2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Body Velocity Estimation in a Leg–Wheel Transformable Robot without A Priori Knowledge of Leg–Wheel Ground Contacts

Pei-Chun Huang, I-Chia Chang, Wei-Shun Yu and Pei-Chun Lin

***Abstract*— The state estimation of legged robots often relies on ground contact detection. However, due to complex mechanisms and other factors, ground contact detection can be challenging to obtain in certain situations. This paper presents a velocity estimation method that combines inertia measurement unit (IMU) and encoders, allowing estimation without using the ground contact state as the a priori. In this paper, the initial estimate derived from IMU integration is refined. Following the computation of velocity and ground contact state probabilities using encoder data, these probabilities are employed to modify particle weights within the particle filter framework. Subsequent resampling ensures that the contact status converges toward the correct result. This paper tests the algorithm through simulations and validates the method with physical experiments, showcasing the feasibility of concurrent ground contact state and velocity estimation.**

I. Introduction

Body state is critical information for mobile robots in localization or dynamic maneuvers. Typically, wheeled robots operate in flat terrain environments, so the estimation of body states is relatively easy using wheel encoders, inertia measurement units (IMU), and global positioning systems. In contrast, the estimation of the body state of the legged robots [1] is a challenge not only because the intermittent contact status of all legs on uneven terrain is hard to obtain but also because the kinematics and dynamics of the legged robots are much more complex than those of the wheeled robots. In addition, leg–wheel hybrid robots are designed to maximize the robot's mobility and efficiency on various terrains. Wheeled-legged robots can be further divided into those that transform toe points into wheels [2] or switch from wheels to legs through deformation [3, 4]. The primary distinction between these two types of robots lies in the wheel-diameter-to-leg-length ratio. When the leg length is relatively longer than the wheel diameter, the dynamics are closer to those of legged robots. In contrast, they resemble wheeled robots more closely when the wheel diameter is somewhat larger. Regardless of the form, body state estimation is even more challenging owing to the complex ground contact condition. In such cases, maximizing the utilization of moments when stable contact between the robot and the ground is established becomes crucial.

State estimation for legged robots has been reported. Bloesch et al. introduced a state estimation model for legged robots based on known ground contact and contact point information [5]. Building upon this foundation, subsequent studies such as [6, 7] addressed ground instability or sliding issues by incorporating additional sensors or adding velocity offset terms. You et al. introduced the concept of an instantaneous contact point for wheeled-legged hybrid robots [8], where this model adjusted the ground contact weights based on force estimates. In the models described above, precise dynamic models are required to estimate ground contact forces, or force sensors need to be installed at toe points. This approach typically relies on knowing the relative positions of contact points concerning the robot's body and is more suitable for application in wheeled-legged robots with smaller wheel diameters.

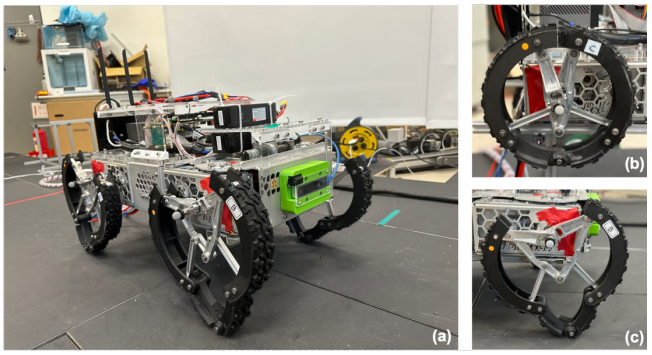

Fig. 1. Photo of the leg-wheel transformable robot (a) and its leg-wheel in wheeled mode (b) and in legged mode (c)

Contact state and contact point estimations are commonly achieved by integrating current sensors with dynamics. Manuelli and Tedrake used a particle filter and a dynamic model to estimate multiple collision points [9]. Bledt et al. proposed a ground contact probability model incorporating force, phase, and foot clearance [10]. Based on this model, Hosseini et al. developed a wheel-legged robot odometry method using a Kalman filter for estimation [11]. These approaches still rely on precise dynamic models, and obtaining accurate ones can be relatively challenging in complex closed-loop linkage configurations. In addition, contact sensors [12] are typically used in scenarios such as clamping fixtures, where they can provide direct measurement information. However, they can be challenging to install on mechanisms that require continuous rotation. In such cases, methods have proposed models based on velocity estimation of touch points, which exhibit better robustness than dynamic models due to kinematic models [13, 14].

In the current research, state estimation typically relies on known contact points, whereas contact point estimation depends on available state information or precise dynamic models. However, when leg designs are complex and both the

This work is supported by the National Science and Technology Council, Taiwan, under contract: MOST 110-2221-E-002-111-MY3

The authors are with the Department of Mechanical Engineering, National Taiwan University (NTU), No.1 Roosevelt Rd. Sec.4, Taipei 106, Taiwan. (Corresponding email: peichunlin@ntu.edu.tw).

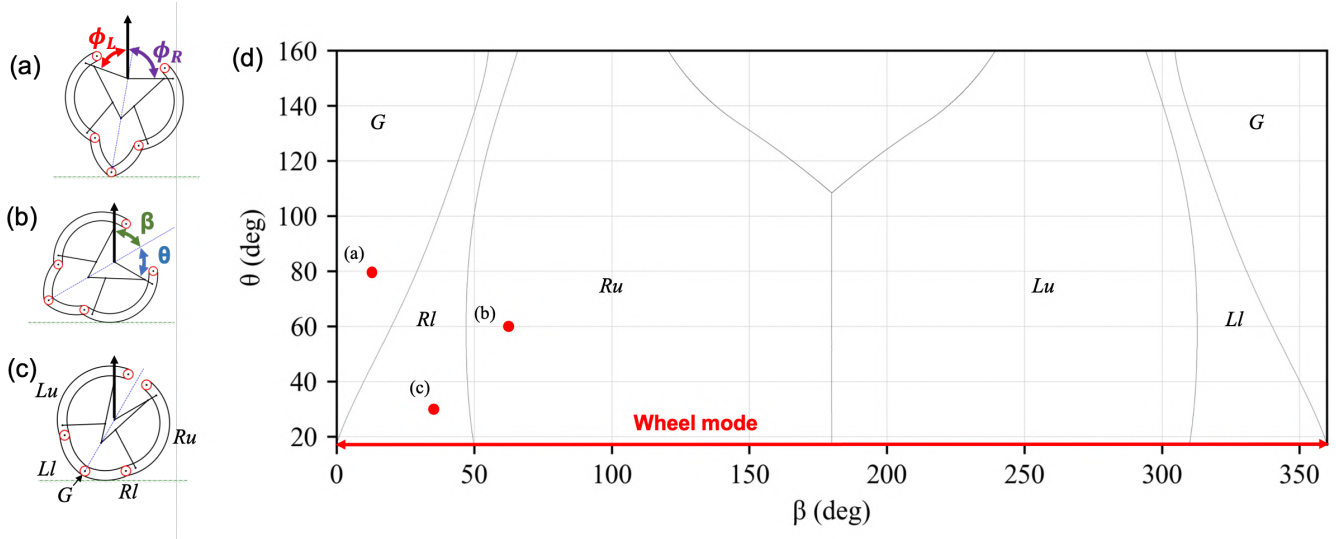

Fig. 2. The leg-wheel of the robot: (a) Contact the ground using its toe point. The symbols (ϕ_R, ϕ_L) represent the motor inputs. (b) Contacting the ground using its upper rim. The symbols (θ, β) determine the leg-wheel extension and orientation, respectively. (c) Contacting the ground using its lower rim. The symbols Lu, Ru, Ll, Rl, and G represent upper left rim, upper right rim, lower left rim, lower right rim, and toe point, respectively. (d) Mapping of (θ, β) to the different contact points of the leg-wheel on even ground. The red dots with labels of (a)-(c) represent the leg-wheel configurations shown in subfigures (a)-(c).

contact states and contact points are unknown, calculating ground reaction forces through dynamics becomes quite challenging, especially for wheeled-legged hybrid robots, where the wheel-diameter-to-leg-length ratio is relatively large. The leg–wheel transformable robot in our lab has a wheel diameter-to-leg length ratio exceeding one-third, allowing for a broader range of potential contact points. Thus, analyzing dynamics when it is unclear whether contact has occurred can be computationally expensive, and dynamic modeling requires greater precision than kinematics. Even exploring the contact forces solely at the toe points is highly complex [15]. Furthermore, relying on anticipated contact states based on gait cycles can become inaccurate in practical scenarios due to factors such as overturning.

In this work, we aim to develop a velocity estimation strategy that can be seamlessly applied across all three modes without knowing the ground contact status between the leg–wheel and the ground. This is opposed to the common strategy, which needs the ground contact information as the priori. This method utilizes an IMU as the prediction model and encoders as the observation model. Under the assumption of non-slipping conditions, it combines both a particle filter and hidden ground contact states to maximize the utilization of ground contact moments for correcting integration errors. Compared with estimation methods employed by other legged and wheeled robots, this approach can estimate velocity using a kinematic model in scenarios with unknown ground contact states. Simultaneously, it can make a certain level of judgment about ground contact status and the estimation of contact points. In contrast to commonly used dynamic models, kinematic models offer higher robustness and require fewer computations, making them more suitable for complex mechanical systems.

The remainder of this paper is organized as follows. Section II provides definitions of state and measurement, along with an explanation of the proposed methodology. Section III demonstrates the setup and results of the simulation. Section IV offers an overview of an experiment validation conducted on our platform. In Section V, we present the concluding remarks.

II. Problem Statement

A. The Robot

Figure 1 shows a photo of the leg–wheel transformable robot utilized in this work. As shown in Figure 1(b)(c) or Figure 2(a)–(c), each leg–wheel is formed by an 11-linkage mechanism moving in the sagittal plane [16]. The leg–wheel is driven by two motors (ϕ_R, ϕ_L), and the motor torques directly drive the leg–wheel to rotate (β) or extend (θ) through the mechanism. The leg–wheel has a leg length 3.4 times the wheel radius, and it is capable of fast transformation between its wheeled and legged modes. The details of its dynamic model and control can be found in [15] [17]. In addition, the front wheels can turn to enable the robot to turn in wheeled mode. The robot utilizes a center pattern generator to control the leg wheels [18], and it can perform a smooth leg–wheel transformation [19]. As for the sensors, the driving motors are equipped with encoders (HT-04, Haitai), and the robot also has an IMU (3DM-CX5-AHRS, MicroStrain) on board.

Owing to the unique leg–wheel design, the robot can operate in three modes: wheeled mode for even terrain, legged mode for extremely rough terrain, and leg–wheel hybrid mode for the terrain in between. During motion, the leg–wheel can contact the ground in five scenarios, as shown in Figure 2(c): upper left rim (**Lu**), upper right rim (**Ru**), lower left rim (**Ll**), lower right rim (**Rl**), and toe point (**G**). Figure 2(d) plots the ground contact conditions of the leg–wheel in different configurations (θ, β) when the robot is horizontally placed on the even ground. When the robot is in wheeled mode (i.e., $\theta = 17°$ and β continuously increases or decreases), the five contact scenarios occur periodically. When the robot is in legged mode, only the toe contact is utilized, that is, θ and β are confined to specific ranges. When the robot is in hybrid

mode, the contact scenarios vary depending on the leg–wheel height and motion setup. When the incline exists, the contact points shift along the β-axis accordingly.

B. State Definition

Figure 3 shows the definitions of the symbols utilized in this work. Index i, j, and k represent the following: Particle index i ranging from 1 to N, leg index j ranging from 1 to 4, and time index k starting from t_0. The robot's state at moment k (s_k) includes the following components in the body coordinate frame: velocity ($\mathbf{v_k}$), gyroscope bias ($\mathbf{b_k^a}$, $\mathbf{b_k^g}$), and a vector representing contact points ($\mathbf{r_k^j}$) of leg j. The system's inputs consist of body angular velocity ($\boldsymbol{\omega_k}$), acceleration ($\mathbf{a_k}$), motor angles, and angular velocities. Note that the motor input angles (φ_R, φ_L) and angular velocities ($\dot{\varphi_R}, \dot{\varphi_L}$) can be converted to $(\theta_k^j, \beta_k^j, \dot{\theta_k^j}, \dot{\beta_k^j})$.

$$s_k := [\mathbf{v_k}, \mathbf{b_k^a}, \mathbf{b_k^g}, \dots \mathbf{r_k^j} \dots]$$

$$u_k := [\boldsymbol{\omega_k}, \mathbf{a_k}, \dots \theta_k^j, \beta_k^j, \dot{\theta_k^j}, \dot{\beta_k^j} \dots]$$

The motion of the robot's body to the points of contact with the ground can be represented in the following equation:

$$\begin{aligned}\mathbf{v_k^{c_j}} = \mathbf{v_k} + \boldsymbol{\omega_k} \times \left(\mathbf{p_j} + \mathbf{lkin_j}\left(\theta_k^j, \beta_k^j\right) + \mathbf{r_k^j}\right) \\ + \mathbf{lkin_j}\left(\theta_k^j, \beta_k^j, \dot{\theta_k^j}, \dot{\beta_k^j}\right) + \mathbf{lkin_j^\omega}\left(\theta_k^j, \beta_k^j, \dot{\theta_k^j}, \dot{\beta_k^j}\right) \times \mathbf{r_k^j}\end{aligned} \quad (1)$$

where $\mathbf{p_j}$ is the vector from the center of mass to the motor axis of the j'th leg in the body coordinate frame, $\boldsymbol{lkin_j}$ represents the vector from the motor axis to the center of the contacting arc, $\boldsymbol{lkin_j^\omega}$ represents the angular velocity of the contacting arc caused by linkage kinematic, and $\mathbf{r_k^j}$ is the vector from the center of the current contacting arc to the contact point in the body coordinate frame, which can be mapped on Figure 2(d). The state prediction equations are described from (2) to (5).

$$\boldsymbol{\Delta}^{\mathbf{k+1}}_{\mathbf{k}}\mathbf{R} = Exp\left(\Delta t\left(\boldsymbol{\omega_k} - \mathbf{b_k^g} - \boldsymbol{\sigma_k^g}\right)\right) \quad (2)$$

$$\mathbf{v_{k+1}} = \boldsymbol{\Delta}^{\mathbf{k+1}}_{\mathbf{k}}\mathbf{R}\left(\mathbf{v_k} + \left(\mathbf{a_k} - \mathbf{b_k^a} - \boldsymbol{\sigma_k^a}\right)\Delta t\right) - \boldsymbol{\sigma_k^v} \quad (3)$$

$$\begin{cases}\mathbf{b_{k+1}^a} = \mathbf{b_k^a} - \Delta t\boldsymbol{\sigma_k^{b_a}} \\ \mathbf{b_{k+1}^g} = \mathbf{b_k^g} - \Delta t\boldsymbol{\sigma_k^{b_g}}\end{cases} \quad (4)$$

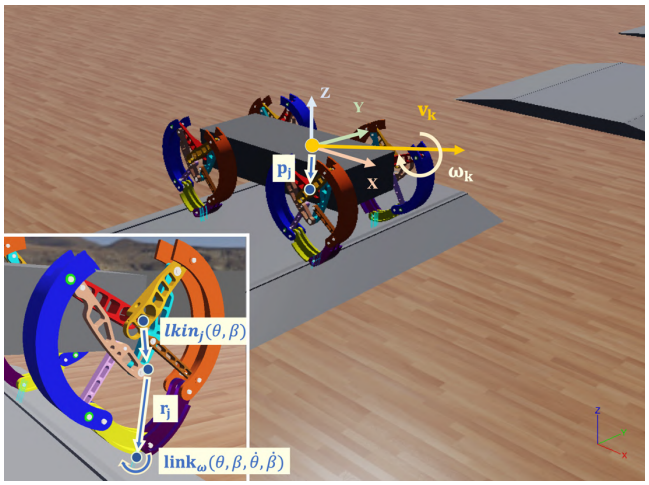

Fig. 3. Definition of the state and measurements, all in body frame.

Algorithm 1

```
for s_k^i, c_k^i in X_k do
    š_{k+1}^i = prediction(s_k^i, u_k)
    c_k^i = random_flip(c_k^i, p_flip)
    for c^j in c_k^i do
        if c^j ∈ contact then
            W_j = ε
        else
            if abs(v_{k+1}^{c_j} − v_{k+1}) < threshold then
                define c^j contact
            else
                W_j = (v_{k+1}^{c_j} − v_{k+1})^2
            endif
        endif
    end for
    w_{k+1}^i = p(m_{k+1} | s_{k+1}^i) w_k^i
end for
s_{k+1} = Σ w_{k+1}^i š_{k+1}^i
if 1./(w_{k+1}^i · w_{k+1}^i) < N_th then
    X_{k+1} = Important − Resample(X_k)
endif
```

where $\boldsymbol{\Delta}^{\mathbf{k+1}}_{\mathbf{k}}\mathbf{R}$ represents the rotation matrix from t=k to t=k+1, $\boldsymbol{\sigma_k^a}$ represents the Gaussian noise of acceleration, and $\boldsymbol{\sigma_k^v}$ is the Gaussian noise of the prediction model. $\boldsymbol{\sigma_k^{b_a}}$ and $\boldsymbol{\sigma_k^{b_g}}$ is the white noise to simulate Brownian motions of bias terms.

$$\mathbf{r_{k+1}^j} = map\left(\overline{\theta_{k+1}^J}, \quad \overline{\beta_{k+1}^J}\right) \quad (5)$$

The prediction of contact vector $\mathbf{r_{k+1}^j}$ is first mapped onto Figure 2(d) by $(\overline{\theta_{k+1}^J}, \overline{\beta_{k+1}^J})$ and can be separated into two cases. If currently at the contact state, $\overline{\theta_{k+1}^J}$ is updated by integrating $\dot{\theta_k^J}$, where $\sigma_k^{\dot{\theta}J}$ is the noise of measured input $\dot{\theta_k^J}$. In addition, $\overline{\beta_{k+1}^J}$ is updated by $\dot{\beta_k^J}$ and $\sigma_k^{\dot{\beta}J}$, but also consider pitch rate term (y-axis) from $(\boldsymbol{\omega_k} - \mathbf{b_k^g} - \boldsymbol{\sigma_k^g})$ and the noise σ_k^n from terrain.

$$\begin{cases}\overline{\theta_{k+1}^J} = \overline{\theta_k^J} + \Delta t\left(\dot{\theta_k^J} - \sigma_k^{\dot{\theta}J}\right) \\ \overline{\beta_{k+1}^J} = \overline{\beta_k^J} + \Delta t\left(\dot{\beta_k^J} - \sigma_k^{\dot{\beta}J} + \left(\boldsymbol{\omega_k} - \mathbf{b_k^g} - \boldsymbol{\sigma_k^g}\right)_y - \sigma_k^n\right)\end{cases} \quad if\ c^j \in contact \quad (5.1)$$

If currently at a non-contact state, $\overline{\theta_{k+1}^J}$ and $\overline{\beta_{k+1}^J}$ are updated by measured θ_{k+1}^j and β_{k+1}^j.

$$\begin{cases}\overline{\theta_{k+1}^J} = \theta_{k+1}^j \\ \overline{\beta_{k+1}^J} = \beta_{k+1}^j\end{cases} \quad else \quad (5.2)$$

C. Measurement Definition

The measured state is the motion velocity of the ground contact point ($\mathbf{v_k^{c_j}}$). Assuming that the contact point remains stationary and that the probability of the predicted state follows a Gaussian distribution with a tiny covariance matrix, the velocity of the contact point calculated based on the state can be obtained using (1).

$$m_k := \left[\mathbf{v_k^{c_1}}, \dots \mathbf{v_k^{c_j}} \dots, \mathbf{v_k^{c_n}}\right]$$

D. Particle Filter

Within the filter, a hidden state $\mathbf{c_k^i}$ for particle i is introduced to indicate whether the four legs of the robot are in

contact with the ground or not, where c^j is the segment of $\mathbf{c_k^i}$ for leg j. The impact of ground contact or non-contact is reflected in the diagonal covariance matrix $\mathbf{W} \in \mathbb{R}^{12\times12}$, which is combined with $\mathbf{W_j} \in \mathbb{R}^{3\times3}$. In the case of ground contact, the covariance matrix values are defined as a minimal number matrix $\boldsymbol{\varepsilon}$, as described by the measurement definition. In the case of non-contact, the covariance matrix is set to the square of the robot's leg motion relative to the body $\left(\mathbf{v_{k+1}^{c_j}} - \mathbf{v_{k+1}}\right)$. The state is directly determined as in contact when the leg motion velocity is below a threshold. As a result, when correctly predicting ground contact, particle weights receive a significant boost due to compliance with the narrow Gaussian function. Conversely, they receive relatively larger gains in the non-contact state. The reason for setting the state as being in contact when the foot motion velocity is close to zero is that at low speeds, and when the foot is not moving, whether in contact or not, the calculated contact velocity will be close to zero. In this scenario, the relationship between the contact velocity and the state is unclear.

In each iteration, the weights from the previous moment (w_k^i) are updated based on (6). During the prediction phase, the ground contact status for each leg within each particle changes by a random flipping function. Eventually, the correct landing state will have higher weights throughout the filtering process. Therefore, particles representing the ground contact state with higher weights are more likely to be sampled during the resampling process.

$$w_{k+1}^i = p\left(m_{k+1}|s_{k+1}^i\right)w_k^i \tag{6}$$

Algorithm I shows the pseudocode of the algorithm.

III. Simulation

A. Simulation Setup

Webots® with sampling at 200Hz is used as the simulation environment to evaluate the proposed estimation strategy. Sensor data include the states from an IMU and motor encoders, and data on ground contact and body velocity are collected for validation. Offsets and random noise are added to the sensory data to simulate the empirical conditions.

TABLE I. Accuracy of Contact Estimation

Contact state accuracy						
Idx	Leg–wheel contact scenarios	Velocity command (m/s)	Slopes of the trapezoidal obstacles			
			0°	5°	10°	20°
1	Upper ring	0.1	0.88	0.87	0.86	0.79
2	Upper ring	0.1	0.90	0.85	0.83	0.82
3	Upper ring	0.3	0.88	0.84	0.83	0.82
4	Upper ring	0.6	0.79	0.76	0.77	0.79
5	Upper ring	1.0	0.75	0.80	0.77	0.77
6	Lower ring	0.05	0.77	0.75	0.75	0.75
7	Lower ring	0.1	0.75	0.75	0.79	0.76
8	Fixed Point	0.3	0.96	0.92	0.88	0.75
9	Fixed Point	0.05	0.82	0.78	0.78	0.78
10	Fixed Point	0.1	0.82	0.77	0.79	0.79

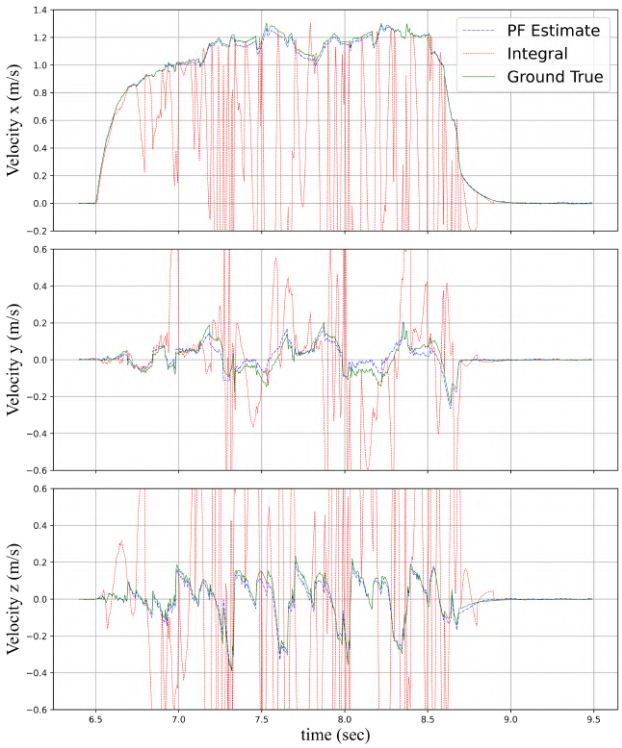

Fig. 4. Estimated velocity of the robot in simulation. The velocity derived by direct integration of IMU data is utilized as the comparison baseline.

In the proposed model, the body's orientation and terrain are not included in the state estimation. Consequently, compensation for terrain irregularities and body orientation is solely based on Gaussian noise within the model. To test the robustness of this method, a wide range of scenarios are simulated, including four ramp terrains with different shapes and 10 motion modes with different speeds and contact points (i.e., can be roughly categorized into rim-based and toe-based ground contacts), as shown in Table I. Therefore, there are 40 simulation runs in total.

B. Velocity Estimation

Table I lists the mean absolute error (MAE) of the velocity estimation of the robot operating in different scenarios. The table reveals that the slopes and obstacle heights have a minimal impact on estimation performance. In contrast, the ground contact mode and robot speed have a more significant influence, where the errors are relatively higher when the speeds are high, even though the performance still maintains a certain accuracy. In contrast, the estimation using merely IMU would accumulate errors rapidly, even in short durations, as shown in Figure 4. The estimation performance of the robot using three ground contact scenarios also varies slightly.

C. Contact State Estimation

In addition to estimating the robot velocity, the ground contact condition is evaluated. Binary judgment is utilized, categorizing the conditions into contact and non-contact. Based on the weighted sum of particle weights, a threshold of middle point 0.5 is used. The contact estimation accuracy is

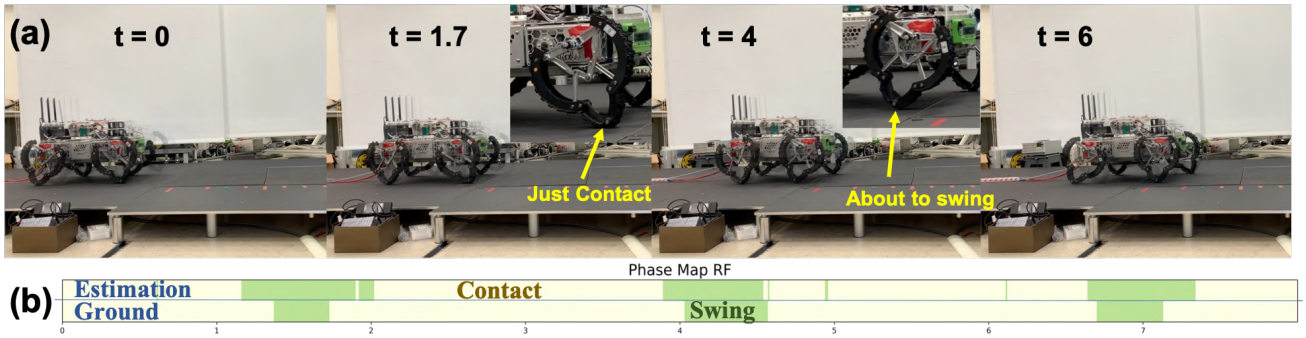

Fig. 5. The walking experiment #B of the robot as the example. (a) Snapshots of the motion sequence, with particular interest in the contact status of the leg–wheels. (b) Ground contact estimation (upper row) and the ground truth (lower row). The light green and yellow bars represent the swing phase and the contact phase, respectively.

TABLE II. ACCURACY OF CONTACT ESTIMATION

Contact state accuracy						
Idx	Leg–wheel contact scenarios	Velocity command (m/s)	Slopes of the trapezoidal obstacles			
			0°	5°	10°	20°
1	Upper ring	0.1	0.88	0.87	0.86	0.79
2	Upper ring	0.1	0.90	0.85	0.83	0.82
3	Upper ring	0.3	0.88	0.84	0.83	0.82
4	Upper ring	0.6	0.79	0.76	0.77	0.79
5	Upper ring	1.0	0.75	0.80	0.77	0.77
6	Lower ring	0.05	0.77	0.75	0.75	0.75
7	Lower ring	0.1	0.75	0.75	0.79	0.76
8	Fixed Point	0.3	0.96	0.92	0.88	0.75
9	Fixed Point	0.05	0.82	0.78	0.78	0.78
10	Fixed Point	0.1	0.82	0.77	0.79	0.79

calculated separately for each leg–wheel and then averaged, and Table II summarizes the results.

The results of ground contact detection are correlated with the accuracy of speed estimation. As evidenced by #8, as the height and slope of obstacles increase, there is a simultaneous decrease in ground contact detection, leading to a degradation in the accuracy of speed estimation.

IV. EXPERIMENT RESULTS

The proposed velocity estimation strategy is experimentally evaluated using the leg–wheel transformable robot in our lab. The ground truth dataset was obtained using a motion analysis system (Vicon) tracking at 500Hz. The velocity is differentiated from position data, transformed into the body coordinate system, and then filtered to 100Hz using a mean filter. The raw data from the IMU was collected at 500Hz, while the encoders were sampled at 400Hz. Both sensor data are downsampled to 100Hz using median filtering to form the input data for the algorithm. The estimated velocity in the control group is obtained by directly integrating the IMU's raw data at intervals of 0.1 seconds.

The experimental runs of the robot with seven different settings are evaluated, including one optimized walking (marked as #A), one non-optimized walking (marked as #B), and the first five trajectories utilized in simulations, as shown in Tables I and II (marked as #C-G). Figure 5 shows an example run of the robot. Note that both the simulated and experimental trajectories were pre-optimized using reduced-order models, and no real-time closed-loop control of the robot was involved. Therefore, although the cost function of the optimization considered factors such as walking stability and energy consumption, the empirical robot could still tip over during movement owing to the discrepancy between the model and the robot. Even though the robot's gait is not perfect, which yields unstable locomotion, the velocity estimation of the robot can still be functional well because the proposed strategy does not rely on ground contact conditions. This is the best advantage of the proposed strategy.

A. Velocity Estimation

Figure 6 shows the time sequence of the estimated velocity of the robot in setting #E, where the robot moves at a speed of 0.3 m/s. It can be observed that at certain moments, the filter successfully reduces the errors associated with the integration model. Figure 7 shows the performance of the robot in all runs. It is worth noting that the error sources differ from the simulation. In the simulation, the primary source of error was along the x-axis of the body. In contrast, in the empirical experiments, there were instances where the errors along the y-axis were more significant, such as in trajectory #B. This difference arises because the legs are not rigid bodies in the empirical experiments. During body rollovers or rapid accelerations/decelerations, the legs experience forces along the y-axis and undergo deformations that result in displacements along the y-axis (as shown in Figure 6, there is

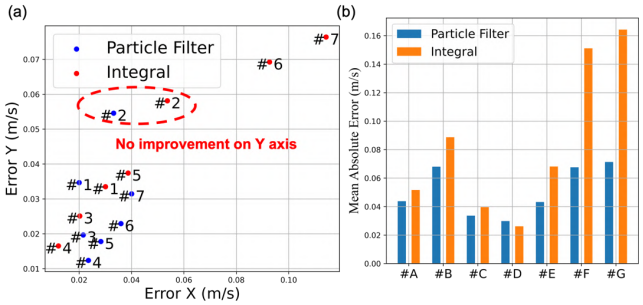

Fig. 7. Mean absolute error of the estimated velocity of the robot: (a) Error distribution plot; the data point close to the original represent better performance. (b) MAE of the robot in different runs.

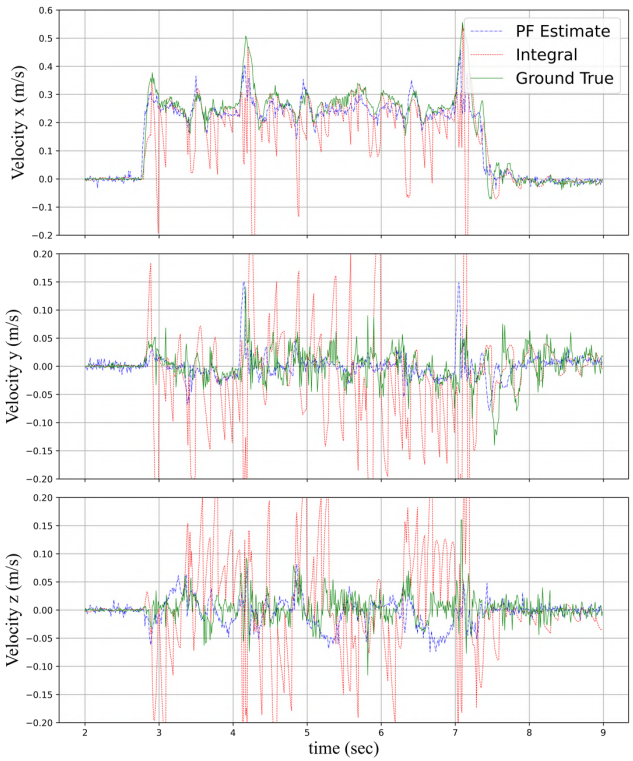

Fig. 6. Estimated velocity of the robot in experiment #E. The velocity derived by direct integration of IMU data is utilized as the comparison baseline.

still significant motion along the y-axis after stopping). These effects are not explicitly accounted for in the model used in the simulation.

B. Contact State Estimation

The contact verification data in the experiments were marked from the videos. The comparison between estimation and verification is shown in Figure 5. At t=2, the right front leg has just made contact, and at t=4, the right front leg has just lifted off. Evidently, the estimation of ground contact still maintains a certain degree of relevance in physical experiments.

V. Conclusion

This paper proposes a velocity estimation strategy using only IMU and encoder data without utilizing the contact information as a priori. The strategy is particularly useful for robots with complex ground contact status, such as leg–wheel hybrid robots. In the simulation, we tested MAE under different conditions, averaging 0.025m/s with a standard deviation of 0.008m/s. In the robot experiments, we obtained an error of 0.05m/s with a standard deviation of 0.016m/s. The similarity in standard deviations indicates that there is no significant difference in errors between the datasets. Moreover, the relatively consistent errors across different conditions imply that the advantages of the estimator become more pronounced at higher speeds. This observation is also supported by Figure 7(b), where in the higher-speed E-G data, the differences between the estimator and simple integration become more noticeable.

In the future, this method could be enhanced by incorporating sensors such as vision and LiDAR to compensate for relatively inaccurate estimations at low speeds. Regarding the impact of leg deformation, the predictive part can operate independently and only update when observational data allows. This way, integration can be performed at a higher frequency, potentially reducing errors.

References

[1] M. Hutter *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2016: IEEE, pp. 38-44.

[2] M. Bjelonic *et al.*, "Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters,* vol. 4, no. 2, pp. 2116-2123, 2019.

[3] Y. Shi, M. Zhang, M. Li, and X. Zhang, "Design and Analysis of a Wheel−Leg Hybrid Robot with Passive Transformable Wheels," *Symmetry,* vol. 15, no. 4, p. 800, 2023. [Online]. Available: https://www.mdpi.com/2073-8994/15/4/800.

[4] K. Tadakuma *et al.*, "Mechanical design of the wheel-leg hybrid mobile robot to realize a large wheel diameter," in *2010 IEEE/RSJ international conference on intelligent robots and systems*, 2010: IEEE, pp. 3358-3365.

[5] M. Bloesch *et al.*, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics,* vol. 17, pp. 17-24, 2013.

[6] D. Wisth, M. Camurri, and M. Fallon, "Preintegrated velocity bias estimation to overcome contact nonlinearities in legged robot odometry," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020: IEEE, pp. 392-398.

[7] J. H. Kim *et al.*, "Legged Robot State Estimation With Dynamic Contact Event Information," (in English), *Ieee Robotics and Automation Letters,* vol. 6, no. 4, pp. 6733-6740, Oct 2021, doi: 10.1109/Lra.2021.3093876.

[8] Y. You *et al.*, "State estimation for hybrid wheeled-legged robots performing mobile manipulation tasks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021: IEEE, pp. 3019-3025.

[9] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016: IEEE, pp. 5062-5069.

[10] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 4399-4406.

[11] M. Hosseini, D. Rodriguez, and S. Behnke, "State Estimation for Hybrid Locomotion of Driving-Stepping Quadrupeds," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, 2022: IEEE, pp. 103-110.

[12] R. S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions Toward Effective Utilization of Tactile Skin: A Review," (in English), *Ieee Sensors Journal,* vol. 13, no. 11, pp. 4121-4138, Nov 2013, doi: 10.1109/Jsen.2013.2279056.

[13] V. Barasuol, G. Fink, M. Focchi, D. Caldwell, and C. Semini, "On the detection and localization of shin collisions and reactive actions in quadruped robots," in *International Conference on Climbing and Walking Robots*, 2019, vol. 49, p. 51.

[14] S. Wang, A. Bhatia, M. T. Mason, and A. M. Johnson, "Contact localization using velocity constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020: IEEE, pp. 7351-7358.

[15] Y.-J. Liu and P.-C. Lin, "Development of a dynamic model of the 11-linkage and closed-chain leg-wheel module," presented at the 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Sapporo, Japan, 2022. [Online]. Available: https://doi.org/10.1109/AIM52237.2022.9863328.

[16] H.-Y. Chen, T.-H. Wang, K.-C. Ho, C.-Y. Ko, P.-C. Lin, and P.-C. Lin, "Development of a novel leg-wheel module with fast transformation and leaping capability," *Mechanism and Machine Theory,* vol. 163, p. 104348, 2021.

[17] Y.-C. Zhuang, Y.-J. Liu, W.-S. Yu, and P.-C. Lin, "A Hybrid Impedance and Admittance Control Strategy for a Shape-

Transformable Leg-Wheel," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2023: IEEE, pp. 299-304.
[18] L.-J. Chen and P.-C. Lin, "Gait Pattern Stabilization using Central Pattern Generator with Foothold Force Optimization for Quadruped Robots," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2022: IEEE, pp. 1658-1663.
[19] H.-Y. Wang, L.-J. Chen, W.-S. Yu, and P.-C. Lin, "A Wheel to Leg Transformation Strategy in a Leg-Wheel Transformable Robot," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2023: IEEE, pp. 293-298.