2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

# Controlling Fast Height Variation of an Actively Articulated Wheeled Humanoid Robot Using Center of Mass Trajectory

Moyin V. Otubela and Conor McGinn

*Abstract*— **Hybrid wheel-legged robots have begun to demonstrate the ability to adapt to complex terrain traditionally inaccessible to purely wheeled morphologies. Further research is needed into how their dynamics can be optimally controlled for developing highly adaptive behaviours on challenging terrain. Using optimal center of mass (COM) kinematic trajectories, this work examines the nonlinear dynamics control problem for fast height adaptation on the hybrid humanoid platform known as *Aerobot*. We explore the dynamics control problem through experimentation with an offline trajectory optimisation (TO) method and a task-space inverse dynamics (TSID) controller for varying the robot's height. Our TO approach uses sequential quadratic programming (SQP) to solve optimal 7th order spline coefficients for the robot's kinematics. The nonlinear Zero Moment Point (ZMP) is used to model a stability criterion that is constrained in the TO problem to ensure dynamic stability. Our TSID controller follows motion plans based on using task jacobians and a simplified passive dynamics model of the *Aerobot* platform. Results exhibit fast height adaptation on the *Aerobot* platform with significantly differing results between the control methods that prompts new research into how it may be controlled online.**

## I. INTRODUCTION

There are many applications of service robots that necessitates them operating alongside people in normal, everyday settings. Environments engineered for human occupancy are carefully designed with anthropomorphic requirements in mind, prompting the need for service robots to be slender enough to travel between doors while simultaneously having the capacity to vary their height to perform tasks such as accessing shelves and tables. The choice of kinematic morphology is among the biggest decisions that designers of service robots make. Bipedal robots, by virtue of sharing similar kinematics with humans, inherit the ability to traverse unstructured terrain via planning discrete combinations of steps. Unfortunately, these legged systems suffer from significant issues regarding their ability to maintain its stability without computationally costly methods and can be practically difficult to implement. An alternative morphology in the literature is the wheeled/tracked robot. This platform is mechanically much simpler and normally possesses higher stability and improved controllability. However, it comes with the significant penalty of having a lower terrain adaptability. New research is investigating novel hybrid robot morphologies that bring together the best of both biped and wheeled systems into a single platform. The potential for these systems to perform promisingly was evident in the DARPA Robotics Challenge where the leading team in a disaster response challenge used the hybrid platform, *DRC-HUBO+*[1], that made intelligent use of its wheeled and legged modes to optimally execute various tasks in the competition. Limited literature exists that explores the dynamics control problem of hybrid systems for performing the level of terrain adaptability normally reserved for biped robots [2].

*Aerobot* is a hybrid wheel-leg humanoid robot that has recently demonstrated impressive locomotive capabilities under real-world situations including step climbing and train boarding [3]. A new controller was recently developed to automate this process using trajectory optimisation methods that generated dynamically stable whole body motion plans [4]. The primary limitation of this method was its offline control architecture, due to the high computational cost of solving the required nonlinear optimisation problems. As a result, the robot could not react to abrupt changes in its environmental conditions. A key motivation of this paper is the development of a new motion planner that overcomes this limitation.

This work aims to build upon concepts that encourage fast and robust control for *Aerobot*. We start this endeavour by demonstrating fast and highly stable dynamic height adjustment, a task that to the best of our knowledge has not been demonstrated on a wheeled system that resembles closely a humanoid morphology. The theory behind our approach is discussed in section III followed with a discussion on how a model of the systems dynamics is built in section IV. Section V exhibits the trajectory optimisation and task-space inverse dynamics methods. Then sections VI, VII and VIII present the results, discussion and conclusions, respectively.

## II. PRIOR WORK

Trajectory optimisation (TO) is a control method that consists of generating temporal sequences of control inputs for adapting a robot's posture to its surroundings and internal states [2], [5]. Typically, the dynamics of these systems are modelled to develop a stability criteria of which is subsequently constrained in an optimisation problem to prevent falling and maintain postural balance. In [6], a trajectory optimization method was shown to control various robot morphologies through modelling the full passive rigid body dynamics of these systems as they interacted with complex terrain via friction and inelastic impacts without a pre-defined contact schedule. *DURUS* [7], a humanoid robot developed at SRI utilizes highly efficient electro-mechanical actuators, active springs at its ankles, and a motion planner that made

*This research was funded by the Irish Research Council.

Moyin V. Otubela and Conor McGinn are with the School of Engineering, Trinity College Dublin, Dublin, Ireland otubelam@tcd.ie, mcginnc@tcd.ie

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

use of the full dynamics of the robot enabling it to achieve highly efficient walking trajectories.

The prior art referenced in the first paragraph has showcased locomotion and traversal of complex, indeterminate terrain using full robot dynamic models, which have relied on inherently computationally intensive methods which greatly slows down the control frequency. It is common to simplify the full dynamics of robotic systems in order to reduce the computational complexity of models. This is normally achieved by relating the model to the behaviour of a robot's centre of mass (COM) which can be used for inferring the stability of a robot [8], [9], [10].

A popular model for understanding a robot's dynamic stability is the nonlinear ZMP. This model uses a robot's COM position, acceleration and rate of change of angular momentum to determine if, at a given instant, a system will tip over [9]. By assuming a constant vertical height of a robot's COM this model becomes linear, and is referred to as the Linear Inverted Pendulum Model (LIPM), which can be seen used popularly by biped robots [11], [12], [13]. Thanks to its linear formulation, Quadratic Programming (QP) can be used to constrain feasible COM motion plans at greatly reduced solver times [14]. However, due to assuming a constant COM height, this model can not predict a robot's stability while performing tasks that include variation of its height. This problem has been addressed by adding additional constraints in the optimisation problem with reference to the LIPM for determining stability. In [15] these constraints came in the form of inequalities that bounded COM accelerations based on the extreme limits of a biped robot's COM heights. This is a valid approach by virtue of a robot's center of pressure residing on a line segment between a valid range of COM heights of a robot but results in overly restricted movement.

The contributions this paper offers the state-of-the-art are threefold. Firstly, with reference to the nonlinear ZMP model, we derive objectives that permit the use of QP solvers for fast height variation and locomotion. Secondly, we develop two controllers for performing fast height variation on the *Aerobot* platform; one with TO and the other using TSID. Using the theory postulated in this paper in section III, we modify the control problem to be solved in order to enhance the robustness of the emergent behaviour of a simulated model of *Aerobot*. For the TS inverse dynamics problem in particular, the complexity of the reference COM trajectory to be tracked is shown to be greatly simplified thanks to our approach. And thirdly, we offer a study based on the results of the controllers' performances. Due to the mechanical design and purpose of the robot, an important performance feature we aim for in the robot is in its ability to maintain an upright torso as it transitions between its extreme sitting and standing body configurations. This is a complex issue by virtue of the mass-distribution and morphology of the robot which have been shown to put the system at risk of tipping over due to the platform's torso having a considerable proportion of the robot's total mass [3], [4]. Our work further investigates *Aerobot*'s dynamics problem through experimenting, contrasting and comparing with two control methods on the novel platform.

## III. Theory

The ZMP model referred to in this paper assumes negligible momentum and is based on the equation provided in [16]. Mathematically, it is denoted in equation 1.

$$x_{zmp} = c_x - \frac{c_z \ddot{c}_x}{g + \ddot{c}_z} \tag{1}$$

where: $c_x$ and $c_z$ are the horizontal and vertical location for a robot's COM, where, $\ddot{c}_x$ and $\ddot{c}_z$ are their accelerations respectively. $g$ is the acceleration due to gravity. As $\ddot{c}_x \rightarrow 0$ it is shown in equations 2a, 2b and 2c that the location of the robot's ZMP is coincident with its COM, no matter the height of the robot's COM or its vertical acceleration as long as it does not cause the robot to lift-off. This suggests that a minimal horizontal acceleration is ideal for minimising the rapid fluctuating of a stability criterion based on the ZMP model. This formulation greatly simplifies and linearises the dynamics problem of which a QP solver can be used to plan optimal COM motion for height adjustment on a robot when assuming that its horizontal movement is negligible.

$$x_{zmp} = c_x - \frac{0}{g + \ddot{c}_z}, \quad \ddot{c}_x \rightarrow 0 \tag{2a}$$

$$x_{zmp} = c_x - \frac{0}{g + \infty}, \quad \ddot{c}_x \rightarrow 0, \quad \ddot{c}_z \rightarrow \infty \tag{2b}$$

$$x_{zmp} = c_x \tag{2c}$$

When $\ddot{c}_z \rightarrow 0$ the LIPM appears as shown in equation 3a. Assuming no vertical acceleration of the robot's COM, the height of the COM can be assumed to be constant. Since acceleration due to gravity is constant, the fraction $\frac{c_z}{g}$ can be represented as a constant number, $K_z$, in equation 3b and 3c. This magnitude of this number is dependent solely on the height of the robot's COM. Consequently, this quality amplifies the fluctuations of a robot's horizontal ZMP. This suggests lower COM heights are more stable than their corresponding higher counterparts. In addition, unlike in equations 2b and 2c, careful attention to the robot's acceleration must be paid to ensure stability as exhibited in equations 3c and 3d.

$$x_{zmp} = c_x - \frac{c_z \ddot{c}_x}{g}, \quad \ddot{c}_z \rightarrow 0 \tag{3a}$$

$$x_{zmp} = c_x - K_z \ddot{c}_x \tag{3b}$$

$$x_{zmp} = c_x - K_z \infty, \quad \ddot{c}_z \rightarrow 0, \quad \ddot{c}_x \rightarrow \infty \tag{3c}$$

$$x_{zmp} = -\infty \tag{3d}$$

The deviation between COM and ZMP can also be seen to be minimized if the coordinated effect of each centre of mass causes the overall centre of mass to have minimum acceleration. In addition, the contributory effect of each centre of mass' acceleration on the ZMP is evidently dependent on the weight distribution of the robot. For masses on the robot with weights tending towards zero, their acceleration has increasingly minimal effect on the robot's ZMP deviation from its centre of mass position.

From these insights, it is apparent that a motion planner may benefit by taking into account these effects to allow for fast locomotion while its ZMP resides close to its COM. This means that minimization of the respective joint coordinated accelerations with the largest effect on ZMP deviation from the COM can be optimised to improve dynamic stability. This prompts the development of a heuristic weighting strategy for a nonlinear TO method where some advantage may be reaped for escaping entrapment from local minima which might be complicated from using multiple nonlinear objectives.

In addition the theory suggests ways in which the nonlinear ZMP model can be converted into a linear one to be used by a QP solver. Hence for a task such as height adjustment, the ZMP can be assumed coincident with the initial COM horizontal location, as long as the robot keeps $\ddot{c}_x$ to a minimum and $\ddot{c}_z$ below the required amount needed to take off from the ground.

## IV. Modelling Aerobot

For modelling and controlling the behaviour of the robot, a template dynamic model was instantiated to perform inverse dynamics for the calculation of force control inputs needed to execute whole body motion plans. The free-floating base for *Aerobot* is located mid-way between its shank drive wheels, see figure 1. This component is under-actuated. To adjust its position angle in simulation, the robot's stabiliser is controlled which subsequently affects the shank's angle due to kinematic and dynamic constraints. The respective $n_j$ actuated joint angles, are then denoted as $q_j \in \mathbb{R}^{n_j}$. Following these notations, the equation of motion for *Aerobot* is denoted as:

$$M(q)\ddot{q} + h(q,\dot{q}) = S^T\tau + J(q)^T\lambda \tag{4}$$

where $M(q) \in \mathbb{R}^{n_j+6\times n_j+6}$ is the mass matrix, $h(q,\dot{q}) \in \mathbb{R}^{n_j+6}$ denoting the Coriolis, centrifugal and gravity terms, $\tau \in \mathbb{R}^{n_j+6}$ the torque vector, $J(q) \in \mathbb{R}^{k\times n_j+6}$ the Jacobian of the robot's supports, k number of contact forces $\lambda \in \mathbb{R}^k$, and $S \in \mathbb{R}^{n_j+6\times n_j+6}$, the selection matrix which is used to map under-actuation to a robot's joints.

We simplify the dynamics model of *Aerobot* by only taking into account dynamics across its saggital plane. This restricts the model to 2 dimensions. As a result, the equations of motion in our model has 6 degrees of freedom, with 2 used to denote the pitch angle of the shank's underactuated DOF and it's vertical height. The dynamic effects of the passive wheels are modelled as a simple point contact with no friction. The dynamic effects of the drive wheels are also simplified as a point contact that assumes no slipping.

Inverse dynamics is done using orthogonal decomposition on its floating base dynamics [17] is realized from these simplifications to instantiate the feed-forward terms of the inverse dynamics trajectory follower controller in our TO method. This 2D model is also leveraged in our TSID implementation.

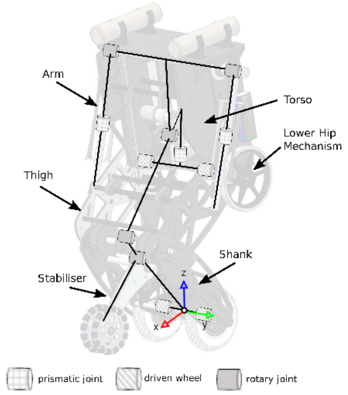


Fig. 1. Kinematics of *Aerobot* platform.

## V. Motion Optimization

A general description of the robot's motion optimization, is given in this section. Two controllers are presented. Firstly, we show our TO method which is solved via a sequential quadratic programme (SQP), available in MATLAB's optimization toolbox. Secondly a TSID controller is exhibited which is solved by quadratic programming (QP) that we execute on MATLAB's optimization toolbox.

### A. Trajectory Optimization

The TO problem solves the local minimum of the sum of weighted objectives that feature performance goals of the robot over discrete instances in time, e.g. maintaining torso angle, see equation 5. In addition, the problem is subject to non-linear equality and inequality constraints which are used to maintain stability with regard to the chosen stability criteria and the robot's physical limitations.

$$\underset{x}{\text{minimize}} \quad w^T f(x,t) \tag{5a}$$

$$\text{subject to} \quad eq(x,t) = 0, \tag{5b}$$

$$ineq(x,t) \leq 0, \tag{5c}$$

$$f_{kin}(x,t) \in \Psi, \tag{5d}$$

where:

- $f \in \mathbb{R}^p$ denotes the vector of objectives.
- $f_{kin}$ denotes the vector of joint position, velocity and acceleration, found via equations 7b, 8c and 8d.
- $w \in \mathbb{R}^p$ denotes the weight vector.
- $\Psi$ denotes the set that describes the kinematic limitations of the robot.
- $t \in \mathbb{R}^n$ is the time vector

In contrast to using Euler integration to calculate the time derivatives of the joints as in our previous work [4], a seventh order spline is used. Since the second derivative of these splines are smooth, the motion plans are constrained to have smooth accelerations, reducing the occurrence of discontinuous torques.

Using the time vector in equation 6a, spline coefficients, as presented in equation 6b, are used to calculate joint position according to equation 6c

$$\Omega(t) = \begin{bmatrix}1 & t & t^2 & t^3 & t^4 & t^5 & t^6 & t^7\end{bmatrix}^T \tag{6a}$$

$$a = \begin{bmatrix}a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7\end{bmatrix}^T \tag{6b}$$

$$q_a(t) = \Omega(t)^T a \tag{6c}$$

Given the solver variable $x = \begin{bmatrix}a^T & b^T & c^T & d^T & e^T\end{bmatrix}^T$ where the vectors b, c, d and e are coefficients unique to each of the robot's joints, the full joint kinematics $q(t) \in \mathbb{R}^{n_j+2}$ can be mathematically denoted with matrix notation as

$$q(t) = \underbrace{\begin{bmatrix}\Omega(t)^T & 0_{1\times32} & \\ 0_{1\times8} & \Omega(t)^T & 0_{1\times24} \\ 0_{1\times16} & \Omega(t)^T & 0_{1\times16} \\ 0_{1\times24} & \Omega(t)^T & 0_{1\times8} \\ & 0_{1\times32} & \Omega(t)^T\end{bmatrix}}_{=T(t)} x \tag{7a}$$

$$q(t) = T(t)x \tag{7b}$$

The velocity 8c and acceleration 8d of the joints can then be derived through using time vectors 8a and 8b.

$$\dot{\Omega}(t) = \begin{bmatrix}0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 & 6t^5 & 7t^6\end{bmatrix}^T \tag{8a}$$

$$\ddot{\Omega}(t) = \begin{bmatrix}0 & 0 & 2 & 6t & 12t^2 & 20t^3 & 30t^4 & 42t^5\end{bmatrix}^T \tag{8b}$$

$$\dot{q}(t) = \dot{T}(t)x \tag{8c}$$

$$\ddot{q}(t) = \ddot{T}(t)x \tag{8d}$$

Thus a trajectory, $q_t$, can be calculated at regular step sizes, $h$, for n waypoints

$$q_t = \begin{bmatrix}q(0) & q(h) & \dots & q((n-1)h)\end{bmatrix} \in \mathbb{R}^{5\times n} \tag{9}$$

where the same notation can be used for getting the trajectories for velocity and acceleration.

***Equality:***

The cartesian acceleration $\ddot{\mathbf{x}}_\mathbf{c} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}$ is constrained to zero to maintain consistent contact configurations over the motion plan. The initial and final joint poses, denoted by $q_0 \in \mathbb{R}^m$ and $q_n \in \mathbb{R}^m$ respectively were constrained to predefined values. The initial state is chosen to be the default/current pose of the robot. An optimal final robot pose, $q_n$ is found by running an independent SQP solver which determines a static pose that maximises the robot's stability and height adjustment. As a result, the motion optimisation seeks the optimal feasible path between two static poses, with equality constraints presented in equation 10

$$q(0) - q_0 = 0, \tag{10a}$$

$$q((n-1)h) - q_n = 0 \tag{10b}$$

***Inequality:***

Stability requirements are satisfied through the use of inequality constraints. To test the hypothesis presented in this work, we formulate a general description that produces plans based on a given stability criterion. Since each criterion is based on the existence of a measurable quantity $x_s$ within a set $S$, the support polygon. Stability for a given motion plan with respect to the chosen criteria is met when

$$x_{s_i} \in S_i \text{ for all i} = 1, 2, \dots, n \tag{11}$$

For the offline motion planner in this section, $x_s = c_x$.

***Objectives:***

Objectives are used to maximise the robot's desired behaviour while transitioning between states. Since height adjustment is guaranteed via the equality constraints, we need not seek position control of the robot's COM.

To best resemble a human form, *Aerobot's* torso angle $q_{torso} \in \mathbb{R}^n$ is desired to remain vertical, where $q_{torso} = \overrightarrow{0}$ is upright and optimal.

Given the derivations made in section III, the ideal $c$ number of COM accelerations $\ddot{x}_{G_t} \in \mathbb{R}^{n\times c}, \ddot{z}_{G_t} \in \mathbb{R}^{n\times c}$ should be minimised so that $\ddot{x}_{G_t}, \ddot{z}_{G_t} = \overrightarrow{0}$. Since some masses on the robot platform appear to contribute more to the instability of the robot, we use a weight vector, $\xi \in \mathbb{R}^c$ which penalises accelerations of a COM according to its contribution on its influence on a robot's ZMP.

$$\begin{aligned} f = q_{torso}^T w_1 q_{torso} + (\ddot{x}_{G_t}\xi_1)^T w_2 \ddot{x}_{G_t}\xi_1 + \\ (\ddot{z}_{G_t}\xi_2)^T w_3 \ddot{z}_{G_t}\xi_2 \end{aligned} \tag{12}$$

where $w_1$, $w_2$, $w_3 \in \mathbb{R}$ are weights in the multi-objective optimisation problem.

***Inverse dynamics***

The inverse dynamics controller described below, was used to follow the trajectories from the TO generator.

Firstly, the actuated and under-actuated motion dynamics of *Aerobot* are separated via calculation of the QR decomposition of its contact Jacobian as done in [17].

$$J_C^T = Q \begin{bmatrix}R \\ 0\end{bmatrix}, \tag{13}$$

where Q is an ortohogonal matrix ($QQ^T = Q^TQ = I$) and $rank(R) = k$. This leaves the below equation of motion for the actuated joints which can be used to derive the control input for following a motion plan

$$S_u Q^T(M\ddot{q} + h) = S_u Q^T S^T \tau \tag{14a}$$

$$S_u = \begin{bmatrix}\mathbf{0}_{(n_j-k)\times k} & \mathbf{I}_{(n_j-k)\times(n_j-k)}\end{bmatrix} \tag{14b}$$

$$\tau = (S_u Q^T S^T)^+ S_u Q^T [M\ddot{q}_d + h] \tag{14c}$$

$$u = \tau + \mathrm{P}(q - q_d) + \mathrm{D}(\dot{q} - \dot{q}_d) \tag{14d}$$

where $(\_)^+$ indicates the Moore-Penros psuedoinverse ($A^+ = A(AA^T)^{-1}$) and $\ddot{q}_d$ is the desired acceleration of the robot's joints.

where:

- $u$ is the control input.
- P and D are matrices representing proportional and derivative feedback control gain, respectively.

### B. TSID

Our approach aims to ensure dynamic stability during fast standing via the passive dynamic model discussed in section IV. To perform standing we calculate a formal optimal COM vertical acceleration profile for tracking in the controller. This controller is captured in the below optimisation problem which solves for the optimal state $x \in \mathbb{R}^{14\times1}$,

$$\underset{x=(\ddot{q},\tau,\lambda)}{\text{minimize}} \quad \frac{1}{2}x^T H x + f^T x \tag{15a}$$

$$\text{subject to} \quad Bx = C, \quad x \in \Gamma \tag{15b}$$

where $\Gamma$ is the set of feasible solver variables. $H \in \mathbb{R}^{14\times14}$ is the hessian, $f \in \mathbb{R}^{14\times1}$ the linear term and $B \in \mathbb{R}^{8\times14}$, $C \in \mathbb{R}^{8\times1}$ being constants to be determined by the state of the robot's dynamics that are dependent on position and velocity.

Position constraints of the robot's joints were set by constricting the acceleration of the robot's joints as the joint limits were approached over a prediction horizon $\delta t$. This constraint is implemented with reference to the quadratic formula $q_{i+1} = q_i + \dot{q}_i\delta t + \frac{1}{2}\ddot{q}_s\delta t^2$, where subscripts $i$, $i+1$ and $s$ denote the detected, predicted and solved kinematic states, respectively.

***Equality:***

The equality constraints are used to apply the full passive dynamic model equations as showcased in section IV. Given the online kinematic states of its joint positions and velocities the constants, B and C are the following

$$B = \begin{bmatrix} M(q) & -S^T & -J(q)^T \\ J_c(q) & 0_{2\times8} & \end{bmatrix} \tag{16a}$$

$$C = \begin{bmatrix} -h(q,\dot{q}) \\ -\dot{J}_c(q,\dot{q})\dot{q} \end{bmatrix} \tag{16b}$$

***Objectives:***

The regularised least squares problem $\frac{1}{2}\|Ax-b\|^2+\lambda\|x\|^2$ is used to derive our quadratic objectives. Thus, the hessian and linear term of the QP problem are the following:

$$A = \begin{bmatrix} J_T & 0_{1\times8} \end{bmatrix} \tag{17a}$$

$$b = \ddot{a}_r + K_p e + K_d\dot{e} - \dot{J}_T\dot{q} \tag{17b}$$

$$H = A^T A + \lambda I \tag{17c}$$

$$f = -A^T b \tag{17d}$$

where $J_T$ and $\dot{J}_T$ are the jacobian and time-derivative jacobian of the task space, respectively. $I$ is an identity matrix and $\lambda$ a weight. In addition the constants $K_p$ and $K_d$ are control gains which work with the same principal as PD controllers where the error state, $e$ and $\dot{e}$ are used to perform feedback control to control a process value towards a set value.

This formulation is used to track the acceleration profile of a target optimal COM motion plan for height variation.

## VI. RESULTS

### A. Experiment setup

The robot simulator, Gazebo [18], and the Robot Operating System (ROS) [19] were used to test the controllers. Motion plans were generated offline first using MATLAB's optimization toolbox where a SQP is accessed through the *fmincon* function. All computations were performed on a Intel Core i5-2500 CPU with 4 cores running at 3.3 GHz. Data was captured at 1KHz for all experiments.

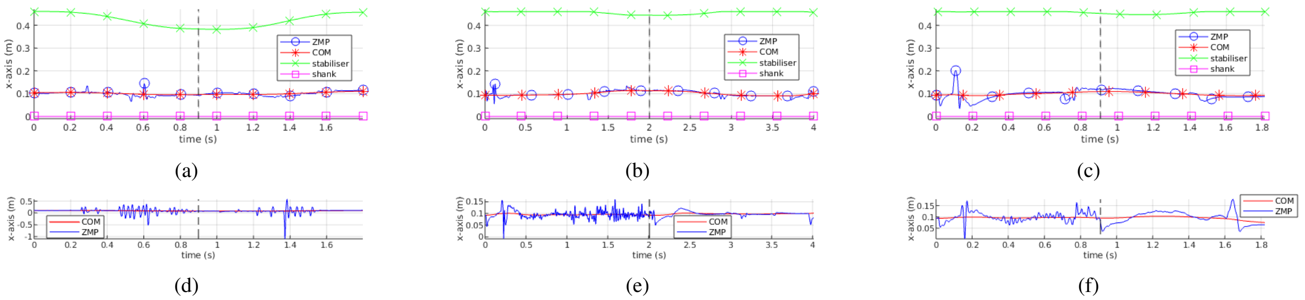
Fast trajectories were produced in the TO method, the vector, $t \in \mathbb{R}^{10}$, was predefined to have a short interval with constant step size, $h \in \mathbb{R}$ set to 100 ms. This enforced the robot to demonstrate standing and sitting with high accelerations at the robot's joints. The following control gains, as mentioned in equation 14d, were the following P = diag($\begin{bmatrix}60 & 75 & 40 & 40 & 40\end{bmatrix}$) and D = diag($\begin{bmatrix}55 & 55 & 33 & 35 & 35\end{bmatrix}$) for the stabiliser, thigh, torso, left and right shoulder joints, respectively. The mean solver time for the offline trajectories is 2940 ms. The feedforward term, $\tau$, reached update rates of up to 500Hz, while the control loop is run at 1KHz for the TO experiment.

For the TSID controller, two optimal profiles for vertical COM motion plans are computed offline. This is done to examine the behaviour of the controller under a decelerated COM motion plan and an accelerated COM motion plan. To maximise stability, the horizontal position of the robot's COM is optimised to stay in the same position in agreement with equation 2c. To balance between least-squares objectives weights were implemented to assign priority over the execution of multiple tasks, which in this paper are the horizontal and vertical COM accelerations. Tuning of the control gain values was carried out to encourage optimal performance. MATLAB's *quadprog* function was used to solve the QP problems. The control gains for this experiment as mentioned in equation 17b, were K$_p$ = diag($\begin{bmatrix}200 & 300\end{bmatrix}$) and K$_d$ = diag($\begin{bmatrix}120 & 50\end{bmatrix}$) for the horizontal and vertical COM motion, respectively. On average, the solver times were 1.57 ms and 8.95 ms for the TSID and COM solvers, respectively.

A separate experiment was conducted to examine the robustness of the controllers to disturbances that are not modelled. This was done by subjecting the robot's hip to loads of up to 300 N while the robot is standing.

### B. Height variation using TO method

Height adjustment was demonstrated successfully with this approach. The heuristic used for selecting weights added the most costs to COM accelerations of joints that accounted for most of the system's weight distribution (kg). Due to its human-inspired form, *Aerobot* has most of its mass at its torso. As a result, the largest cost was applied on the acceleration of its torso's COM. Some of the robot's



(a) (b) (c)

(d) (e) (f)

Fig. 2. Time series data showing COM and ZMP trends from the simulations. Vertical dotted lines denote where the squatting motion for *Aerobot* commences. Dynamically stable standing and sitting using (a) TO method with optimised COM acceleration, (b) TSID controller and (c) TSID controller with higher COM acceleration trajectory. (d)-(f) ZMP and COM trends when unmodelled forces were introduced for each of the corresponding experiments above. Corner frequencies $f_c$ =980Hz, $f_c$ =500Hz and $f_c$ =100Hz, were used to in a low pass filter for $q$, $\dot{q}$ and $\ddot{q}$, respectively, in all experiments.

components had a small amount of mass so the weights for them were reduced accordingly. This resulted in the ZMP trajectory presented in figure 2(a). In this case, we can see that the ZMP stays closer to the COM, promoting a more stable height variation motion plan. This behaviour is not as apparent when a disturbance is applied, see figure 2(d).

### C. Height variation using TSID

Results demonstrated successful variation of *Aerobot*'s height via tracking of an optimal COM trajectory. The platform's COM is seen to maintain good proximity with its initial position. However, there are a number of key differences in its performance when compared to the TO-based controller. The movement of the robot inherits more jerk at the beginning of the height variation process which corresponds to more discontinuous ZMP behaviour around its COM. In addition, the robot's torso struggled to maintain an upright configuration of which is evident in this paper's accompanying video. Lastly, between the two COM motion plans tested, the robot's ZMP fluctuated about its COM with a higher magnitude when the faster offline COM trajectory was utilised. This is also the case when unmodeled disturbances were introduced, see figure 2(e) and 2(f).

## VII. DISCUSSION

Dynamic stability is seen to be maintained in experiment VI-B without causing wide fluctuation of the robot's ZMP. This addresses a common problem in the literature where upon random occurrences amplified by jerky control, a robot's ZMP can find itself outside its constricted support region. In an online control scheme, the subsequent trajectory optimization problem can become infeasible to solve as a result of this condition.

The ZMP similarly remains in close proximity to the robot's COM for slower motion plans in the TS inverse dynamics controller. However it did not satisfy the condition needed for application of the linearised ZMP model due to its behaviour, as shown in figure 2(c) when tasked with tracking high acceleration COM plans. Be that as it may, due to constrictions placed on the ground reaction forces (to be at least +30N) on the full passive dynamic model leveraged, dynamically stable height variation could still be performed by virtue of its capacity to react to the robot's immediate states quickly. For instance, it can be executed in a closed loop at high control rates (≥100Hz) to solve for torque control inputs that re-balance the ground reaction forces at its wheels to ensure dynamic stability. This is evident in our video where we demonstrate good disturbance rejection capacity when *Aerobot*'s torso is subjected to large loads that are not modelled during its standing up procedure.

Unwanted behaviours were also present under the TSID control scheme that prompts future research. Given the results achieved on the trajectory optimisation method implemented, a possible remedy may be realised via the use of additional objectives for more sophisticated acceleration control which may include implementing additional regularization terms for tracking ideal joint level trajectories in the regularized least squares QP problem or via including torso orientation task objectives. Performances may also be improved for both TO and TSID controllers by accelerating their respective execution speeds.

## VIII. CONCLUSION

This research demonstrates motion planning for height variation for an actively articulated wheeled humanoid robot using theory that promotes fast motions predominantly vertical or horizontal. SQP is leveraged to solve the trajectory optimisation problem which stabilised the behaviour of the robot's ZMP via minimisation of its COM accelerations. The generated plan was then tracked using an inverse dynamics controller. The study through this approach's proves that the platform is indeed capable of transitioning between two extreme states rapidly while minimising the robot's overall COM horizontal acceleration for good agreement with the theory presented in section III. The TSID controller also demonstrates successful results in demonstrating height variation on *Aerobot*. However, it is apparent that more sophisticated planning methods are required to ensure that the compulsory objectives required for good functioning of the robot are met robustly. This is also the case if one is to capitalise on the linearised ZMP models for dynamic movement showcased in section III.

## ACKNOWLEDGMENT

This research is funded by the Irish Research Council.

# REFERENCES

[1] H. Bae, I. Lee, T. Jung, and J.-H. Oh, “Walking-wheeling dual mode strategy for humanoid robot, drc-hubo+,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1342–1348.

[2] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, “Keep rollin’-whole-body motion control and planning for wheeled quadrupedal robots,” *arXiv preprint arXiv:1809.03557*, 2018.

[3] C. McGinn, M. F. Cullinan, M. Otubela, and K. Kelly, “Design of a terrain adaptive wheeled robot for human-orientated environments,” *Autonomous Robots*, vol. 43, no. 1, pp. 63–78, 2019.

[4] M. Otubela, M. Cullinan, and C. McGinn, “Controlling aerobot: Development of a motion planner for an actively articulated wheeled humanoid robot,” in *2019 IEEE international conference on robotics and automation*. IEEE, 2019.

[5] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, “Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints,” *IEEE Robotics and Automation Letters*, 2019.

[6] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[7] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki, and A. D. Ames, “Realizing dynamic and efficient bipedal locomotion on the humanoid robot durus,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1794–1801.

[8] E. Garcia, J. Estremera, and P. G. De Santos, “A classification of stability margins for walking robots,” *Robotica*, vol. 20, no. 6, pp. 595–606, 2002.

[9] P. Sardain and G. Bessonnet, “Forces acting on a biped robot. center of pressure-zero moment point,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[10] J. Zico Kolter and A. Y. Ng, “The stanford littledog: A learning and rapid replanning approach to quadruped locomotion,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 150–174, 2011.

[11] A. Ames, J. Rogers, F. Hammond III, Y. Wardi, A. Thomaz *et al.*, “Dynamic humanoid locomotion: Hybrid zero dynamics based gait optimization via direct collocation methods,” Ph.D. dissertation, Georgia Institute of Technology, 2016.

[12] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, “Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 1028–1035.

[13] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, “Versatile and robust 3d walking with a simulated humanoid robot (atlas): A model predictive control approach,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1943–1950.

[14] K. Van Heerden, “Planning com trajectory with variable height and foot position with reactive stepping for humanoid robots,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6275–6280.

[15] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber, “A robust linear mpc approach to online generation of 3d biped walking motion,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 595–601.

[16] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, “Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5148–5154.

[17] M. Mistry, J. Buchli, and S. Schaal, “Inverse dynamics control of floating base systems using orthogonal decomposition,” in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 3406–3412.

[18] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2149–2154.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “Ros: an open-source robot operating system,” vol. 3, 01 2009.