ACoM yaw angle relative to the pelvis to 0.

We use a task hierarchy [25] in our whole body controller, shown in Fig. 5 as Tiers. In experiments, we noticed that the QP solver would trade the swing foot orientation tracking performance for the ACoM tracking performance, when the robot cannot regulate ACoM yaw to 0 with only the upper body. Therefore, in order to prevent the ACoM task from impairing the leg tasks, we set the ACoM task to a lower priority than the leg tasks.

Besides the above task objectives, we also add nominal pose tracking to handle the system’s redundancy. This task can be either in the null space of the ACoM task or at the same level as ACoM. The parameters for this nominal upper body pose controller can be used to sculpt the desired motion. For example, if we want more arm swing and less spine rotation, we can increase the cost weight on the spine joint.

C. Joint limits

The joint limit controller takes the current joint position and range of motion, and outputs the minimum and maximum joint acceleration for the whole body controller as constraints. Joint limits are used for both aesthetic and self-collision avoidance on the real robot. When regulating the ACoM yaw to 0, the robot could generate excessive arm swing motion or spine rotation, which does not look natural compared to a human walking gait. For Nadia, the legs are much heavier than the arms, and in order for the arms to cancel out the angular momentum contribution from the swing leg, the arms have to swing very quickly.

IV. EXPERIMENT AND RESULT

A. ACoM optimization and result

We optimized for an ACoM function for Nadia (Fig. 3) using Alg. 1. We randomly select 1000 pairs of poses mirrored about the sagittal plane. When sampling poses, we keep the gripper, wrist and ankle joints at neutral positions, because their contribution to the centroidal angular momentum is relatively small. Thus the configuration space for the ACoM has 19 dimensions. The basis functions are monomials in terms of these 19 joint angles, with all possible monomials up to 3rd order being used (e.g. \( q_1, q_1^2, q_1q_2, q_1^3, q_1q_2q_3, \ldots \)). This results in a total of 1539 basis functions. The optimization converged within about 2 minutes or 10 iterations, with much of the convergence taking place in the first 3 iterations. After the optimization, we dropped terms whose coefficients (in \( \Theta \)) are smaller than 1e-8.

We optimized for an ACoM function for Nadia (Fig. 3) using Alg. 1. We randomly select 1000 pairs of poses mirrored about the sagittal plane. When sampling poses, we keep the gripper, wrist and ankle joints at neutral positions, because their contribution to the centroidal angular momentum is relatively small. Thus the configuration space for the ACoM has 19 dimensions. The basis functions are monomials in terms of these 19 joint angles, with all possible monomials up to 3rd order being used (e.g. \( q_1, q_1^2, q_1q_2, q_1^3, q_1q_2q_3, \ldots \)). This results in a total of 1539 basis functions. The optimization converged within about 2 minutes or 10 iterations, with much of the convergence taking place in the first 3 iterations. After the optimization, we dropped terms whose coefficients (in \( \Theta \)) are smaller than 1e-8.

The goal of the problem in Eq. (AO) is to minimize \( B^\omega_{WC} \) in Eq. (5). We can also view it from a different perspective. From Eq. (1), (2), (3), and (AO), we can see that mathematically we are also looking for a frame \( C \) whose angular velocity relative to the base is as close to \( M_B^{-1} M_C \dot{q} \) as possible (i.e. minimizing the difference between \( B^\omega_{BC} \) in Eq. (1) and \( M_B^{-1} M_C \dot{q} \) in Eq. (2)). Miyata et al. [23] called \( M_B^{-1} M_C \dot{q} \) the relative angular velocity. Therefore, to evaluate the ACOM optimization result, we compare \( B^\omega_{BC} \) to the relative angular velocity, shown in Fig. 6a. The average errors of the velocities for each axis are about [0.034, 0.035, 0.061] rad/s. Additionally, we are also interested in the difference between the real angular momentum \( H_{CoM} \) in Eq. (2) and the angular momentum approximated by ACoM

\[
\tilde{H}_{CoM} = M_B \dot{B} \omega_{WB} + B^\omega_{BC},
\]

which is shown in Fig. 6b. We can see that the \( H_{CoM} \) is relatively close to \( \tilde{H}_{CoM} \). The average errors of the CAM for each axis are about [0.74, 0.84, 0.32] kg·m²/s. This means that
when we regulate the ACoM frame to identity \( p \omega_{WC} \approx 0 \), we are also regulating the real centroidal angular momentum closely to 0.

B. Arm swing and spine yaw rotation

Although the controller for the upper body motion was designed for straight-line walking, we found that, in simulation, the robot was also able to walk forward, sideways, and turn. We also demonstrated the robot walking forward on hardware. The video clips can be found in the accompanying video for this paper.

Fig. 7 shows the centroidal angular momentum of straight-line walking for both the ACoM controller and the controller with a fixed upper body pose. We see that the angular momentum profiles look similar between the simulation and the hardware, and that the CAM is smaller when ACoM yaw is regulated to 0. We note that there were issues with Nadia’s leg actuator at the time of hardware experiment and the update rate of the control loop was not fast. These issues partially contributed to the non-smoothness in the hardware plot. Additionally, we found in simulation that the foot yaw moment with the ACoM controller is smaller than the one with the fixed upper body pose, shown in Fig. 8. This is an advantage of using the upper body to counter the angular momentum generated by the legs during walking [23], [30].

V. CONCLUSION AND FUTURE WORK

We introduced the “Angular CoM” in layman’s terms and with clear frame specifications and problem motivation, so it is more accessible to a general robotic audience. A formulation of the ACoM problem without a small angle approximation was provided, including an algorithm that solves the problem quickly by exploiting its structure. An ACoM function was optimized for the humanoid robot Nadia, and was used to induce arm swing and spine yaw rotation in a walking motion. Additionally, we showed the mathematical connection between the ACoM and centroidal angular momentum, and that the angular momentum approximated by the ACoM is close to the real angular momentum in experiments.

The ACoM enables us to servo the “average” orientation of the entire system. Thus, it can free up the base link (e.g. pelvis) to achieve high-level goals such as natural walking with natural pelvis motion and stepping up/down on terrain. In this paper, we mostly demonstrated more natural arm swing and spine rotation. Future work will utilize the ACoM to achieve more complex behaviors, such as whole-body natural walking. Additionally, the benefit of an “average” orientation becomes more significant when we want the robot to execute more agile and fast motions, especially for robots with heavy or long limbs like IHMC Nadia and Agility Robotics Cassie. We are interested in using the ACoM to stabilize the orientation of these robots in running. Finally, in our problem formulation, we assume zero centroidal angular momentum and also choose a uniform distribution over the feasible configuration. Future work will explore motions with non-zero angular momentum, and more task-specific pose distributions to improve the ACoM approximation.

VI. ACKNOWLEDGMENT

We thank Sylvain Bertrand, Brandon Shrewsbury, Evan Yu, James Foster and Stephen McCrory for software setup and instruction. We also thank Stefan Fasano and Joseph Godwin for helping with the hardware experiment. This work was supported by ARL Cooperative Agreement W911NF2120241, ONR Grant No. N00014-22-1-2593, and ONR Contract No. N00014-19-1-2023.
REFERENCES

[1] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in Proceedings. 1991 IEEE International Conference on Robotics and Automation, pp. 1405–1406, IEEE Computer Society, 1991.

[2] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, “Optimization-based Full Body Control for the DARPA Robotics Challenge,” Journal of Field Robotics, vol. 32, no. 2, pp. 293–312, 2015.

[3] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCreary, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Dura, 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,” in Proceedings of the IEEE-RAS International Conference on Humanoid Robots, (Atlanta, GA), oct 2013.

[4] S. Kundersma, R. Deits, M. Fallon, A. Vanzemuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, “Optimization-based locomotion planning, estimation, and control design for Atlas,” Autonomous Robots, vol. 10, no. 1, pp. 129–159, 2015.

[5] C. Hubicki, A. Abate, P. Clary, S. Rezazadeh, M. Jones, A. Peekema, J. Van Why, R. Domres, A. Wu, W. Martin, et al., “Walking and running with passive compliance,” IEEE Robotics & Automation Magazine, vol. 1, 2016.

[6] W. C. Martin, A. Wu, and H. Geyer, “Experimental evaluation of deadbeat running on the atrias biped,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1085–1092, 2017.

[7] Y.-M. Chen and M. Posa, “Optimal reduced-order modeling of bipedal locomotion,” in 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 8753–8760, IEEE, 2020.

[8] D. E. Orin and A. Goswami, “Centroidal momentum matrix of a humanoid robot: Structure and properties,” in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 655–659, IEEE, 2008.

[9] P. M. Wensing and D. E. Orin, “Improved computation of the humanoid centroidal dynamics and application for whole-body control,” International Journal of Humanoid Robotics, vol. 13, no. 01, p. 1550039, 2016.

[10] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, “Design of a momentum-based control framework and application to the humanoid robot Atlas,” International Journal of Humanoid Robotics, vol. 13, no. 1, 2016.

[11] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics,” IEEE-RAS International Conference on Humanoid Robots, 2014.

[12] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2245–2252, IEEE, 2018.

[13] B. Katz, J. Di Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadruped control,” in 2019 International Conference on Robotics and Automation (ICRA), pp. 6295–6301, IEEE, 2019.

[14] “Nadia humanoid.” https://www.ihmc.us/nadia-humanoid/. Accessed: 2022-09-05.

[15] R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, “Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning,” arXiv preprint arXiv:2207.04163, 2022.

[16] V. Zordan, D. Brown, A. Macchietto, and K. Yin, “Control of rotational dynamics for ground and aerial behavior,” IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 10, pp. 1356–1366, 2014.

[17] R. L. Hatton and H. Choset, “Optimizing coordinate choice for locomoting systems,” in 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 4493–4498, IEEE, 2010.

[18] M. Travers, R. Hatton, and H. Choset, “Minimum perturbation coordinates on so (3),” in 2013 American Control Conference, pp. 2006–2012, IEEE, 2013.

[19] A. Khiripin and A. A. Rizzi, “Natural pitch and roll,” Dec. 13 2016. US Patent 9,517,561.

[20] Y. Nakamura and R. Mukherjee, “Nonholonomic path planning of space robots via bi-directional approach,” in Proceedings., IEEE International Conference on Robotics and Automation (ICRA), pp. 1764–1769, IEEE, 1990.