Push Recovery of a Quadrupedal Robot in the Flight Phase of a Long Jump

Beste Bahceci, Omer Kemal Adak, and Kemalettin Erbatur
Sabanci University, İstanbul, Turkey
Email: {bestebahceci, omerkemal, erbatur}@sabanciuniv.edu

***Abstract*—Legged robots are well-suited for operation in challenging natural environments, such as steep obstacles or vast gaps in the ground. Aside from difficult terrain, robots may also encounter unanticipated impact forces while performing jumping gaits. When performing their gaits, legged robots should be able to maintain and regain their stability in the face of external perturbations. External disturbances should be detected, and necessary actions should be taken to maintain the robot's balance in order to ensure optimum landing conditions. This paper considers flight phase disturbances in the form of a push on the robot body and introduces a novel push recovery algorithm that uses angular momentum to generate reference trajectories for a quadrupedal robot with waist joints during the flight phase of a long jump. This method creates joint position reference trajectories for the quadrupedal robot's waist and rear hip joints in order to achieve the required orientation of the robot in the air. In order to track reference trajectories, PID joint control is utilized. The robot model employed for the computations is comprehensive because components of the robot body - the leg links and three torso sections - are represented with independent mass values. The proposed push recovery trajectory generation approach is computationally efficient and hence suitable to be employed in real-time applications. The suggested method is used to simulate a quadrupedal robot to test the push recovery algorithm following external disturbances in the flight phase of a long jump. The results demonstrate that the suggested approach performs well in terms of angular position and angular velocity accuracy and it can achieve a posture suitable for landing.**

***Index Terms*—quadrupedal robot, free-fall manipulators, legged robots, jumping motion, trajectory generation, push recovery**

I. Introduction

Mammals have an uncanny ability to plan and execute complicated actions appropriate to the terrain and job at hand, for instance, leaping to cross a gap in the terrain. Although animals can successfully navigate rough terrains, nature-inspired robots need to plan and execute dynamic behaviors to navigate the most difficult terrains. Legged robots have a significant capability for use in challenging natural environments. Legs give significant mobility for negotiating uneven terrain, particularly high barriers or large gaps in the terrain. In addition to rough terrain conditions, robots can face unexpected impact forces while performing jumping gaits. Being pushed by another robot or living organism, bumping into trees or rocks, and being hit with foreign objects are among examples of external effects. Legged robots should successfully maintain and restore their stability against external disturbances when performing their gaits in an irregular natural environment or lacking contact points. The external disturbance should be noticed as early as possible, and appropriate measures should be taken to preserve the balance of the robot in order to achieve conditions for successful landing. Posture recovery after a push on the robot body can enhance the functionality of a legged robot greatly.

Several study data address a robot's capacity to restore its position following a disruption of the robot movement. One of the main strategies for balance recovery of a bipedal robot while walking after an external disturbance is the machine learning method [1]-[13]. Foot placing strategies in order to regulate the center of mass (COM) position of humanoid robots in response to external perturbation are employed widely [14]. In addition to foot placing, replanning locomotion mode strategies involving walking, running, and hopping can be employed [15]. Zero Moment Point (ZMP) based walking pattern generation is another approach for foot placement of bipedal robots [16], [17]. Foot placing strategy is an effective way of maintaining the stability of robot gaits. However, it can be used only when there is contact with the ground.

Humanoid robots are widely modeled via Linear Inverted Pendulum Models (LIPM) to adjust the trajectory of the center of mass (COM) in order to maintain balance and stability in walking and push recovery [18]-[24]. LIPM can also be employed for quadrupedal robots. LIPM based foot placement strategy for push recovery of a quadrupedal walking robot is proposed in [25]. A linear inverted pendulum and flywheel model is applied on a quadrupedal robot modeled as a bipedal robot to achieve stable walking while being subject to an external disturbance [26]. In the LIPM, the robot is modeled as single mass and robot parts such as legs are modeled massless. This leads to relatively large modeling errors. In [27], an optimization problem is defined to calculate the admissible main body accelerations such that the quadrupedal robot achieves a stable gait and the support

Manuscript received January 21, 2022; revised June 7, 2022 .

doi: 10.18178/ijmerr.11.7.486-493

feet do not slip under external disturbances. A foot placement strategy has been developed for a quadrupedal robot in order to achieve a stable trot gait with a stable posture in [28].

Angular momentum is one of the main variables of concern for posture stabilization of legged robots while performing hard to balance gaits or for push recovery scenarios. The flywheel with the LIPM provides a technique to include the angular momentum of the robot for posture stabilization. The method is employed for push recovery of bipedal robots while performing walking gaits [29]-[32]. Inverted pendulum model with angular momentum is also employed for push recovery of bipedal walking [33]-[35], and postural stability [36]. Angular momentum can also be used for footstep modification for bipedal robots in maintaining the balance of walking gaits [37], [38] and push recovery scenarios [39], [40]. Another similar method is planning angular-momentum-rate-of-change trajectories to reduce recovery step length after external disturbances [41]. Hip and ankle strategies can be used with angular momentum for postural balancing of a bipedal robot [42] or walking stabilization under external perturbations [43]-[46]. Angular momentum of COM can also be utilized for whole-body controllers to regulate joint torques to achieve stable walking and push recovery of legged robots [47]-[49]. Angular momentum calculations in these studies are mainly performed for a single mass located at the COM position of the legged robots. Consequently, the arms and legs of robots are not included in angular momentum computations in these models. This causes modeling errors. In [50], a five-mass model for angular momentum calculations is presented for push recovery of the humanoid robot model to obtain a more accurate model.

The common factor of the above mentioned methods is that robots have at least one contact point with the ground. Therefore, robots can be modeled as linear inverted pendulums, or foot placing strategies can aid robots to balance since ground reaction forces can be employed to cancel angular momentum caused by external disturbances. In the flight stage of a long jump, robots do not contact the ground, and therefore ground reaction forces are no longer applicable. In addition, modeling a robot in the flight stage with a single mass is impractical since the angular momentum of the robot will solely depend on the motion of various robot parts such as legs.

This paper proposes a novel reference generation algorithm using angular momentum for push recovery of a quadrupedal robot with waist joints in the flight phase of a long jump. This algorithm generates reference curves for waist joints and rear hip joints of the quadrupedal robot in order to achieve the desired orientation of the robot in the air. The robot model used for the calculations is detailed, so every part of the robot body, such as legs and three torso parts, is modeled as an individual mass. Moreover, the trajectory generation method is computationally efficient enough to be used in real-time applications. Finally, the proposed algorithm is employed in a quadrupedal robot simulation to demonstrate posture recovery after external disturbances in the flight phase.

II. Methodology

A. Push Recovery Algorithm

The trajectory generation method uses the desired and actual values of body angular position and velocity. It is assumed that these variables are available. Since the workspace of the robot is three dimensional, three equations can be employed. In order to achieve desired angular position and velocity, three joints are chosen to manipulate the overall angular momentum of the robot body in the flight stage of a long jump. The discretized angular momentum expression is

$$L(t+\Delta t)-L(t)=\tau\times\Delta t, \qquad (1)$$

where t is time, $L(t)$ is angular momentum with respect to robot coordinate frame (Fig. 1) at time t, Δt is a single time step in discretization, and τ is the torque acting on the quadrupedal robot. τ can be calculated as

$$\tau=r^{COM}\times M\times g, \qquad (2)$$

where r^{COM} is the position of robot COM with respect to robot coordinate frame, M is the total mass of the robot, and g is the gravitational acceleration. Angular momentum at time t can be found by

$$L(t)=I_b\times\omega_b(t), \qquad (3)$$

where I_b is the moment of inertia of the total robot body with respect to the robot coordinate frame. $\omega_b(t)$ is the angular velocity of the robot body. Angular momentum at time $t+\Delta t$ can be expressed as

$$L(t+\Delta t)=I_b\times\omega_{b,d}(t+\Delta t)+I_i\times\omega_i(t+\Delta t) \qquad (4)$$

Here, $\omega_{b,d}$ is the desired angular velocity of the robot body, ω_i is the angular velocity of a chosen robot joint and I_i is the moment of inertia of the moving robot part because of chosen robot joint. The moment of inertia of different components of the quadrupedal robot is calculated such as

$$\begin{aligned} I_{cx}&=\frac{1}{12}m\times(l_y^2+l_z^2)\\ I_{cy}&=\frac{1}{12}m\times(l_x^2+l_z^2)\\ I_{cz}&=\tfrac{1}{12}m\times(l_x^2+l_y^2), \end{aligned} \qquad (5)$$

where I_{cx}, I_{cy} and I_{cz} are the moments of inertia measured about x, y, and z axes stationed at the center of mass of the rectangular prism that describes a robot body part (Fig. 2). In order to calculate the moments of inertia with respect to robot coordinate axes, the parallel axis theorem is employed

$$I_i=I_{ci}\times m_i\times {d_i}^2 \qquad (6)$$

Here, I_i is the moment of inertia of ith body component, m_i is mass of ith body component, and d_i is the distance between axes. For instance when calculating the moment of inertia around y-axis for middle torso component, first I_{cy} is calculated around y_{1c}, which is the y-axis through the COM of the body part and I_i is computed with the parallel axis theorem. As can be seen in Fig. 2, d_i is the distance between y_{1c} and the y-axis of robot coordinate frame. Since the angular position of the robot, equivalently, roll, pitch, and yaw angles ($\alpha, \beta,$ and γ) are needed to be manipulated, three joint angles are chosen as action variables to generate a trajectory. Hence, three independent variables are utilized to control the three orientation angles. The joint variables used are the two spine joint angles θ_b, $\emptyset_b$ and the two rear leg joint angles $\theta_{ullr}, \theta_{ulrr}$. Rear leg joint angles are constrained to be equal in magnitude and opposite in direction for applying maximum precession of the overall angular momentum around the z-axis of the robot's coordinate frame. Therefore, they can be treated as a single independent variable. These joints are chosen since their main inertial axes are at y, x, and z directions, respectively. In order to simplify computations and to decrease computational time, chosen joint angles are assumed to affect only their corresponding main inertial axes. For instance, it is assumed that, change in θ_{ullr} effects only angular momentum around the z-axis. After calculating the moments of inertia and combining (1), (2), (3) and (4), the following equations are obtained.

$$\dot{\emptyset}_b = \frac{\tau_x + I_{b,x} \times \omega_{b,x}(t) - I_{b,x} \times \omega_{b,d,x}(t+\Delta t)}{I_{i,x}}$$

$$\dot{\theta}_b = \frac{\tau_y + I_{b,y} \times \omega_{b,y}(t) - I_{b,y} \times \omega_{b,d,y}(t+\Delta t)}{I_{i,y}} \qquad (7)$$

$$\dot{\theta}_{ullr} = \frac{\tau_z + I_{b,z} \times \omega_{b,z}(t) - I_{b,z} \times \omega_{b,d,z}(t+\Delta t)}{2 \times I_{i,x}}$$

Here, $\dot{\emptyset}_b$, and $\dot{\theta}_b$ are angular velocities of spine joints. $\dot{\theta}_{ullr}$ represts angular velocities of the rear leg joints. τ_x, τ_y and τ_z are the torque components acting on the quadrupedal robot about *x*, *y* and *z*-axes, respectively, as expressed in the robot coordinate frame. $\omega_{b,x}, \omega_{b,y}$ and $\omega_{b,z}$ are the angular velocity component of the robot body in the robot coordinate frame. $I_{b,x}$, $I_{b,y}$ and $I_{b,z}$ stand for the moments of inertia of the total robot body with respect to the robot coordinate frame, about *x*, *y* and *z*-axes. $\omega_{b,d,x}, \omega_{b,d,y}$ and $\omega_{b,d,z}$ are the desired angular velocities of the robot body around *x*, *y* and *z*-axes in the same coordinate frame. Desired angular velocities can be computed as

$$\omega_{b,d,x}(t+\Delta t) = \tfrac{\alpha_d - \alpha}{\Delta t},$$

$$\omega_{b,d,y}(t+\Delta t) = \tfrac{\beta_d - \beta}{\Delta t}, \qquad (8)$$

$$\omega_{b,d,x}(t+\Delta t) = \frac{\gamma_d - \gamma}{\Delta t}$$

And

$$\tau_x = r_x{}^{COM} \times M \times g \times \Delta t,$$

$$\tau_y = r_y{}^{COM} \times M \times g \times \Delta t, \qquad (9)$$

$$\tau_z = r_z{}^{COM} \times M \times g \times \Delta t,$$

where α_d, β_d and γ_d are the desired roll, pitch and yaw angles, respectively.

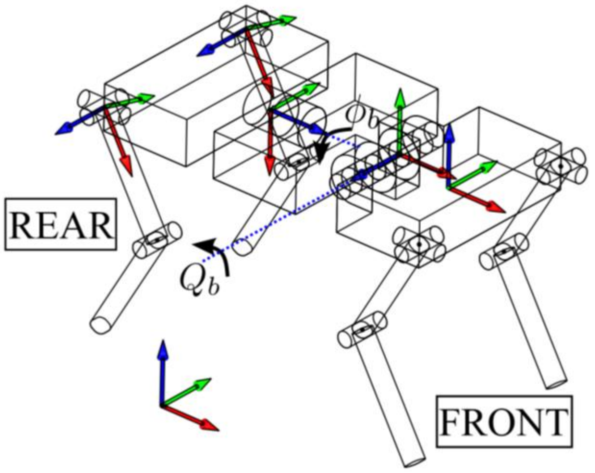

Figure 1. Quadruped robot model for simulation. θ_b is front spine joint angle around corresponding blue axis, $\emptyset_b$ is rear spine joint angle around the corresponding blue axis and θ_{ulrr}, θ_{ullr} are rear hip joint angles about corresponding blue axes. $\alpha, \beta,$ and γ are body orientation angles about the world coordinate axes (red, green, and blue axes).

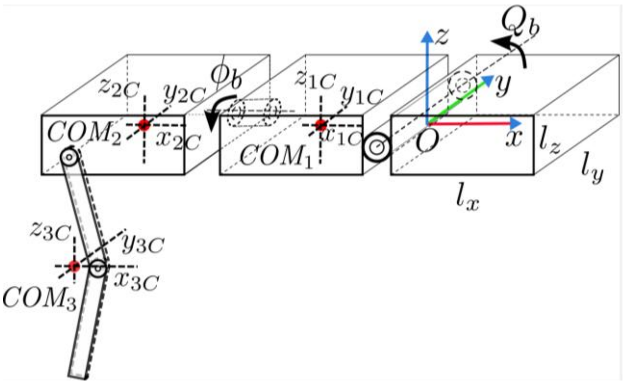

Figure 2. Quadruped robot model for the moment of inertia computations. COM_1 is the center of mass of the middle torso component, COM_2 is the center of mass of rear torso component and COM_3 is the center of mass of the rear leg. The robot coordinate frame axes is shown with three different colors (red for *x*-axis, green for *y*-axis, and blue for *z*-axis).

B. *Application of Push Recovery Algorithm on Quadrupedal Robot*

The quadruped robot kinematic arrangement in Fig. 1. is used for reference trajectory computation. The suggested quadruped model features 20 Degrees of Freedom (DOF), with three DOF on each leg and two at the spine. In addition, six DOF are added for the robot body's linear and angular position. Every joint in the legs and body is revolute. The hip hosts an adduction/abduction joint. There are flexion/extension joints at the hip and the knee. Three distinct body building blocks coupled by rotational joints are utilized for dynamic spine motion. The front spine joint revolves around the *y*-axis of the robot coordinate frame for pitch motion, whereas the rear spine joint axis is the *x*-axis of the robot coordinate frame for roll motion. In order to produce a realistic robot model, all joint angles are restricted to be between -45 and 45 degrees. Home position of the robot can be seen in Fig. 1. The overall mass of the robot is 90 kg. It consists of a body mass of 50 kg and leg masses of 10 kg each. Front and rear body parts are 20 kg each and middle body part weighs 10 kg.

When the front spine joint moves, two torso components, and two leg components move with it; therefore front body joint manipulates a mass of 50 kg; likewise rear spine joint manipulates 40 kgs. However, rear leg joints manipulate 20 kgs for two legs.

Since the moment of inertia is a function of mass, and rear leg joints are chosen to regulate angular position around the *z*-axis, the performance of manipulating angular position around the *z*-axis is weaker than *x* and *y*-axes.

After reference trajectories are computed with push recovery algorithm, calculated trajectories are applied in quadrupedal robot simulation. Robot model used in simulation is the same model used for reference trajectory generation. Lagrangian approach is employed to derive the equations of motion [51]. In order to track obtained reference trajectories a proportional-integral-derivative (PID) torque controller is employed in the simulation environment.

III. Simulation Results and Discussion

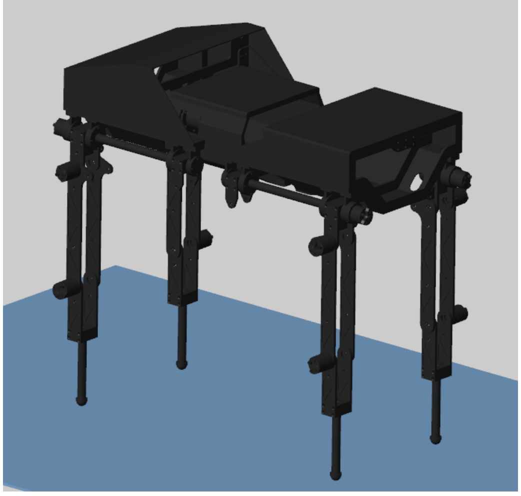

Figure 3. Quadrupedal robot model in simulation environment

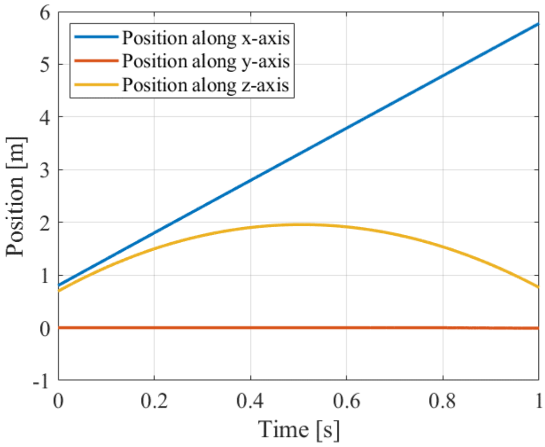

Figure 4. Linear positions of the quadrupedal robot performing jumping gait along the x-axis (blue), y-axis (red), and z-axis (yellow).

In order to create a jumping scenario, take-off velocity is set to 5 m/s at *x* and *z*-directions. The resulting motion has a flight phase of one second. Quadrupedal robot in simulation environment is presented in Fig. 3. Linear positions of the quadrupedal robot during jump can be seen in Fig. 4. Disturbances are generated and applied to the robot model at 0.2, 0.5, and 0.8 seconds after the jump. Various torques are applied around *x*, *y*, and *z*-axes as expressed in the robot coordinate frame, resulting in equal rise in angular acceleration around these axes (see Fig. 5). In the figure, applied torques are 28.7 Nm, 76.6 Nm, and 102 Nm around *x*, *y*, and *z*-axes, respectively. Push recovery algorithm calculates joint trajectories for front and rear joints, right and left hip adduction/abduction joints. Conventional PID control is employed to track resulting joint reference trajectories. Calculated joint reference trajectories and actual joint angles are presented in Fig. 6 to Fig. 9. In Fig. 6, reference and actual positions of front spine joint are presented. The PID controller is quite successful. The rear spine joint actual and reference positions can be seen in Fig. 7. Actual positions and reference trajectories of the right and left hip adduction/adduction joint positions are presented in Fig. 8 and Fig. 9.

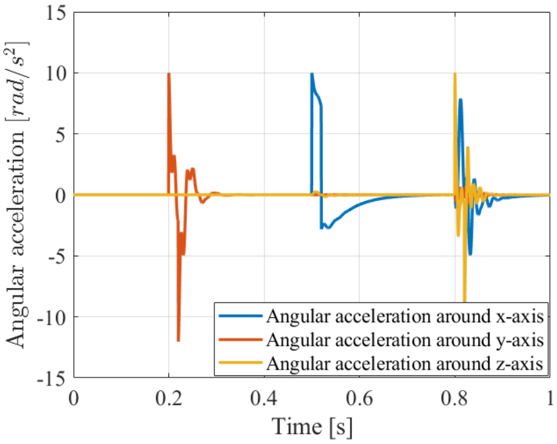

Figure 5. Angular accelerations of the quadrupedal robot around the x-axis (blue), y-axis (red), and z-axis (yellow) with push recovery algorithm.

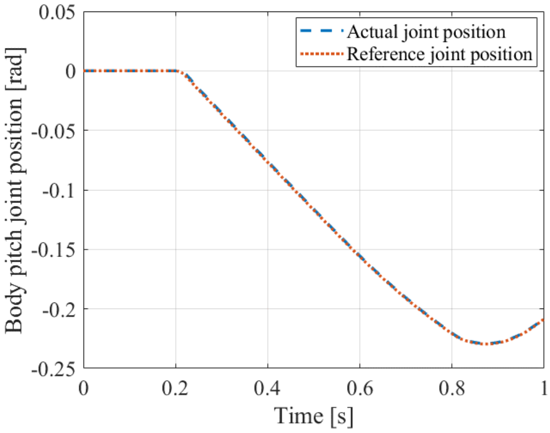

Figure 6. Front spine joint (body pitch joint) actual position (blue) and reference position (red) with push recovery algorithm.

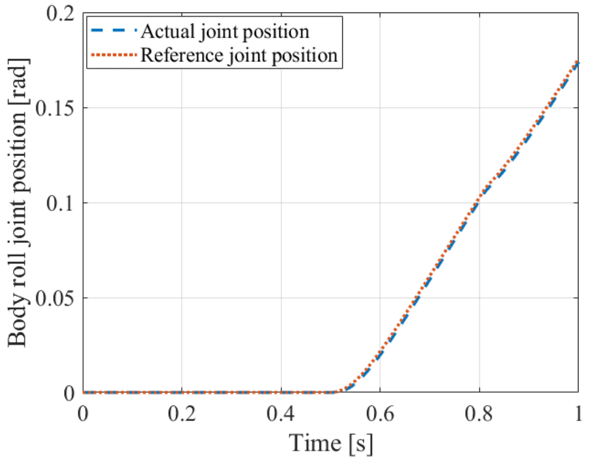

Figure 7. Rear spine joint (body roll joint) actual position (blue) and reference position (red) with push recovery algorithm.

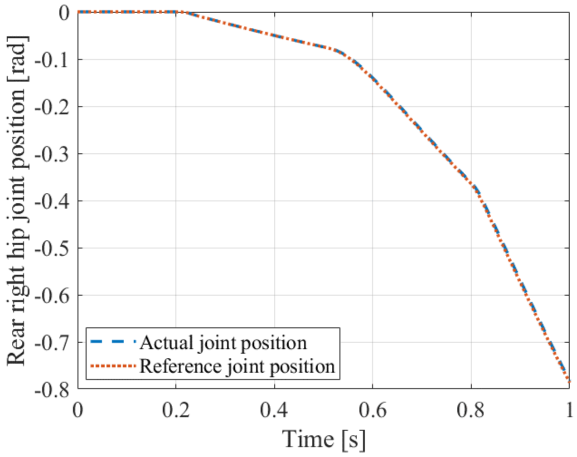

Figure 8. Rear right hip joint actual position (blue) and reference position (red) with push recovery algorithm.

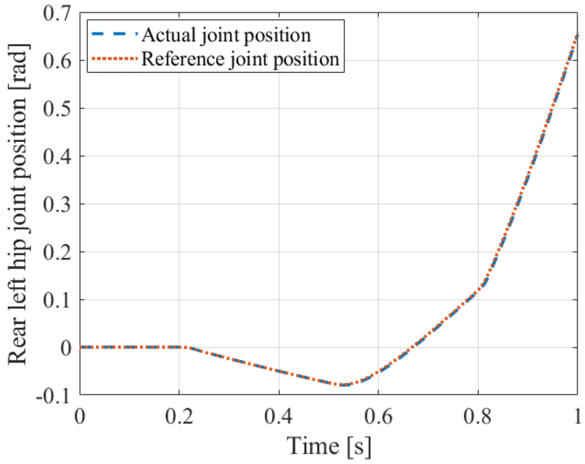

Figure 9. Rear left hip joint actual position (blue) and reference position (red) with push recovery algorithm.

In Fig. 10, roll, pitch, and yaw angles of the quadrupedal robot with respect to the world coordinate frame are presented. After the external disturbance, desired angular position for the robot to reach is chosen to be zero radians around *x*, *y*, and *z*-axes. Results show that maximum error is around 0.009 rad around the *x*-axis, 0.001 rad around the *y*-axis, and 0.0005 rad around the *z*-axis.

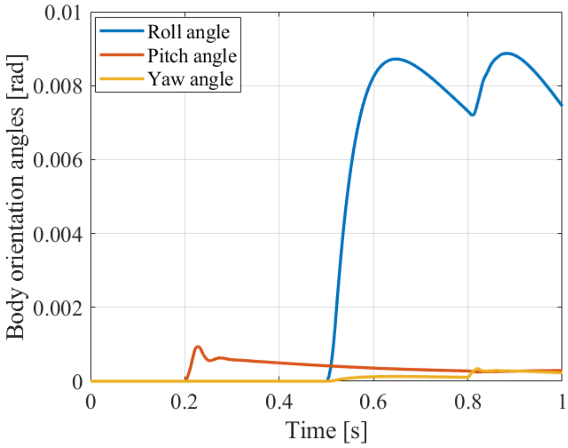

Figure 10. Angular positions (roll, pitch, and roll) of the quadrupedal robot around the world coordinate frame x-axis (blue), y-axis (red), and z-axis (yellow) with push recovery algorithm.

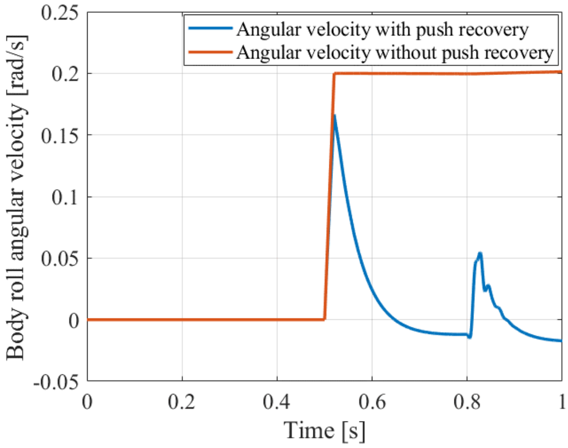

Figure 11. Angular velocity around the x-axis, the blue line is results with the push recovery algorithm, the red line is results without push recovery algorithm.

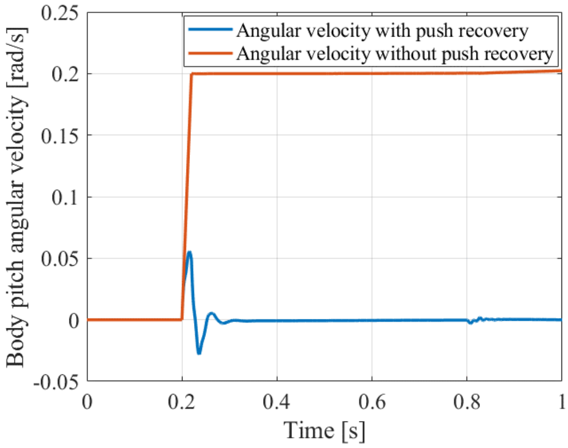

Figure 12. Angular velocity around the y-axis, the blue line is results with the push recovery algorithm, the red line is results without push recovery algorithm.

In order to display the effect of the push recovery algorithm, the same scenario is applied to the quadrupedal robot model without a push recovery algorithm. Fig. 11 presents a comparison between angular velocity around the *x*-axis with and without this algorithm. In the absence of the push recovery system, the angular velocity of the robot increases and settles to 0.2 rad/s. In contrast, when the push recovery algorithm is activated, angular velocity does not reach 0.2 rad/s, and instead it drops down to around 0 rad/s. Similarly, angular velocity around the *y*-axis reaches 0.2 rad/s without the push recovery algorithm since angular accelerations are set equal. After algorithm is applied, it peaks to 0.05 rad/s and converges to zero after 0.2 seconds (Fig. 12). Lastly, in Fig. 13, angular velocity around the *z*-axis with and without the push recovery algorithm is presented. Like with previous results, after applying the algorithm, angular velocity does not increase up to 0.2 rad/s and settles to about zero rad/s in 0.1 seconds.

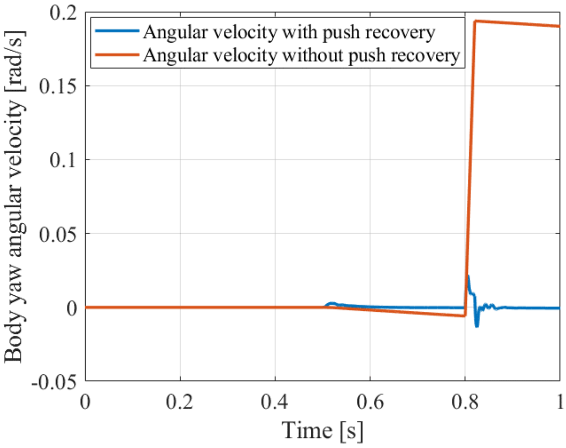

Figure 13. Angular velocity around the z-axis, the blue line is results with the push recovery algorithm, the red line is results without push recovery algorithm.

IV. Conclusion

This study presents a novel push recovery algorithm for the flight phase of legged robot jumping gaits. The algorithm of recovery is based on angular momentum variables. Reference trajectories are obtained for a quadrupedal robot with waist joints in the flight phase of a long jump. The method generates joint position reference trajectories for waist joints and rear hip joints of the quadrupedal robot to reach the desired orientation of the off-ground robot. Computed trajectories are applied on the robot in the simulation environment. The simulation calculates robot's orientation with these trajectories. PID control is employed to track the joint reference trajectories. The model used is detailed with individual link masses taken into account and it is computationally efficient to be used in real-time applications.

Lastly, the proposed push recovery algorithm is tested in the simulation environment with external disturbances in the flight phase of a long jump. Results show that the algorithm achieves small orientation and angular velocity errors and is a candidate for implementation work.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Beste Bahçeci worked on theoretical work. Beste Bahçeci and Ömer Kemal Adak simulated the robot model. Beste Bahçeci and Kemalettin Erbatur wrote the paper. All authors shared ideas and discussed results, and all authors had approved the final version.

Acknowledgment

This study is supported financially by Sabanci University with a tuition waiver of Beste Bahçeci and Ömer Kemal Adak.

References

[1] V. B. Semwal, S. A. Katiyar, R. Chakraborty, and G. C. Nandi, “Biologically-inspired push recovery capable bipedal locomotion modeling through hybrid automata,” *Robotics and Autonomous Systems*, vol. 70, pp. 181–190, Aug. 2015.

[2] D. Ferigo, *et al*., “On the emergence of whole-body strategies from humanoid robot push-recovery learning,” arXiv:2104.14534 [cs, stat], Apr. 2021.

[3] A. Parashar, A. Parashar, S. Goyal, and B. Sahjalan, “Push recovery for humanoid robot in dynamic environment and classifying the data using K-Mean,” in *Proc. Second International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '16*, Udaipur, India, 2016, pp. 1–6.

[4] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee, “Learning full body push recovery control for small humanoid robots,” in *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2047–2052.

[5] D. Luo, X. Han, Y. Ding, Y. Ma, Z. Liu, and X. Wu, “Learning push recovery for a bipedal humanoid robot with dynamical movement primitives,” in *Proc. IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Seoul, South Korea, Nov. 2015, pp. 1013–1019.

[6] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee, “Online learning of a full body push recovery controller for omnidirectional walking,” in *Proc. 11th IEEE-RAS International Conference on Humanoid Robots*, Bled, Slovenia, Oct. 2011, pp. 1–6.

[7] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee, “Online learning of low dimensional strategies for high-level push recovery in bipedal humanoid robots,” in *Proc. IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 1649–1655.

[8] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee, “Practical bipedal walking control on uneven terrain using surface learning and push recovery,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sept. 2011, pp. 3963–3968.

[9] H. Kim, D. Seo, and D. Kim, “Push recovery control for humanoid robot using reinforcement learning,” in *Proc. Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, Feb. 2019, pp. 488–492.

[10] V. B. Semwal, K. Mondal, and G. C. Nandi, “Robust and accurate feature selection for humanoid push recovery and classification: Deep learning approach,” *Neural Comput & Applic*, vol. 28, no. 3, pp. 565–574, Mar. 2017.

[11] V. B. Semwal and G. C. Nandi, “Toward developing a computational model for bipedal push recovery–a brief,” *IEEE Sensors J.*, vol. 15, no. 4, pp. 2021–2022, Apr. 2015.

[12] V. B. Semwal, P. Chakraborty, and G. C. Nandi, “Less computationally intensive fuzzy logic (type-1)-based controller for humanoid push recovery,” *Robotics and Autonomous Systems*, vol. 63, pp. 122–135, Jan. 2015.

[13] J. Rebula, F. Canas, J. Pratt, and A. Goswami, “Learning capture points for humanoid push recovery,” in *Proc. 7th IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA, USA, Nov. 2007, pp. 65–72.

[14] A. Yasin, Q. Huang, Q. Xu, and W. Zhang, “Biped robot push detection and recovery,” in *Proc. IEEE International Conference on Information and Automation*, Shenyang, China, Jun. 2012, pp. 993–998.

[15] T. Kamioka, *et al*., “Dynamic gait transition between walking, running and hopping for push recovery,” in *Proc. IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Birmingham, Nov. 2017, pp. 1–8.

[16] M. Shafiee, G. Romualdi, S. Dafarra, F. J. A. Chavez, and D. Pucci, "Online DCM trajectory generation for push recovery of torque-controlled humanoid robots," in *Proc. IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, Toronto, ON, Canada, Oct. 2019, pp. 671–678.

[17] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, Oct. 2012, pp. 3411–3416.

[18] L. F. Wu and T. H. S. Li, "Fuzzy dynamic gait pattern generation for real-time push recovery control of a teen-sized humanoid robot," *IEEE Access*, vol. 8, pp. 36441–36453, 2020.

[19] A. H. Adiwahono, C. M. Chew, W. Huang, and V. H. Dau, "Humanoid robot push recovery through walking phase modification," in *Proc. IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, June 2010, pp. 569–574.

[20] Akash, S. Chandra, Abha, G. C. Nandi, "Modeling a bipedal humanoid robot using inverted pendulum towards push recovery," in *Proc. International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, India, Oct. 2012, pp. 1–6.

[21] B. J. Stephens and C. G. Atkeson, "Push Recovery by stepping for humanoid robots with Force Controlled joints," in *Proc. 10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, Dec. 2010, pp. 52–59.

[22] A. H. Adiwahono, C. M. Chew, W. Huang, and Y. Zheng, "Push recovery controller for bipedal robot walking," in *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore, Jul. 2009, pp. 162–167.

[23] S. Dafarra, F. Romano, and F. Nori, "Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub humanoid robot," in *Proc. IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Cancun, Mexico, Nov. 2016, pp. 152–157.

[24] E. C. Whitman, B. J. Stephens, and C. G. Atkeson, "Torso rotation for push recovery using a simple change of variables," in *Proc. 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Osaka, Japan, Nov. 2012, pp. 50–56.

[25] W. Shang, Z. Wu, Q. Liu, L. Duan, and C. Wang, "Foot placement estimator for quadruped push recovery," in *Proc. IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Suzhou, China, Jul. 2019, pp. 1530–1534.

[26] N. Dini and V. J. Majd, "An MPC-based two-dimensional push recovery of a quadruped robot in trotting gait using its reduced virtual model," *Mechanism and Machine Theory*, vol. 146, p. 103737, Apr. 2020.

[27] M. Khorram and S. A. A. Moosavian, "Push recovery of a quadruped robot on challenging terrains," *Robotica*, vol. 35, no. 8, pp. 1670–1689, Aug. 2017.

[28] J. W. Chung, I. H. Lee, B. K. Cho, and J. H. Oh, "Posture stabilization strategy for a trotting point-foot quadruped robot," *J Intell Robot Syst*, vol. 72, no. 3–4, pp. 325–341, Dec. 2013.

[29] R. C. Luo and C. W. Huang, "A push-recovery method for walking biped robot based on 3-D flywheel model," in *Proc. IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Yokohama, Nov. 2015, pp. 002685–002690.

[30] S. M. Kasaei, N. Lau, A. Pereira, and E. Shahri, "A reliable model-based walking engine with push recovery capability," in *Proc. IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Coimbra, Portugal, Apr. 2017, pp. 122–127.

[31] R. C. Luo, Jun Sheng, C. C. Chen, P. H. Chang, and C. I. Lin, "Biped robot push and recovery using flywheel model based walking perturbation counteraction," in *Proc. 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Atlanta, GA, Oct. 2013, pp. 50–55.

[32] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proc. 6th IEEE-RAS International Conference on Humanoid Robots*, University of Genova, Genova, Italy, Dec. 2006, pp. 200–207.

[33] Y. Wei, B. Gang, and W. Zuwen, "Balance recovery for humanoid robot in the presence of unknown external push," in *Proc. International Conference on Mechatronics and Automation*, Changchun, China, Aug. 2009, pp. 1928–1933.

[34] A. Hosseinmemar, J. Baltes, J. Anderson, M. C. Lau, C. F. Lun, and Z. Wang, "Closed-loop push recovery for inexpensive humanoid robots," *Appl Intell*, vol. 49, no. 11, pp. 3801–3814, Nov. 2019.

[35] K. Guan, K. Yamamoto, and Y. Nakamura, "Push recovery by angular momentum control during 3D bipedal walking based on virtual-mass-ellipsoid inverted pendulum model," in *Proc. IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, Toronto, ON, Canada, Oct. 2019, pp. 120–125.

[36] J. Lack, "Integrating the effects of angular momentum and changing center of mass height in bipedal locomotion planning," in *Proc. IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Seoul, South Korea, Nov. 2015, pp. 651–656.

[37] Y. Kojio, *et al*., "Footstep modification including step time and angular momentum under disturbances on sparse footholds," *IEEE Robot. Autom. Lett*., vol. 5, no. 3, pp. 4907–4914, Jul. 2020.

[38] Y. Kojio, *et al*., "Unified balance control for biped robots including modification of footsteps with angular momentum and falling detection based on capturability," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019, pp. 497–504.

[39] C. H. Chang, H. P. Huang, H. K. Hsu, and C. A. Cheng, "Humanoid robot push-recovery strategy based on CMP criterion and angular momentum regulation," in *Proc. IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, South Korea, Jul. 2015, pp. 761–766.

[40] S. K. Yun and A. Goswami, "Momentum-based reactive stepping controller on level and non-level ground for humanoid robot push recovery," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 3943–3950.

[41] R. J. Griffin, A. Leonessa, and A. Asbeck, "Disturbance compensation and step optimization for push recovery," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 5385–5390.

[42] S. H. Hyon, R. Osu, and Y. Otaka, "Integration of multi-level postural balancing on humanoid robots," in *Proc. IEEE International Conference on Robotics and Automation*, Kobe, May 2009, pp. 1549–1556.

[43] J. Zhao, S. Schutz, and K. Berns, "Biologically motivated push recovery strategies for a 3D bipedal robot walking in complex environments," in *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenzhen, China, Dec. 2013, pp. 1258–1263.

[44] B. Stephens, "Humanoid push recovery," in *Proc. 7th IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA, USA, Nov. 2007, pp. 589–595.

[45] P. Ghassemi, M. T. Masouleh, and A. Kalhor, "Push recovery for NAO humanoid robot," in *Proc. Second RSI/ISM International Conference on Robotics and Mechatronics (ICROM)*, Tehran, Iran, Oct. 2014, pp. 035–040.

[46] M. Shafiee-Ashtiani, A. Yousefi-Koma, M. Shariat-Panahi, and M. Khadiv, "Push recovery of a humanoid robot based on model predictive control and capture point," in *Proc. 4th International Conference on Robotics and Mechatronics (ICROM)*, Tehran, Iran, Oct. 2016, pp. 433–438.

[47] Y. H. Lee, *et al*., "Whole-Body control and angular momentum regulation using torque sensors for quadrupedal robots," *J Intell Robot Syst*, vol. 102, no. 3, p. 66, Jul. 2021.

[48] S. H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Auton Robot*, vol. 33, no. 4, pp. 399–414, Nov. 2012.

[49] R. Schuller, G. Mesesan, J. Englsberger, J. Lee, and C. Ott, "Online centroidal angular momentum reference generation and motion optimization for humanoid push recovery," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5689–5696, Jul. 2021.

[50] R. C. Luo, W. C. Hung, and R. Chatila, "Mimicking human push-recovery strategy based on five-mass with angular momentum model," in *Proc. IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, Oct. 2016, pp. 716–721.

[51] O. K. Adak, B. Bahceci and K. Erbatur, "Modeling of a quadruped robot with spine joints and full-dynamics simulation environment construction," *International Journal of Intelligent Robotics and Applications*, (in press).

Beste Bah çeci is a Ph.D. candidate in Mechatronics Engineering at Sabanci University, Istanbul, Turkey. She received a B.S. and M. Sc. degree in Mechatronics Engineering from Sabanci University, Istanbul, Turkey, in 2011 and 2013. In 2011, she joined the Robotics Research Laboratory at Sabanci University Istanbul, Turkey, and studied on control of industrial robots. In 2013 she joined the Legged Robotics Research Laboratory at Sabanci University, Istanbul, Turkey. Her primary research interest is legged robot locomotion. She is the author of more than five papers that have been published in international conference proceedings and journals.

Omer Kemal Adak is a Ph.D. candidate in Mechatronics Engineering at Sabanci University, Istanbul, Turkey. He received the B.S. (honors) degree in Mechanical Engineering from Yildiz Technical University, Istanbul, Turkey, the M.Sc. degree in Mechatronics Engineering from Sabanci University, Istanbul, Turkey, in 2011 and 2013. In 2011, he joined the Legged Robotics Research Laboratory at Sabanci University, Istanbul, Turkey. His primary research interest is legged robot locomotion. He is the author of more than ten papers that have been published in international conference proceedings and journals. He is the reviewer of several IEEE conference proceedings.

Kemalettin Erbatur received the B.S. (honors) degree in Electronic Engineering and Mathematics from Bogazici University, Istanbul, Turkey, the M.Sc. degree in Control Systems from Imperial College, London, U. K, and the Ph.D. degree from Bogazici University, in 1992, 1993, and 2000, respectively. In 1993, he joined the Robotics and Automation Group, TUBITAK Marmara Research Centre, Gebze-Kocaeli, Turkey. During 1997–1998, he was the Leader of this group. In the period 1998 2000, he was an Expert Researcher in the Electronic Systems Group. Between 2000 and 2002, He was a Visiting Researcher in the New Industry Research Organization, Kobe, Japan, followed by a visiting faculty position at the Electrical and Computer Engineering Department, Yokohama National University. In 2003, he was a collaborator of the International Rescue System Institute, Kobe, Japan, as a faculty of Sabanci University. Since 2000, he has been a faculty member in the Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey. His primary research interest is in intelligent motion control and legged robot locomotion. He is the author of more than 60 papers that have been published in international conference proceedings and journals.