

Contact Force Distribution Using Centroidal Momentum Feedback for Quadruped Locomotion

1st Erim Can Ozcinar
Biomechatronics Laboratory
Dept. of Mechanical Engineering
Ozyegin University
34794 Istanbul, Türkiye
erim.ozcinar@ozu.edu.tr

2nd Ozkan Bebek
Robotics Laboratory
Dept. of Mechanical Engineering
Ozyegin University
34794 Istanbul, Türkiye
ozkan.bebek@ozyegin.edu.tr

3rd Barkan Ugurlu
Biomechatronics Laboratory
Dept. of Mechanical Engineering
Ozyegin University
34794 Istanbul, Türkiye
barkan.ugurlu@ozyegin.edu.tr

Abstract—This paper proposes a method to distribute contact forces to eliminate undesired torso orientation fluctuations and regulate heading during dynamic quadruped trot-walking motion. The proposed method makes use of centroidal momentum, an essential physical quantity in characterizing the whole-body behavior and overall balance of the robot. The contact forces are computed using centroidal momentum feedback and injected into the locomotion controller via virtual model control, which can render virtual forces to regulate robot-environment interaction. In combining the virtual model control and centroidal momentum feedback, one can attain dynamically feasible contact force distribution, a key objective in controlling dynamic quadruped locomotion. In order to validate the proposed method, a series of simulation experiments were conducted using a realistic model of our quadruped robot *Kara*. As a result, we obtained dynamically consistent trot-walking behavior in which torso orientation and heading were well regulated.

Index Terms—quadruped walking, locomotion control, contact force distribution.

I. INTRODUCTION

Nowadays, quadruped robots are pretty prevalent and draw much attention. There are several commercialized and product-level quadruped robots. For instance, Boston Dynamics' Spot allows operators to automate routine inspection tasks, and data capture safely [1]. Unitree Robotics commercialized A1, a versatile quadruped robot with limited payload capability [2]. Istituto Italiano di Tecnologia (IIT) developed a hydraulically powered quadruped robot named Hyq, a versatile robot with advanced locomotion properties [3]. MIT developed the quadruped robot Mini Cheetah capable of displaying agile motions, e.g., back flip [4].

Despite the fact that the literature offers impressive quadrupeds, we argue that advanced controller synthesis for quadrupeds could still be an active research area. One of the most crucial concepts for legged robots is the dynamic balance while moving. Dynamic balance of these robots can be achieved with Zero Moment Point (ZMP) based pattern generation [5]. Although ZMP-based pattern generation achieves dynamic balance, these legged robots experience undesired forces due to interaction with the environment, and these forces cause undesired fluctuations in the torso. Therefore, these robots should be sufficiently compliant and have high



Fig. 1. The quadruped robot model in our simulation environment [7].

tracking performance. To achieve these contradictory goals simultaneously, model-based controllers, e.g., inverse dynamics and operational space control, may be very convenient to use [6].

In this sense, contact dynamics is essential when synthesizing a model-based controller. To this end, optimization-based control methods have been developed over the years. To name a few, Righetti et al. proposed an inverse dynamics controller by minimizing the quadratic cost of contact constraints [6]. Herzog et al. proposed a momentum optimal control problem as a Quadratically Constrained Quadratic Program. They reduced the size of the problem by formulating it sequentially while preserving its sparse structure [8]. Moreover, Xin et al. implemented a Cartesian impedance control for tracking and compliance and used Quadratic Programming with friction cone constraints to compute optimal torques closest to desired values obtained via Cartesian impedance control [9].

In a slightly diverse manner, Wensing et al. proposed a prioritized task-space control to develop dynamic kicking and jumping motions for humanoid robots [10]. In this method, they characterized the motions in task space and then solved

the task-space control problem in a real-time conic optimizer which computes joint torques and contact forces at real-time rates of approximately 200 Hz. Likewise, Sentis et al. proposed a compliant control strategy with a multi-level hierarchical control structure that establishes general priorities among behavioral primitives [11].

Contribution: The main objective of the approaches mentioned above is to optimize joint torques using computationally expensive Quadratic Programming (QP) or Conic Optimization Problems (CQP). However, we argue that contact force distribution can be addressed using algorithms that allow real-time processing, and therefore, this paper aims to contribute toward this goal. We propose a method to distribute the contact forces via Virtual Model Control (VMC) by combining it with centroidal momentum feedback, which can be implemented in real-time.

The remainder of this paper is as follows. In section II, the model of the quadruped robot, Kara, is explained. Section III discusses an overview of the locomotion control framework, and VMC and centroidal momentum concepts are explained. In section IV, simulation results are presented and discussed. Finally, the conclusion is presented in section V.

II. QUADRUPEL ROBOT MODEL

At Ozyegin University, we designed and built a high-powered quadruped robot named Kara. A realistic simulation model of the robot was constructed in RaiSim v1.1.5, a physics engine for multi-body systems built in cross-platform [7]; see Fig. 1. The robot has 12 actuated joints driven by custom-built electrical actuators. There are 3 DoFs (Degrees of Freedom) on each leg of the robot, as shown in Fig.2. Two of the actuators are attached to the torso. The formed hip joint thus allows motion along the A/A (Abduction/Adduction) and F/E (Flexion/Extension) axes. The third actuator is also located near the torso and actuate the knee joint along the F/E axis with the help of a parallelogram. All actuators are equipped with absolute encoders with 18-bit resolution. In addition, an AHRS (Attitude and Reference Heading System) unit is placed on the torso to obtain acceleration and orientation information.

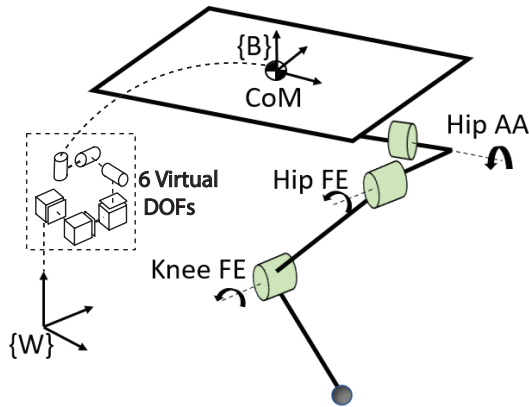


Fig. 2. Joint configuration of quadruped robot Kara. W represents the inertial frame and B represents the floating base frame.

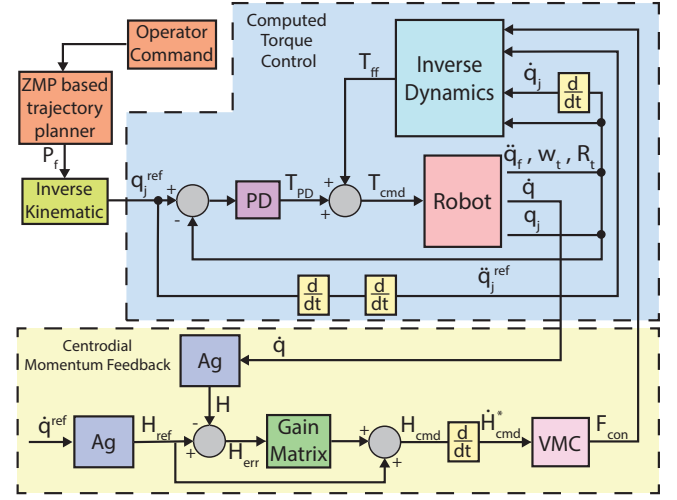


Fig. 3. Overview of locomotion control framework.

III. LOCOMOTION CONTROL

Fig. 3 shows the overview of the locomotion control framework. In this figure, q_j denotes the actuated joint positions of each leg, and q denotes the generalized coordinates of the robot that is defined as $q = [q_f \ q_j]^T$. Here, q_f symbolizes the floating base coordinates. For the dynamic locomotion of the robot, a ZMP-based trajectory planning algorithm was implemented for trot-walking [3]. The trajectory planning algorithm takes the commands from the operator. These commands are mean center of mass (CoM) velocity, torso yaw orientation, single support (T_s), and double support (T_d) durations. The trajectory planner computes the desired foot trajectory for each leg in accordance with the operator's commands. The inverse kinematic algorithm computes the desired joint positions (q_j^{ref}) for each leg. Implemented computed torque control, whose output is command torque T_{cmd} for each joint, which is fed to the system. The computed torque control method consists of two main blocks. The first one is the inverse dynamic block. In this block, system dynamics are modeled using a recursive Newton-Euler algorithm. This recursive Newton-Euler algorithm takes joint positions (q_j), joint velocities ($\dot{q}_j$), generalized floating base accelerations ($\ddot{q}_f$), floating base angular velocities (W_t), floating base orientation (R_t) and desired joint accelerations ($\ddot{q}_j^{ref}$). The second block of computed torque control is the PD controller to compensate for the unmodeled dynamics in the inverse dynamic block. Moreover, centroidal momentum feedback is implemented to compute torque control via centroidal momentum matrix (Ag) and virtual model controller (VMC) to compute contact forces which yields minimizing the fluctuations on the torso. This feedback mechanism is explained in the proceeding subsections.

A. Virtual Model Control

Virtual model control is a control framework that allows the computation of the desired joint torques by rendering virtual components. These virtual components can be defined as

springs, dampers, masses, bearings, and so on [12]. It was reported that the virtual model controller was more advantageous than tuning the joint level controller gains in order to change the dynamic behavior of the system [13]. To that end, desired dynamic behavior of the system can be rendered through virtual model control. For the implementation of virtual model control, desired forces and moments acting on the torso (F_t and M_t) are defined as virtual forces. These virtual forces and moments should be balanced with the ground reaction forces f_c^i , where i indicates which foot has contact with the ground. In doing so, desired ground reaction forces can be computed as follows, in accordance with the desired virtual forces and moments.

$$\begin{bmatrix} f_c^1 \\ \vdots \\ f_c^i \\ \vdots \\ f_c^n \end{bmatrix} = \begin{bmatrix} 1_{3 \times 3} & 1_{3 \times 3} & \dots & 1_{3 \times 3} \\ S(^1P_G) & S(^2P_G) & \dots & S(^iP_G) \end{bmatrix}^\dagger \begin{bmatrix} F_t \\ M_t \end{bmatrix} \quad (1)$$

In (1), $1_{3 \times 3}$ indicates three by three identity matrix, ^iP_G indicates the foot contact position with respect to the center of mass and $S(*)$ indicates the skew-symmetric matrix of an input vector.

B. Centroidal Momentum

The centroidal momentum of a robot is the sum of the projected momentum of each link relative to the combined (composite) CoM [14]. The spatial momentum of each link is a vector consisting of linear and angular momenta. Its calculation concerning the i^{th} link can be expressed below.

$$h_i = I_i v_i \quad (2)$$

In (2), I_i is the spatial inertia of the i^{th} link, and v_i is the spatial velocity of i^{th} link. The spatial inertia of the i^{th} body is calculated as follows:

$$I_i = \begin{bmatrix} \bar{I}_i & m_i S(c_i) \\ m_i S(c_i)^T & m_i \end{bmatrix} \quad (3)$$

In (3), $\bar{I}_i$ is the Cartesian inertia tensor, m_i represents the mass, and c_i is the relative center of mass position of i^{th} body. By summing up all the projected spatial momenta of each link, centroidal momentum is calculated. Thus, the spatial momentum of each link with respect to its body frame can be transformed to the composite CoM by using a spatial momentum transformation matrix (${}^iX_G^T$):

$${}^iX_G^T = \begin{bmatrix} {}^G R_i & {}^G R_i S(^i p_G) \\ 0 & {}^G R_i \end{bmatrix} \quad (4)$$

In Eq. (4), ${}^G R_i$ represents the rotation matrix and ${}^i p_G$ represents the position vector. Finally, centroidal momentum can be derived by adding all the transformed momenta [14].

$$H = \sum_{i=1}^N {}^iX_G^T h_i \quad (5)$$

The centroidal momentum is a function of generalized velocities. Therefore, it can be expressed as below,

$$H = A_g \dot{q}, \quad (6)$$

where H is the centroidal momentum, A_g is the centroidal momentum matrix, and $\dot{q}$ is the generalized velocity vector. A_g matrix can be computed as follows.

$$A_g = {}^1X_G^T \Psi_1^T U_1 M \quad (7)$$

In (7), ${}^iX_G^T$ represents spatial momentum transformation matrix, Φ_1 is the joint matrix which transforms quantity represented in the world frame to floating-base frame and Ψ_1 is defined as $\Psi_1 = \Phi_1^{-1}$. Furthermore, U_1 is defined as $U_1 = [1_{6 \times 6} \ 0_{6 \times 12}]$ where $1_{6 \times 6}$ is identity matrix and $0_{6 \times 12}$ is zero matrix. M is the mass matrix of robot [14].

C. Contact Force Distribution via Centroidal Momentum

Contact force distribution is required for computing the feedforward torques via inverse dynamics. In a naive approach, contact forces can be distributed taking into account the total weight of the robot, and then desired contact forces can be computed accordingly. However, since this method only characterizes vertical forces, it has been observed that this method causes drift in the yaw angle of the robot.

In order to overcome this problem, the centroidal momentum method is used, and contact force distributions are computed via VMC with centroidal momentum feedback; see Fig. 3. To do so, desired centroidal momentum can be computed by multiplying the centroidal momentum matrix with reference states ($\dot{q}^{ref}$) of the robot, and centroidal momentum error H_{err} can be calculated by using reference centroidal momentum H_{ref} . The command centroidal momentum H_{cmd} can be selected as a multiplication of H_{err} with a gain matrix. To compute force distributions, the derivative of the command centroidal momentum H_{cmd} can be selected as virtual components. As a next step, desired contact forces, F_{con} , can be computed in accordance with these virtual components and plugged into the inverse dynamics block of the computed torque controller. In this sense, plugging the derivative of the desired centroidal momentum alone as a virtual component may not minimize the centroidal momentum error. To minimize the centroidal momentum error, H_{err} , a controller with feedback and feedforward controllers is implemented such that the centroidal momentum command, H_{cmd} , is computed.

$$H_{cmd} = H_{ref} + K_h H_{err}, \quad (8)$$

where K_h is a diagonal matrix that stores centroidal feedback controller gains. Furthermore, the gravitational force acting on the whole system is directly added to the derivative of the linear centroidal momentum command; $\dot{H}_{cmd}^* = \dot{H}_{cmd} + [0 \ 0 \ mg \ 0 \ 0 \ 0]^T$ where m and g represent the mass of the whole system and gravity respectively. Finally, the derivative of the centroidal momentum command ($\dot{H}_{cmd}^*$) is

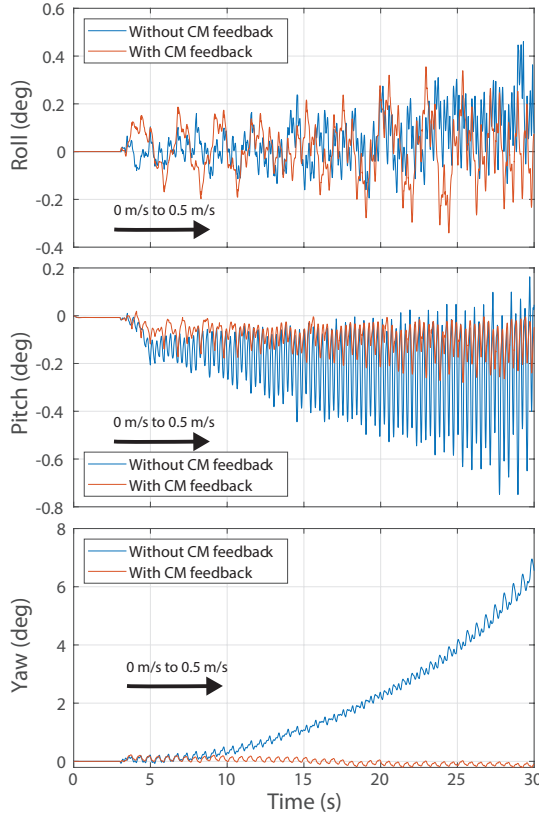


Fig. 4. Torso orientation, scenario 1.

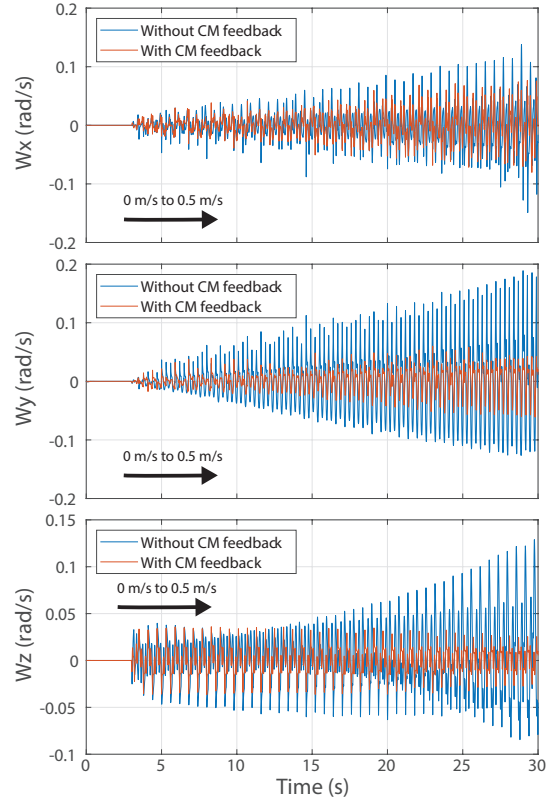


Fig. 5. Torso angular velocity, scenario 1.

selected as a virtual component of VMC, and finally, F_{con} is yielded:

$$F_{con} = \begin{bmatrix} 1_{3 \times 3} & 1_{3 \times 3} & \dots & 1_{3 \times 3} \\ S(^1P_G) & S(^2P_G) & \dots & S(^iP_G) \end{bmatrix}^{\dagger} \dot{H}_{cmd}^* \quad (9)$$

IV. SIMULATION RESULTS

Simulation experiments were conducted in RaiSim v1.1.5 as explained in Section II. Three different scenarios were considered: i) trot-walking at 0.5 m/s, ii) trot-walking at 0.5 m/s with a 4 kg payload, and iii) trot-walking with a sinusoidal rolling task. In each scenario, simulations were conducted two times: First, centroidal momentum feedback was removed, and contact forces were simply distributed by checking the contact feet and dividing the robot's total weight to contact feet equally. Second, the simulation scenario was repeated with the proposed contact force distribution method in effect. It is ensured that the computed contact forces always exists within the friction cone [15].

A. Scenario 1: Trot-Walking at 0.5 m/s

In this scenario, the robot started trot-walking after 3 seconds, and its mean velocity was increased with a ramp input from 0 m/s to 0.5 m/s (slope: 0.02). In Fig. 4, torso Euler angles (Roll, Pitch, Yaw) and angular velocities (ω_x , ω_y , ω_z) are compared. The blue lines indicate the behavior of the torso without the proposed method, while the orange

lines show the torso behavior with the proposed centroidal momentum feedback. Examining Fig. 4 and Fig. 5, RMS (Root Mean Square) error values concerning the torso orientation were 0.1089deg in roll, 0.2386deg in pitch, 2.5709deg in yaw when the proposed controller was not activated. On the contrary, these values were dropped to 0.1053deg, 0.0999deg, 0.0986deg, respectively. As a result, we observed a significant improvement in the yaw axis. The robot heading was truly well-regulated. Furthermore, pitch axis orientation was also quite improved, whereas the decrease in roll axis was relatively insignificant. A positive outcome was also observed in torso angle fluctuations; the RMS error values of angular velocity measurements along the roll, pitch, and yaw axes decreased by 29.7%, 65.8%, 61.7%, respectively.

The robot's dynamic balancing ability during walking was also improved with the help of the proposed contact force distribution method. When it was activated, the RMS values of ZMP errors along the x-axis and y-axis decreased for about 15.0% and 64.2%, respectively, during trot walking with increasing COM velocity. See Fig. 6 for changes in ZMP errors concerning both the x-axis and y-axis.

B. Scenario 2: Trot-Walking with a 4 kg Payload

In this scenario, a 4 kg payload was added to the top of the torso and trot-walking with the same parameters concerning the first scenario was applied. As seen in Fig. 7, the proposed method caused RMS error values to decrease in the torso

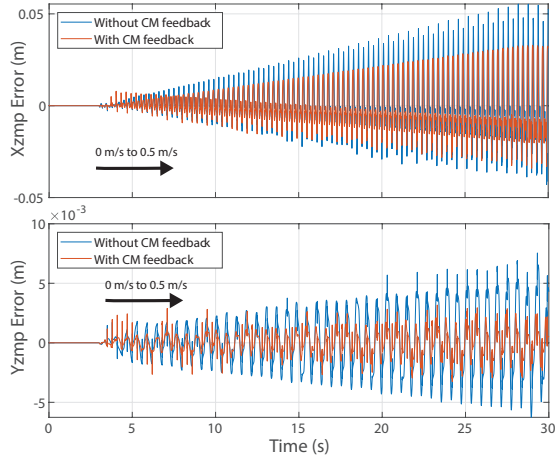


Fig. 6. ZMP error, scenario 1.

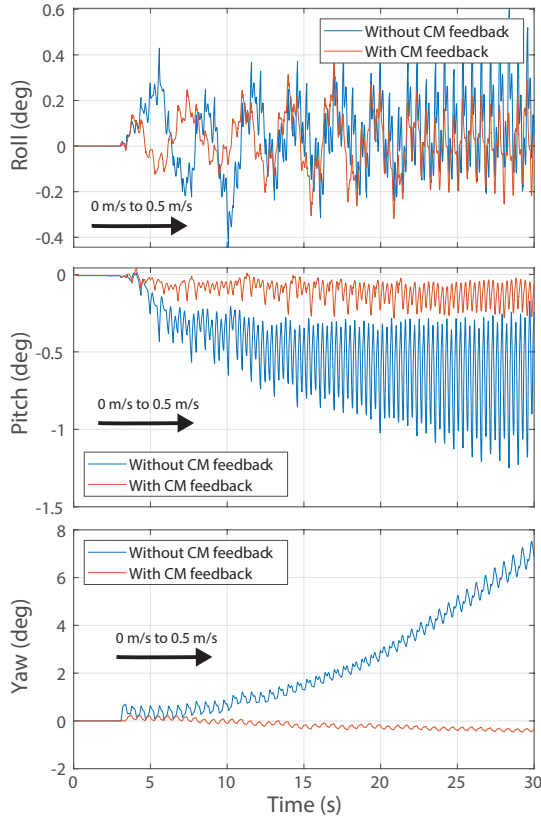


Fig. 7. Torso orientation, scenario 2.

orientation; namely, 29.1% in roll angle, 78.7% in pitch angle, 92.5% in yaw angle. Furthermore, RMS error values of the torso angular velocities decreased by approximately 56.0% along the roll axis, 73.8% along the pitch axis, and 72.0% along the yaw axis; see Fig. 8.

Observing Fig. 9, RMS value of X_{zmp} error dropped from 0.0153 m to 0.0109 m and RMS value Y_{zmp} error decreased from 0.003 m to 7.8×10^{-4} m. Hence, centroidal momentum feedback with the proposed control scheme minimizes the

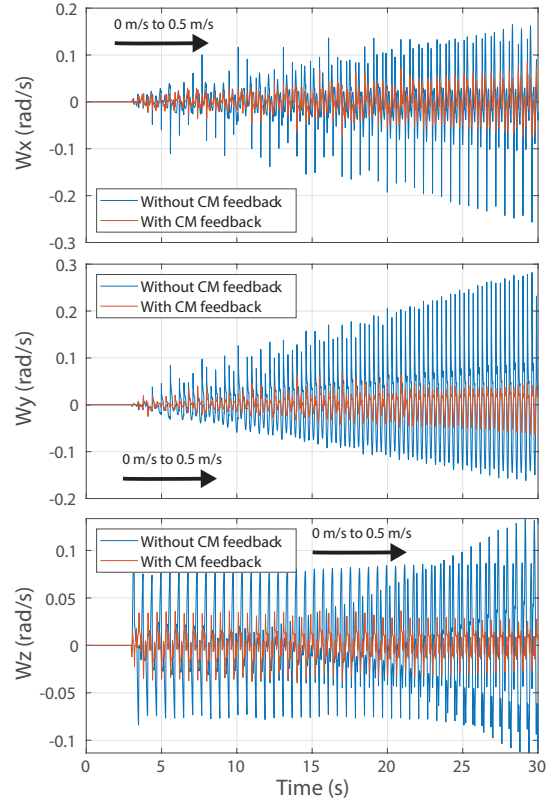


Fig. 8. Torso angular velocity, scenario 2.

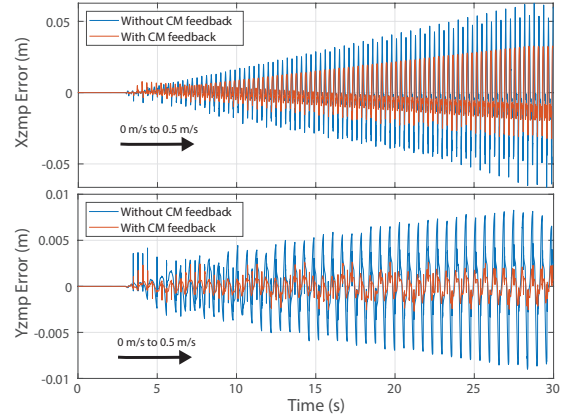


Fig. 9. ZMP error, scenario 2.

ZMP errors and increases the balancing capability of the robot despite the 4 kg payload.

C. Scenario 3: Trot-Walking with Sinusoidal Rolling Task

In this scenario, sine input with a frequency of 0.2 Hz and amplitude of 10 deg in the roll angle is set for the torso of the robot while the robot is walking with increasing COM velocity. Centroidal momentum feedback decreases the fluctuations in the orientation of the torso while keeping the torso orientation at desired input; see Fig. 10. RMS errors of the torso roll, pitch, and yaw angles decrease from 0.2258 deg, 0.3849 deg,

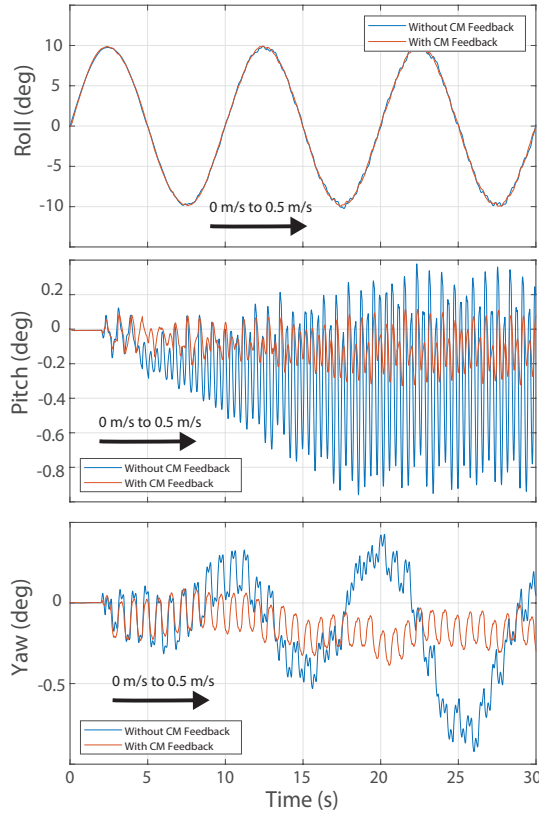


Fig. 10. Torso orientation, scenario 3.

0.3242 deg to 0.1857 – deg, 0.1292 deg, 0.1676 deg, respectively.

V. CONCLUSION

To conclude, this paper proposed a control method to eliminate undesired orientation fluctuations on the torso and regulate heading during dynamic trot-walking by distributing the desired contact forces in accordance with the centroidal momentum feedback with virtual model control. The proposed algorithm was tested in a simulation environment that included a realistic model of our quadruped robot Kara. In these simulations, three different scenarios were considered: i) dynamic trot-walking at 0.5 m/s, ii) dynamic trot-walking at 0.5 m/s with a 4 kg payload, and iii) dynamic trot-walking with a sinusoidal rolling task. All simulations showed that the proposed controller decreased the fluctuations regarding the torso orientation and significantly regulated the heading of the robot during trot-walking. In addition, the ZMP error was reduced concerning all scenarios.

The aforementioned simulations are performed at a sampling rate of 1 kHz, where the computational load of the proposed method takes, on average, only 36.26 μ s per cycle. This result implies that the proposed controller can be implemented in real-time. In order to implement this controller on robot hardware, floating base linear and angular velocities are needed. The angular velocity of the floating base can be obtained from IMU; however, IMU only measures linear accel-

eration. To overcome this issue, a state estimation algorithm could be used to obtain floating base linear velocities [16]. The proposed controller will be implemented in our quadruped robot Kara in future work.

ACKNOWLEDGMENTS

The authors thank J. Hwangbo for providing free academic licenses of the Raisim simulator. www.raisim.com

Two representative video clips from the simulation studies can be viewed via <https://youtu.be/zZsdMjYjkyk>

REFERENCES

- [1] E. Guizzo, “By leaps and bounds: An exclusive look at how boston dynamics is redefining robot agility,” *IEEE Spectr.*, vol. 56, no. 12, pp. 34–39, Dec. 2019.
- [2] G. Bellegarda and A. Ijspeert, “CPG-RL: Learning central pattern generators for quadruped locomotion,” *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.
- [3] B. Ugurlu, I. Havoutis, C. Semini, K. Kayamori, D. G. Caldwell, and T. Narikiyo, “Pattern generation and compliant feedback control for quadrupedal dynamic trot-walking locomotion: experiments on RoboCat-1 and HyQ,” *Autonomous Robots*, vol. 38, no. 4, pp. 415–437, Feb. 2015.
- [4] B. Katz, J. D. 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)*. IEEE, May 2019.
- [5] M. Vukobratović and B. Borovac, “Zero-Moment Point — Thirty Five Years of Its Life,” *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, Mar. 2004.
- [6] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, “Control of legged robots with optimal distribution of contact forces,” in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Oct. 2011, pp. 318–324.
- [7] J. Hwangbo, J. Lee, and M. Hutter, “Per-contact iteration method for solving contact dynamics,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018. [Online]. Available: www.raisim.com
- [8] A. Herzog, S. Schaal, and L. Righetti, “Structured contact force optimization for kino-dynamic motion generation,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 2703–2710.
- [9] G. Xin, W. Wolfslag, H.-C. Lin, C. Tiseo, and M. Mistry, “An optimization-based locomotion controller for quadruped robots leveraging cartesian impedance control,” *Frontiers in Robotics and AI*, vol. 7, Apr. 2020.
- [10] P. M. Wensing and D. E. Orin, “Generation of dynamic humanoid behaviors through task-space control with conic optimization,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 3103–3109.
- [11] L. Sentis and O. Khatib, “Synthesis of whole-body behaviors through hierarchical control of behavioral primitives,” *International Journal of Humanoid Robotics*, vol. 02, no. 04, pp. 505–518, Dec. 2005.
- [12] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, “Virtual model control: An intuitive approach for bipedal locomotion,” *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, Feb. 2001.
- [13] C. D. Remy, M. Hutter, M. Hoepflinger, M. Bloesch, C. Gehring, and R. Siegwart, “Quadrupedal robots with stiff and compliant actuation,” *Automatisierungstechnik*, vol. 60, no. 11, pp. 682–691, Nov. 2012.
- [14] 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, Mar. 2016.
- [15] Y. Fujimoto and A. Kawamura, “Simulation of an autonomous biped walking robot including environmental force interaction,” *IEEE Robotics and Automation Magazine*, vol. 5, no. 2, pp. 33–42, Jun. 1998.
- [16] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, “Probabilistic contact estimation and impact detection for state estimation of quadruped robots,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, Apr. 2017.