A Quadruped Inertial Parameter Estimation Method with Bisection Search and Sinusoidal Excitations

Chi-Yen Lee[1] and Zachary Manchester[1]

***Abstract*— In this paper, we introduce a two-step calibration routine to identify the planar center of mass (CoM) position and the effective centroidal dynamics parameters of any quadruped using only joint sensors and an inertial measurement unit (IMU). Our proposed calibration routine consists of two steps: A bisection search method is used to locate the position of the planar CoM, and a sinusoidal excitation method is used to extract moments of inertia about each body axis. We verifu the inertial parameter identification method in simulation, and we implemented the center of mass finding algorithm in both simulation and hardware. The results of hardware CoM finding experiments verified in a balancing controller that requires 5mm CoM position accuracy.**

***Paper Type* Original Work**

I. INTRODUCTION

For model-based control algorithms, the accuracy of the system's model directly impacts the performance of the controller. Past and recent works on robot system identification focused primarily on identifying the full inertial properties of each individual link [1, 2]. However, many of the best-performing state-of-the-art controllers require only a simplified centroidal model of the robot [3]. The primary contribution of our work is to introduce a drastically simplified method for extracting any quadruped's centroidal inertial parameters. We introduce a simple two-step calibration routine to identify the planar center of mass (CoM) and the effective centroidal dynamics parameters using only joint sensors and an inertial measurement unit (IMU). Our proposed calibration routine consists of two steps:

- A bisection search method to locate planar CoM.
- A sinusoidal excitation method to extract moments of inertia for each body axis.

The ideas behind these routines are simple enough to be applied to nearly any quadrupedal system with a set of joint sensors and an IMU. Our algorithms require no specific controller nor complicated physical setup. We demonstrate the methods in both hardware and simulation. A video demonstrating the experiments is also available online[1].

II. BISECTION SEARCH

The main idea behind the bisection search method is to leverage the tilting direction of the robot when standing on two feet as an indicator of the center of mass error relative to a support line. When a quadruped lifts two of its feet off the ground during a stance, the support polygon formed by the four feet becomes a support line between the remaining diagonal feet, as shown in Figure 1. The robot should remain balanced for a brief moment in an unstable equilibrium point if the CoM is directly on the support vector [4]. Otherwise, the robot will tilt in the direction of the CoM offset as illustrated in Figure 2a and 2b. Using this knowledge, we shift the diagonal support line using a bisection search in the robot's body frame until the it moves below the CoM location. The details of the search method are summarized in Algorithm 1, where we take in diagonal pairs of nominal foot position vectors r_1, r_2 and r_3, r_4, an upper and lower search bound x_u and x_l, and a t_{thres} that represents the time it takes for the robot to tilt into a static pose. Line 2-5 offset the support vector by x_m, the midpoint of x_l and x_u, and use inverse kinematics to calculate a desired joint configuration q_d. Line 6-8 then move the robot into the desired static stance with a joint position controller, and Line 9-12 then lift two of the diagonal leg and let the robot lean toward the direction of the CoM offsets. Depending on the direction of tilt, as determined by the pitch error, we update x_u or x_l to narrow the search direction until the two numbers come within a certain threshold (which was set to 0.005 m in Algorithm 1). We perform the same process for both pairs of diagonal feet (front right feet and rear left feet v.s. front left feet and rear right feet), and we identify two lines that we know the planar CoM must be located on. The center of mass location can be obtained by finding the intersection of the two lines.

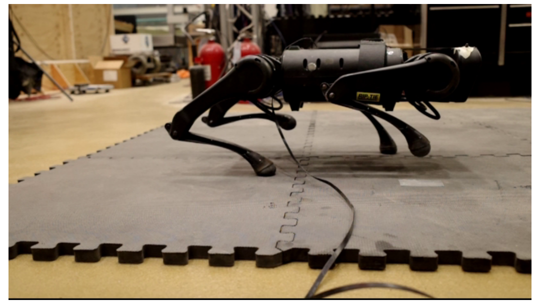


Fig. 1: An image of an Unitree A1 quadruped balancing with its CoM directly above the support vector line formed by its front left and rear right feet.

III. TRUNK INERTIAL PARAMETER ESTIMATION WITH SINUSOIDAL EXCITATIONS

Instead of trying to estimate the inertial parameters for the full-body model, we focus on fitting the inertial parameters for a simplified centroidal model. The governing equation of

Authors are with [1] the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

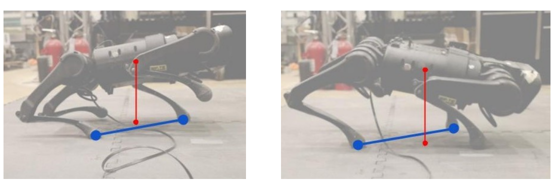
[1]https://youtu.be/oWS1gqfT1m0

Algorithm 1 FindSupportVector(r_1, r_2, r_3, r_4 x_u, x_l, t_{thres})

Require: $x_u > x_l$
while $x_u - x_l > 0.005$ **do**
  $x_m \leftarrow (x_u + x_l)/2$
  $\hat{r}_1.x \leftarrow r_1.x + x_m$
  $\hat{r}_2.x \leftarrow r_2.x + x_m$
  $q_d = \text{invKin}(\hat{r}_1, \hat{r}_2)$
  repeat
    jointPositionControl(q, q_d)
  until $q_d \approx q$
  q_n = liftDiagonalFeet(q)
  repeat
    jointPositionControl(q, q_n)
  until $t > t_{thres}$
  if $pitcherror > 0$ **then**
    $x_l = x_m$
  else
    $x_u = x_m$
  end if
end while
return $\hat{r}_1$, $\hat{r}_2$

Algorithm 2 FindCOM(r_1, r_2, r_3, r_4 x_h, x_l, t_{thres})

Initialize x_l, x_h
Initialize r_1, r_2, r_3, r_4
$\hat{r}_1, \hat{r}_2$ = FindSupportVector(r_1, r_2, r_3, r_4, x_h, x_l, t_{thres})
$\hat{r}_3, \hat{r}_4$ = FindSupportVector(r_3, r_4, r_1, r_1, x_h, x_l, t_{thres})
x_c, y_c = FindIntersection([$\hat{r}_1, \hat{r}_2$], [$\hat{r}_3, \hat{r}_4$])



(a) (b)

Fig. 2: Depending on the center of mass (red) location relative to the support vector line (blue), the robot will either tilt forward like in 2b or backward like in 2a

. .

a centroidal model quadruped can be written as,

$$\begin{bmatrix} \ddot{p} \\ \frac{d}{dt}(I\omega) \end{bmatrix} = \sum_{i=0}^{n} \begin{bmatrix} \frac{f_i}{m} \\ \hat{r_i} \times f_i \end{bmatrix} - \begin{bmatrix} g \\ 0 \end{bmatrix}, \tag{1}$$

where f_i is the ground reaction force for foot i, p is the CoM position, I is the moment of inertia, ω is the angular velocity, r_i is the foot position relative to the CoM, and g is gravity. Assuming that the angular velocity is small and the off-diagonal terms on the inertia tensor are negligible, we simplify the rotational dynamics in each axis as

$$I_j \dot{\omega}_i + C_j \omega_i = \tau_j, \tag{2}$$

where I_j, ω_j, C_j, and τ_j are the robot's moment, angular velocity, damping constant, and total torque for a rotation axis j. I_j and C_j can be identified by forming a regressor matrix with a dataset of ω_j, $\dot{\omega}_j$ and τ_j. To obtain the angular acceleration $\dot{\omega}_j$, we avoid doing numerical differentiation by providing a sinusoidal input to the system. We extract the dominant frequency $\mathcal{F}_j$, amplitude a_j, and phase shift ϕ_j via Fast Fourier Transform, and we take the derivative of the wave function analytically to get $\dot{\omega}_j$, giving us the expression

$$\omega_j(t) = a_j \sin(2\pi \mathcal{F}_j t + \phi_j) \tag{3}$$

$$\dot{\omega}_j(t) = a_j 2\pi \mathcal{F}_j \cos(2\pi \mathcal{F}_j t + \phi_j). \tag{4}$$

With an analytical expression of the angular velocity and angular acceleration, we form the regressor matrix by sampling ω_j and $\dot{\omega}_j$ at a number of discrete time steps: $0 \ldots t_f$. With the regressor matrix, we arrive at the following linear least-squares problem

$$\begin{bmatrix} \dot{\omega}_j(0) & \omega_j(0) \\ \dot{\omega}_j(t_1) & \omega_j(t_1) \\ \vdots & \vdots \\ \dot{\omega}_j(t_n) & \omega_j(t_n) \end{bmatrix} \begin{bmatrix} I_j \\ C_j \end{bmatrix} = \begin{bmatrix} \tau_j(0) \\ \tau_j(t_1) \\ \vdots \\ \tau_j(t_f) \end{bmatrix}. \tag{5}$$

We apply this sinusoidal excitation process for each of the three rotation axis, and we perform separate the least squared optimization for each axis to extract the moment of inertia and damping constant.

IV. RESULTS

We implemented the center of mass search on both hardware and simulation. As of the time of writing, we have only performed the moment of inertia identification in simulation.

A. Simulation Results

We ran both calibration routines in Gazebo with the full Unitree A1 quadruped model. Table I shows the simulated and estimated moment of inertia and CoM. To obtain the effective moment of inertia values, we use the parallel axis theorem and calculate the total moment of inertia of the robot's trunk, hips, and thigh links. Similarly, we calculate the CoM position by calculating the weighted sum of the CoM of each link in the robot's body frame. Monte Carlo simulation of the calibration routine is able to consistently identify the CoM position within 5mm of the ground truth position. The error is also directly proportional to the joint and end-effector error. Figure 3 shows the norm of the CoM position error during a bisection search, with the search error converging to under 5mm in roughly six iterations.

B. Experimental Results

We implemented the CoM bisection search algorithm on hardware for an Unitree A1 quadruped. While it is hard to experimentally determine the ground truth location of the CoM on hardware, we implemented a balancing controller that balances the robot on two diagonal feet. This under-actuated balancing scenario requires an accurate center of

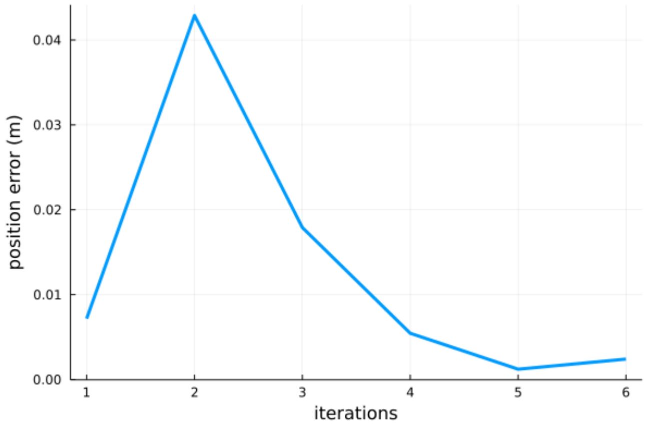


Fig. 3: Norm of the position error on the CoM estimates over six iterations for a calibration test in simulation.

TABLE I: Simulated Result of the Calibration Routine

	I_x	I_y	I_z	p_x	p_y
Simulated	0.116	0.349	0.399	-0.0093	0.00085
Estimated	0.0856	0.306	0.363	-0.0073	0.00089

mass position estimate. From our simulation and hardware experiments, we determine that the CoM needs to be within 5mm for the robot to balance properly with our controller. We were able to successfully balance the robot on two legs using the center of mass position estimate we obtained from our calibration routine. A video demonstration of the calibration process is available in Section I.

V. CONCLUSION AND FUTURE WORK

In this paper, we demonstrated our center of mass search algorithm works on both hardware and simulation. Our current calibration routine only identifies the CoM position on the x and y plane of the robot body frame. However, a similar technique can be used to back out the z-axis CoM position by tilting the robot at an angle and accounting for geometry constraints given a known $x - y$ CoM location. The sinusoidal excitation method, although it can identify the moment of inertia in simulation up to two decimal digit accuracy, have yet to be tested on hardware. Our future work will test this technique for the Unitree A1 robot and adapt it according to the challenges we encounter.

REFERENCES

[1] Christopher G. Atkeson, Chae H. An, and John M. Hollerbach. "Estimation of Inertial Parameters of Manipulator Loads and Links". In: *The International Journal of Robotics Research* 5.3 (Sept. 1986), pp. 101–119. ISSN: 0278-3649, 1741-3176.

[2] Guido Tournois et al. "Online payload identification for quadruped robots". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC: IEEE, Sept. 2017, pp. 4889–4896. ISBN: 978-1-5386-2682-5.

[3] Jared Di Carlo et al. "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid: IEEE, Oct. 2018, pp. 1–9. ISBN: 978-1-5386-8094-0.

[4] Marc H Raibert. "Research on legged machines can lead to the construction of useful legged vehicles and help us to understand legged locomotion in animals." In: *Communications of the ACM* 29.6 (1986), p. 16.