Real-time Terrain Anomaly Perception for Safe Robot Locomotion Using a Digital Double Framework

Garen Haddeler[a,b], Hari P. Palanivelu[b], Fabien Colonnier[b], Yung Chuen Ng[b], Albertus H. Adiwahono[b], Zhibin Li[c], Chee-Meng Chew[a], Meng Yee (Michael) Chuah[b]

[a] *Department of Mechanical Engineering, National University of Singapore, Singapore*
[b] *Institute for Infocomm Research (I[2]R), Agency of Science Technology and Research (A*STAR), Singapore*
[c] *Department of Computer Science, University College London, United Kingdom*

Abstract

Digital twinning systems are effective tools to test and develop new robotic capabilities before applying them in the real world. This work presents a real-time digital double framework that improves and facilitates robot perception of the environment. Soft or non-rigid terrains can cause locomotion failures, while visual perception alone is often insufficient to assess the physical properties of such surfaces. To tackle this problem we employ the proposed framework to estimate ground collapsibility through physical interactions while the robot is dynamically walking on challenging terrains. We extract discrepancy information between the two systems, a simulated *digital double* that is synchronized with a *real robot*, both using exactly the same physical model and locomotion controller. The discrepancy in sensor measurements between the *real robot* and its *digital double* serves as a critical indicator of anomalies between expected and actual motion and is utilized as input to a learning-based model for terrain collapsibility analysis. The performance of the collapsibility estimation was evaluated in a variety of real-world scenarios involving flat, inclined, elevated, and outdoor terrains. Our results demonstrate the generality and efficacy of our real-time digital double architecture for estimating terrain collapsibility.

Keywords: robot sensing, legged robot locomotion, digital twins, field robotics

---

1. Introduction

Legged mobility has advanced greatly using model-based control techniques [1, 2, 3, 4], as well as learning-based methods that demonstrated remarkable robustness in both blind and perceptive locomotion [5, 6, 7]. In vision-aided locomotion, new advancements can decide online foot placement on uneven terrains using visual information [8, 9, 10, 11, 12]. However, perception through physical contact is not yet well researched but is critical to differentiate between rigid landscapes, such as concrete, asphalt, or hard rock; and non-rigid ones, such as mud, sand, or unstable rock piles. Clearly, using visual feedback alone is insufficient to obtain this crucial information that can only be perceived through physical interactions.

Maintaining the dynamic stability of legged robots on collapsible, deformable, or unstable/moveable terrains remains a research challenge. Early detection of the physical properties of terrain surfaces when the foot comes into contact with the ground is crucial to prevent locomotion failures. Given the short contact time during dynamic locomotion, traditional approaches via the estimation of stiffness do not offer sufficiently early detection to prevent failures.

This work is motivated to investigate the use of physics simulation that is becoming more computationally efficient and can be leveraged to apply the digital twinning idea for legged locomotion. In particular, we aim to develop new solutions with real-time performance to improve terrain anomaly detection and environmental awareness. Various high-fidelity simulators have proven their extensive use in robotics, i.e. Pybullet [13], Mujoco [14], Gazebo [15] and Isaac [16], which offer high precision physics for developing control algorithms that can be effectively and directly deployed to the real robots.

High-fidelity physics simulation can be used to execute the locomotion controller as well as fully simulate and observe all the robot states, which captures both the interaction and interplay of physics (eg, ground

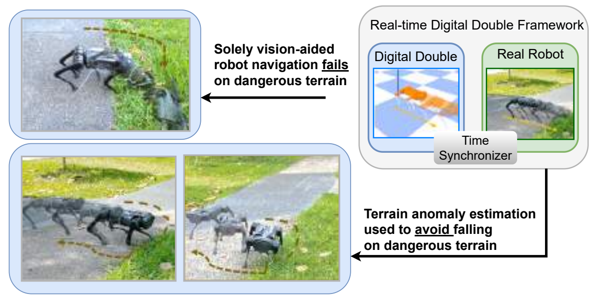

Figure 1: Real-time digital double framework enables robot perception of terrain anomalies and prevention of locomotion failures in presence of environmental uncertainties. Dangerous terrains such as pitfalls covered by grass are not identifiable by vision detection only.

impacts) and control performance. In contrast, such detailed dynamics and coupled effects of locomotion controllers are not offered by traditional motion planning based on ideal equations of motion, which motivates us to employ the very same controller and high-fidelity physics simulation altogether.

These aforementioned factors share the common ground with the digital twin technology where physics simulation is leveraged to monitor the physical process. Thus, we draw inspiration from the digital twins and use it to enhance the perception of the ground surface by using the discrepancies between real and simulated systems. This method allows the real robot to actively respond to detected terrain anomalies and paves a way for future research on decision-making in real-world applications, particularly in unanticipated emergencies.

Based on the fundamental principles of physics, we can deduce that by fully simulating the physical interactions being governed by the very same control system, we are able to access the full physical states of the robot, and thus extract more useful information to detect and discern subtle signals that are caused by soft, collapsible, or any unexpected ground properties. In our framework, a fully-simulated robot is running in parallel, synchronizing with a *real robot*. For example, the digital robot is traveling on the ideally expected rigid ground, while the *real robot* is traversing through the real terrain with unknown properties. Then, by interpreting the motion discrepancies of two systems, the estimation algorithm uses such differences between digital and real system states to perceive the non-rigidness or anomalies of the ground at each time step.

Hence, by exploiting the high-dimensional discrepancy between simulated and real-world observations, our method effectively detects terrain anomalies during dynamic walking on various grounds with uncertainties, where quick reaction time is critical. Fig. 1 shows a failure case of traditional vision-aided navigation on the ground with unknown properties where a pitfall is covered with grass. Our real-time digital double framework is able to detect such cases, thus triggering reflex control to withstand and avoid a potential fall.

1.1. Related Work

A digital twin is a digital representation produced by integrating real-world physical properties [17]. The digital information would be a 'twin' of the information embedded inside the physical system, that is synchronized to a simulated environment during the system's entire existence. It is created through the simulation of physical forces on rigid bodies over time and helps to prevent undesirable emergent behavior in complex systems. The use of digital twin technology to make decisions in actual robots by estimating information from the past and present and forecasting future states has grown in recent years [18, 19]. It is also used to monitor operations and logistic data in [20, 21, 22].

Kritzinger et al. [23] proposed a categorization of industrial digital twins based on data integration level. They defined 'digital shadow' as a one-way state update from physical to digital system, while 'digital twin' updates both ways for real-time decision making. Drawing inspiration from these works, we developed a method to synchronize physical and digital models in real and simulated mobile robots and observe differences in sensing measurement. Thus, we hereby define the simulated robot as *digital double*.

In this study, the physics simulator – Pybullet with Bullet as its physics engine – was chosen for the *digital double* based on a prior benchmark [24]. It is widely used by many in research involving robot manipulators and deep learning such as [25], [26], [27]. The physics engine Bullet is a well-documented and lightweight library that is well-known in the locomotion community and can simulate rigid body collision, joint actuation, and non-rigid object dynamics with high fidelity at a good real-time factor [28].

Several methods for ground terrain classification have been specifically developed for wheeled robots based on various sensors. Wheel-terrain vibrations are trained with a classifier to recognize sand, gravel, and clay [29]. Others have used a self-supervised learning approach with vision and proprioception [30]. Another method for terrain classification uses a single-axis accelerometer on a tactile probe attached to a low-velocity mobile robot [31]. However, these methods are not suitable or compatible with floating-base legged robots, because require the robot torso orientation to be controlled independently, which is not the case for floating-base systems.

Legged robots are versitile for navigating challenging terrains, but the risk of falling increases in unknown and unpredictable environments, which demands new research to identify ground characteristics effectively. Mathematical models to estimate foot-terrain interaction have been developed, starting with Krotkov's early work [32] and continuing with more recent efforts by Ding *et al.* [33]. However, these models were only developed and tested in static experimental setups and not on actual mobile robots. As investigated in Section 3.4, real-world testing shows that despite normal forces can be used to examine terrain-foot interactions, they exhibit significant fluctuations even during single-foot load tests, making it difficult to use as an input signal for movements that undergo dynamic foot-ground contacts. Moreover, dynamic walking gaits also introduce ground impacts, high noise, and spikes in the force data from the contact foot, which lacks of accuracy and is difficult to use based on mathematical modeling (i.e. ground stiffness, damping) which entirely depends on a specific set of quantities in model-based formulas.

In presence of limited reliability of single-source signals, the alternative is to utilize learning algorithms for soil and terrain classification that can work with redundant data inputs. Various approaches have been proposed, such as using Adaboost classifier with haptic feedback [34], self-supervised learning using vision [35], and supervised learning with RGB-D data and joint torque measurements [36]. However, these classification techniques are limited to terrains similar to previously seen examples.

Previous studies on terrain estimation or classification equipped legged robots with extra foot-mounted sensors and modified the static walking gaits [37, 38, 39, 40, 41], or extended stance-swing duration to more accurately capture sensory data [42, 33]. However, these methods rely on pre-trained ground features and become less effective in identifying unknown ground properties. Moreover, most legged robots have limited onboard sensing (the body IMU and joint encoders), thus estimating unknown ground characteristics during very dynamic gaits remains a difficult task.

1.2. Contribution

This study expands the prior work [43] extensively, and conducted more rigorous comparisons, real-world validations, and extended integration for successful safe locomotion in the wild. Specifically, the research work offers the following contributions:

- Formalization of the theoretical fundamentals of using motion discrepancy for the inference of non-directly measurable states in a real-time digital double framework, and its application in estimating terrain anomalies and collapsibility, enhancing safe legged locomotion.

- A formalized design process that utilizes simulation data to select and compare different neural network models for effective estimation of collapsibility in the real world with high accuracy and low latency.

- A performance comparison of the collapsibility estimation using our proposed method versus the traditional stiffness calculation, demonstrating the advantages of the proposed approach in the context of dynamic locomotion.

- Extensive real-world experiments validated on various types of uneven terrains such as inclined slopes, elevated steps, and outdoor ground (including pitfalls covered by grass).

- The proposed method's potential for a wide range of applications is demonstrated through its successful early detection of non-rigid collapsible terrains and execution of a velocity-based reflex behavior to prevent robot falls.

The paper is structured as follows. Section 2 elaborates the real-time digital double framework, including a collapsibility metric and dataset, followed by an explanation of using motion discrepancy as input for the learning model. Section 3 evaluates the proposed framework in real-world scenarios and compares our approach with the traditional methods. Section 4 concludes the work and highlights future research.

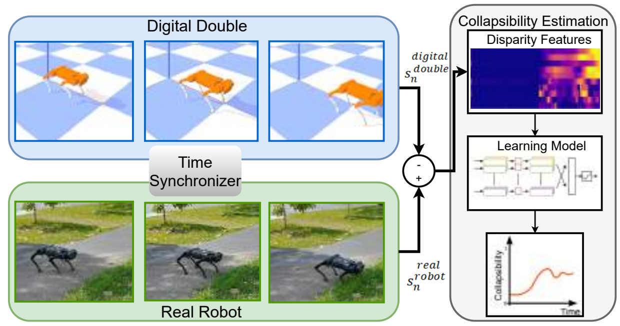

Figure 2: The proposed real-time digital double framework: The time synchronizer module ensures a consistent timing between the *real robot* and the *digital double*. The *real robot* traverses over unpredictable terrain as the *digital double* traverse over ideal ground. The disparity between both robots is inferred as the collapsibility of the terrain, and thus *real robot* is able to perceive the non-rigid ground property.

2. Methodology

Section 2.1 presents the design of our real-time digital double framework and its use on legged robots. Collapsibility detection is then formulated as supervised learning in Section 2.2. Collapsibility is defined in the context of our study, with details on data collection (Section 2.3) and the formulation of neural network input (Section 2.4).

2.1. Real-time Digital Double Framework

The core function of the real-time digital double framework is to synchronize the loop-timings at each control step for both the real and the digital locomotion controller running in parallel. The real-time digital double framework, described in Fig. 2, is designed to

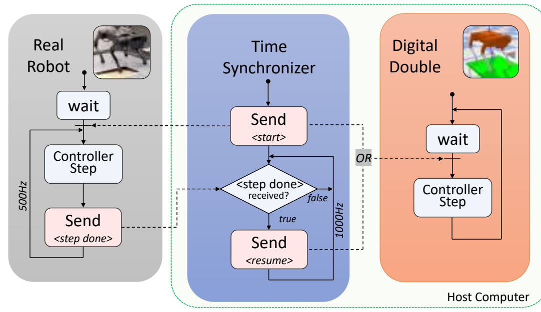

Figure 3: Time synchronizer and its interaction with the *real robot* and its *digital double* controllers. Same controllers are running on them and each time step is synchronized through the time synchronizer module.

reproduce the robot behavior in a simulated environment in real time. To achieve this, a full multi-body dynamic model of the robot is used with an accurate Unified Robotic Description Format (URDF) file, as a digital clone of a Unitree A1 robot [44].

The A1 robot has 12 actuated joints, where each leg consists of the abduct, hip, and calf joint motors. Moreover, the same locomotion framework is used for both the simulated and the *real robot*. It is composed of two modules, the Model Predictive Controller (MPC) and the Whole-Body Controller (WBC) as in [2]. The *digital double* is simulated in the Pybullet physics simulation [13].

To ensure that the real robot and its digital counterpart perform a matching motion profile at each time step, a control synchronization between the two is critical for creating a *digital double*. To overcome the communication issues between two systems that have non-linear and unpredictable time delays, we propose to observe these two individual dynamic systems' states (real and digital robot), hault the physics simulation which runs faster than wall-clock time (ie, real-time factor $>$ 1.0) and awaits for the real world to match the same timing.

This technique is similar to synchronized control of two multi-agent systems where one of the agents is in a simulation. Therefore, we designed the time synchronizer (see Fig. 3) to guarantee that the *real robot* and the *digital double* run their control loops at the same rate. It is based on the premise that the simulation is able to match the rate of the *real robot* control step and pause the physics engine if it runs faster than a real-time factor of 1.0, which is feasible given the computational efficiency and speed of modern physics engines and computers.

To perform this fucntion, the time synchronizer module interfaces with the *real robot* and the *digital double* by establishing socket connections between them. Once the connections are established, each controller provides the signal *ready* to the time synchronizer at the beginning of each control loop and then waits to receive a common *start* signal to initialize at the same starting time. With the *start* signal broadcasts, the *real robot* and the *digital double* run their controller steps at the same time.

Then, the *real robot* controller sends a *ready* flag and a *control step* counts to the time synchronizer module. Similarly, the *digital double* controller also communicates the same information to the time synchronizer module. When the *control step* in *real robot* controller is found to be ahead or matched of its *digital double* counterpart, the time synchronizer will send out the *start* signal to the *digital double* to unpause the simulated

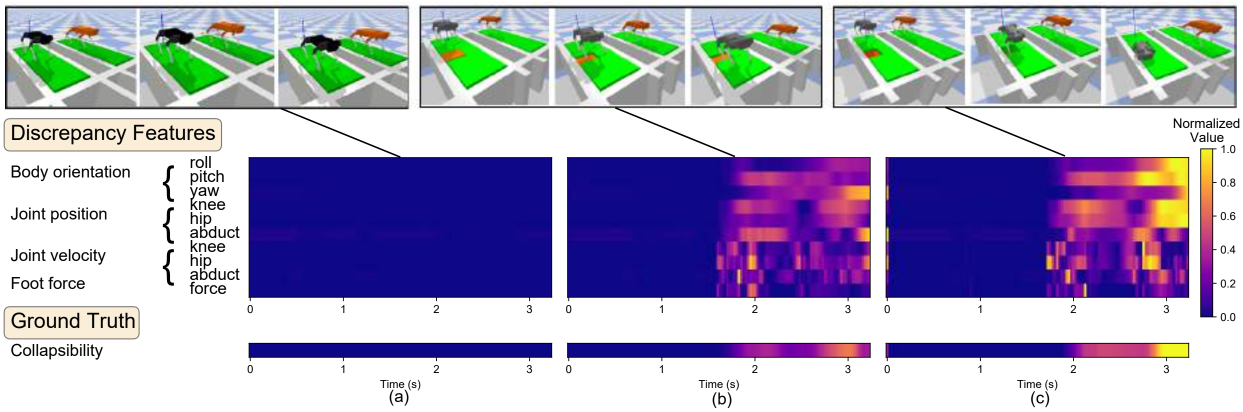

Figure 4: Simulation examples of a robot interacting with different types of terrains. Image visualization shows normalized discrepancy features Body orientation (roll, pitch, yaw), FR leg joint position, velocity (knee, hip, abduct), and foot force errors, where each feature is normalized by its maximum value. The ground truth collapsibility $C_f \in [0, 1]$ is used to describe the moment the robot enters the non-rigid platform. According to ground truth collapsibility, distinct inconsistencies occur (depicted in colors other than blue) when the robot physically interacts with non-rigid tiles. (a) walks on the rigid ground; (b) uses its FR leg to step on the semi-collapsible ground; and (c) uses its FR leg to step on the collapsible ground.

world. Otherwise, the *start* signal will be $false$ and the simulated world pauses, and time synchronizer will wait for the *real robot* controller to catch up with the *digital double*.

2.2. Collapsible Terrain

In the context of this research, we focus on the use of such a developed framework to discern unexpected ground conditions, assuming the ideal expectation is rigid ground. Similar to the previous work [43], here the term ‘collapsible terrain’ refers to the ground that appears to be safe to walk over according to visual cues but actually can deform or collapse while the robot is walking over. The context of our application is particularly for the uncharted scenarios that are not resolved by vision-dominant solutions.

We formulate a dimensionless measurement to describe the relation between the vertical displacement of the surface and the maximum permissible deviation of the terrain. The permissible deviation is the kinematic reachability of the robot leg, which is the maximum reachable range – falling is inevitable since the ground sinks deeper than the leg can possibly reach.

As elaborated in Section 1.1, instead of classifying particular terrain properties which have to be determined beforehand plus installing additional sensing hardware, we can observe robot motion anomalies and infer collapsibility information based on the physical causality of dynamical movements. These motion anomalies can be extracted by comparing the states of *real robot* and high-fidelity *digital double*, which are created using the real-time digital double framework proposed in Section 2.1.

For individual feet, the terrain's collapsibility is noted as $C \in \mathbb{Q} : C \in [0, 1]$. $C \approx 0$ is considered as *rigid, non-collapsible* terrain that is described as having stiffness of hard ground with a deformation of $x \approx 0cm$. In comparison to rigid ground, *semi-collapsible* terrain $C \in (0, 1)$, a value of C between 0 and 1, is substantially less stiff. But according to the definition, the terrain deformation is kept in the interval $0 < x < x_{max}$. Therefore, semi-collapsible ground, even if it deforms upon contact with the ground, can be regarded as relatively safe to walk on. Even if theoretically it is feasible to traverse across these terrains, it will be challenging, and therefore robots should be aware of such terrain and may avoid it if it necessary. When terrain is *collapsible* $C \rightarrow 1$, it signifies that the permissible maximum sinking depth is exceeded, which will lead the robot to fall or a failure of the locomotion control.

2.3. Dataset Collection

To train the neural network model in a supervised learning manner, a dataset is collected in a fully simulated environment for both the simulated '*real robot*' and the *digital double*. In the simulation, two quadrupedal robots are walking simultaneously over different areas; one over a non-rigid collapsible ground and the second over hard ground. From the different scenarios, we extract both the ground truth values of collapsibility C_{gt} and the sensory measurements. To simulate non-rigid ground, a tile is allowed displacement with customizable stiffness and damping. Identical environmental setup is used as thoroughly explained in our previous article [43].

A total of 40 trials were conducted: 10 repetitions for each leg (4 legs, ie, Front-Left (FL), Front-Right (FR), Back-Left (BL), Back-Right (BR)) that first enters and steps on the movable platform, which simulates semi-collapsible (5 trials) and collapsible terrain (5 trials), respectively. We designed these four separate stepping cases for each leg because the disturbance of the robot's body differs and is unique depending on which leg. And then, to correct and restore stable walking, the control system in turn produces different and various reactions and motion patterns. Therefore, to form a complete dataset and gain full observation of various processes, we tested and collected data of each individual leg as well as their interpretation of each other.

For all the repetitions, the gait and the reference velocity are kept constant, ie, trotting at $1.5m.s^{-1}$. Among all configurations, 20 % of the data is used for testing and the rest is for training. The sampling rate is chosen as 50Hz and the duration of each semi-collapsible and collapsible trial is 8 seconds with an average of 400 data samples and 4 seconds with an average of 200 data samples, respectively.

2.4. Input of the Neural Network (NN)

The neural network inputs are considered as the state discrepancy between the *real robot* and *digital double*, at the same time step (n). The input of the neural network (x) is the states that accumulated through a time window k as Eq. 1.

The state discrepancy is expressed as Eq. 2. The body orientation Θ is written as a vector of roll, pitch, and yaw Euler angles. The knee, hip, and abduct joints' positions and velocities are q and $\dot{q}$, respectively, and f indicates the reactive forces on the feet. R and D, are stand for *real robot* and *digital double*, respectively.

$$x = \left[s_n, s_{n+1}, ..., s_{n+k-1}, s_{n+k}\right] \tag{1}$$

$$s = \begin{bmatrix} \Theta^R \\ q^R \\ \dot{q}^R \\ f^R \end{bmatrix} - \begin{bmatrix} \Theta^D \\ q^D \\ \dot{q}^D \\ f^D \end{bmatrix} \tag{2}$$

Fig. 4 shows the error in the FR foot force, as well as the errors in the knee, hip, and abduct joint positions and velocities and body orientation. The errors are presented in a discrepancy feature image, with each feature being normalized by its maximum value. The image shows three cases labeled as simulated snapshots, with the robot walking across FR-placed tiles that are non-collapsible, semi-collapsible, and collapsible. The normalized error image displays distinct patterns and the differences in the errors increase as the ground truth collapsibility increases.

The motion discrepancy of real and simulated robots does, in fact, show a correlation to how collapsible the surface is. When the robot steps on collapsible terrain, there is a noticeable increase in the discrepancy between the real measurements and their expected values from the digital double. This occurs because the foot reaction forces are not strong enough to prevent the foot position errors from increasing. As a result, the errors persist even though the legs have feedback control. The patterns in these errors can be learned by machine learning models and used to estimate the collapsibility, as shown in Fig. 4.

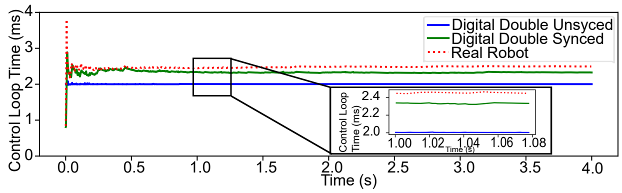

Figure 5: Comparison of using *digital double* with and without using time synchronizer. The control loop time is taken during trotting on flat ground in the real world. Zoomed area shows the control loop time in $0.08s$ time window.

3. Results

The effectiveness of the real-time digital double framework is evaluated in Section(3.1). The accuracy and inference time of different learning models for collapsibility estimation are compared in Section 3.2. The selected neural network is then tested for real-world implementation in Section 3.3. Finally, the results of ground stiffness and proposed collapsibility estimation methods are compared in Section 3.4.

3.1. Assessment of the Real-time Digital Double Framework

The *real robot* and its *digital double* walked on a flat surface to demonstrate the real-time digital double framework performance. The

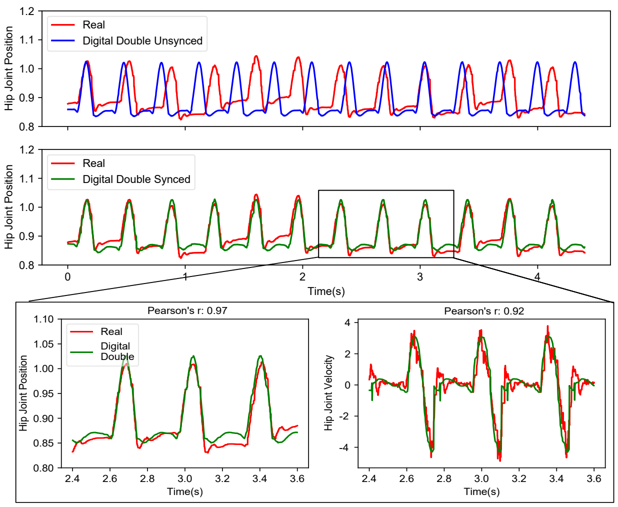

Figure 6: The joint position profiles of the FR hip joint, from the *real robot* and its *digital double* respectively during trotting on flat ground in the real world. Comparison of a) with and b) without using the real-time digital double framework. The high Pearson's r score with real-time digital double framework shows a strong correlation of hip joint position and hip joint velocity signals from the *real robot* and its *digital double* in between $2.3 - 3.6$ second.

robot's onboard computer handled the locomotion controller, while the time synchronizer and physics simulation were on a host PC connected to the robot through Ethernet. The host PC had an Intel Core i7-10870H CPU with 8 cores clocked at 2.20 GHz and a GeForce RTX 3070 GPU for simulation rendering.

In the experiments, Fig. 5 shows the control loop time between the real robot, the digital double with a time synchronizer, and the digital double without a time synchronizer. Control loop time is calculated by measuring the passed time in a single iteration of the legged locomotion controller.

The digital double without a time synchronizer has a faster average control loop time ($2.0ms$), compared to the real robot ($2.5ms$) within a $0.08s$ time window, causing misalignment in time between the joint profiles of the two systems. This is due to the simulation computer having more powerful computation and leads to a real-time factor of more than one. In contrast, the use of the time synchronizer mitigates this difference and makes the average control loop time closer to the real robot ($2.3ms$). As a result, the controllers of both the real robot and the digital double with time synchronizer performed similar actions at a given time, which reduces the reality gap between the real and simulated system.

Fig. 6 compares the FR hip joint position of the *real robot* and its *digital double* with and without the time synchronizer. The results show that without the time synchronizer, the *digital double* fails to remain synchronized with the real robot over time. On the other hand, when using the time synchronizer, the control loop is re-aligned each time, thus both the real and digital robots generate matching profiles in terms of FR hip joint position and velocity, indicating that the time synchronizer successfully maintains the synchronization between the two systems.

When the robot walks on flat ground, the *digital double* has similar dynamic results to the *real robot*: both joint position and velocity profiles have matching amplitudes. Within the time window of 1 second, phases of the *real robot* and *digital double* are fully synchronized. A strong positive correlation between joint position and velocity is observed with Pearson's r scores of 0.97 and

0.92 respectively, where Pearson's r score in statistics is defined as a linear correlation coefficient of two sets of data.

3.2. Simulation Comparison of NN Designs

We investigated the performance of the three different network structures trained to estimate collapsibility in the simulation environment. Three studied network structures are:

- Convolution Neural Network (CNN) model consisting of 2 1D-convolution layers followed by ReLu activation and Max pool layers. The two convolution layers have kernel sizes of 5 and 3, producing 280 and 120 channels respectively. The model then has 2 fully connected layers with 120 hidden neurons to generate a single output value.
- Gated Recurrent Unit (GRU) model composed of 1 GRU unit with 64 hidden layers and 1 fully-connected layer to provide a single value output.
- Recurrent Neural Network (RNN) model composed of 1 RNN unit with 64 hidden layers and 2 fully-connected layers to provide a single value output.

The input for learning models is chosen as the set of features (Θ, q and, $\dot{q}$) which are body orientation angle (yaw, pitch, roll), joint angle position and velocity (hip, thigh, knee). The sliding window time is 0.04 seconds and inputs are gathered over 0.5 seconds. 100 epochs are processed at a constant learning rate of 10^{-4}. The CNN, GRU, and RNN models have 243641, 217217 and 80305 trainable parameters, respectively.

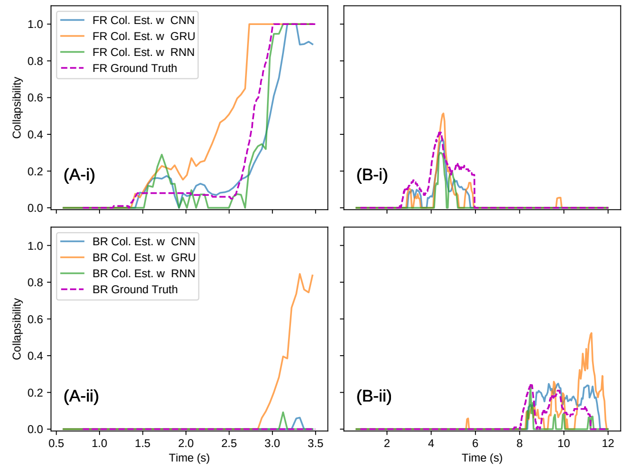

Figure 7: Collapsibility estimation in Simulation Environment with Different NN Models. (A) The case where a collapsible tile was placed FR and its sinking depth was more than x_{max} ($x \geq 10cm$, $C_{gt} \rightarrow 1$), (B) The case where a semi-collapsible tile with $C_{gt} \in (0, 1)$ was placed at FR, (i) Front-Right (FR) foot estimation, (ii) Back-right (BR) foot estimation.

The ground truth for collapsibility is calculated by observing terrain deformation in simulation and is divided by the maximum allowed terrain deformation of 10 cm.

The Mean Absolute Error (MAE) is a metric used to evaluate the performance of models. It is calculated as the average of the absolute differences between the predicted values and the ground truth values. The formula is expressed as: $MAE(\%) = 100\frac{1}{k}\sum_{t=0}^{k}|C_{gt}(t) - \hat{C}(t)|$, where $C_{gt}(t)$ is the ground truth collapsibility at timestamp t, $\hat{C}(t)$ is the predicted collapsibility, k is the number of time stamps, and $|C_{gt}(t) - \hat{C}(t)| \in [0, 1]$.

The results of the estimation for different

Table 1: Statistics for the different model over the experiments presented in Fig 7

	FR MAE(%)		BR MAE(%)		Inference Time	
	col.	semi-col.	col.	semi-col.	col.	semi-col.
CNN	7.06	**2.65**	**0.10**	2.95	**2.51**	**2.4**
GRU	14.42	3.27	5.29	4.72	20.1	20.4
RNN	**6.40**	4.08	0.16	**2.88**	7.18	7.11

models and the ground truth of collapsibility are shown in Fig. 7. It displays the collapsibility estimation for front and back feet of the robot stepping on a collapsible (7-A) and a semi-collapsible tile (7-B). The results show that all proposed models are able to capture temporal patterns among the data, can accurately estimate the rigid, semi-collapsible, and collapsible tiles and differentiate the terrain deformation for each foot.

The average MAE and average input/output response time of the models are presented in Table 1. The results indicate that the CNN model has one of the smallest MAEs for both front and back feet and the fastest input/output time, compared to RNN and GRU models in collapsible and semi-collapsible cases. While the RNN model is slightly more accurate in two cases, compared to the CNN model, it has almost three times longer inference time which would cause delays in detection. Hence, the CNN model is selected in this study for developing the collapsibility estimation.

3.3. Collapsibility Estimation in Real-world Implementation

The proposed collapsibility estimation framework was validated through indoor and outdoor experiments to demonstrate its robustness and generalizability. The experiments were conducted using a Unitree A1 quadruped robot [44] and a host computer running the real-time digital double framework. The collapsibility ground truth (C_{gt}) was determined through 2D motion capture program Kinovea [45] for indoor trials, and through the robot's success or failure in traversing different depths ($> 10cm$) in outdoor experiments. The indoor experiments included flat terrain, inclined slopes, and elevated single steps. In outdoor scenarios, the framework was tested by estimating the collapsibility of a pitfall covered with grass, enabling the use of a velocity reference-based reflex controller for failure prevention and safety.

3.3.1. Flat Ground

The real-world semi-collapsible and fully collapsible cases were tested on flat ground with a similar setup to the simulation, as shown in Fig.4. The FR foot successfully estimated collapsibility with 6.9% MAE on the flat collapsible ground, as seen in Fig. 8-A. MAE results show the data from $0 - 3.5s$ before the robot fell down completely.

On the flat semi-collapsible ground, the robot was able to traverse with its FR and BR legs and both were able to estimate

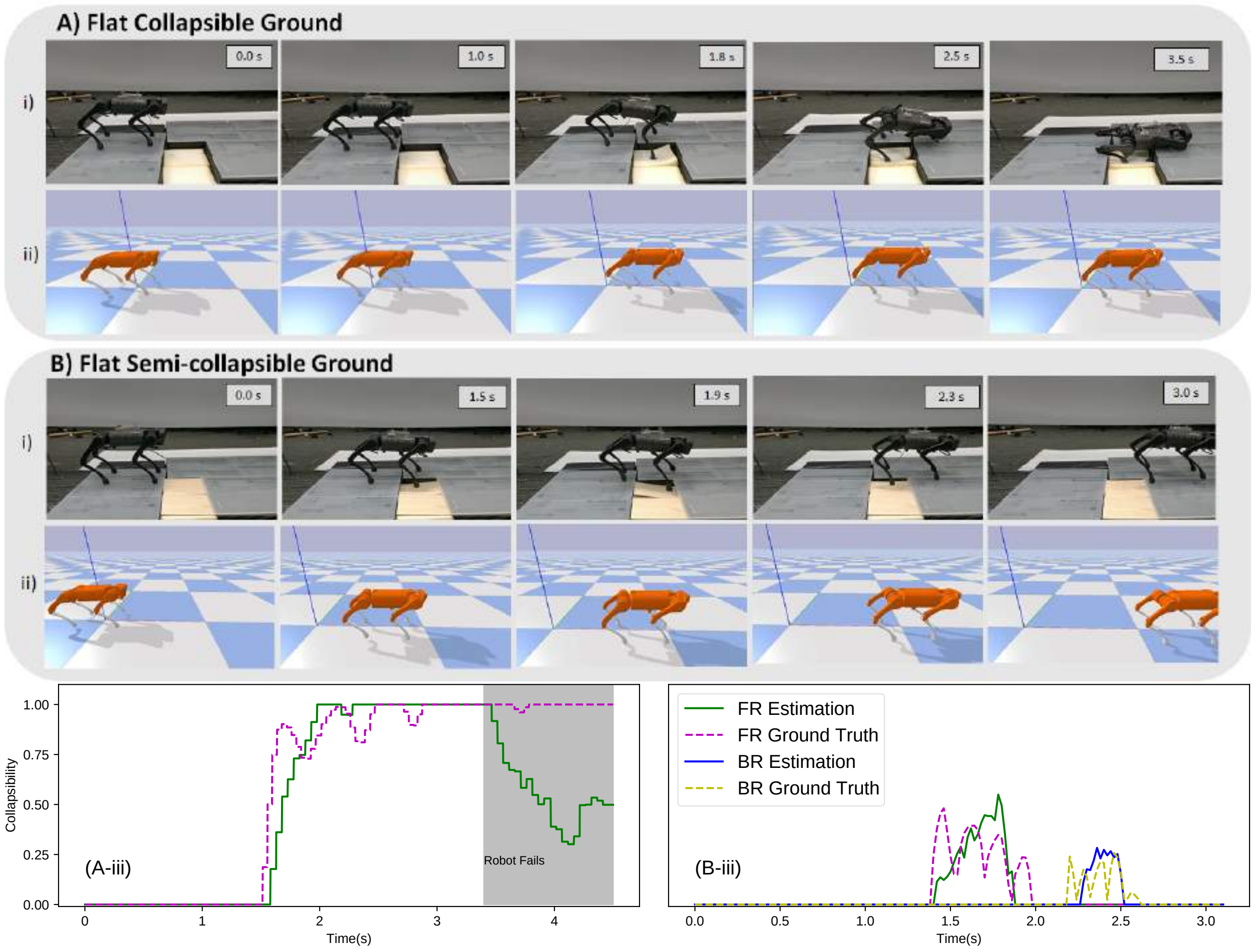

Figure 8: Online collapsibility estimation on **flat ground**: (A) *real robot* steps over a collapsible tile with its front-right (FR) feet, (B) *real robot* steps over a semi-collapsible tile with its FR and back-right (BR) feet. i) Snapshots of the *real robot* at 3 different instants, ii) *digital double* snapshots at the corresponding timestamps, iii) collapsibility estimation over time compared to ground truth.

the collapsibility with MAE of 2.87% and 1.03% respectively, as seen in Fig. 8-B. The real flat ground experiments show that our digital double framework provides consistent results, showing sufficient accuracy using trained learning-based models in simulation and used with real data directly.

3.3.2. Inclined Ground

The following experiment was performed on a 7-degree inclined slope, which was placed in both real-world and simulated environments, even though this scenario was not encountered in the training dataset. The robot was given different directional commands to

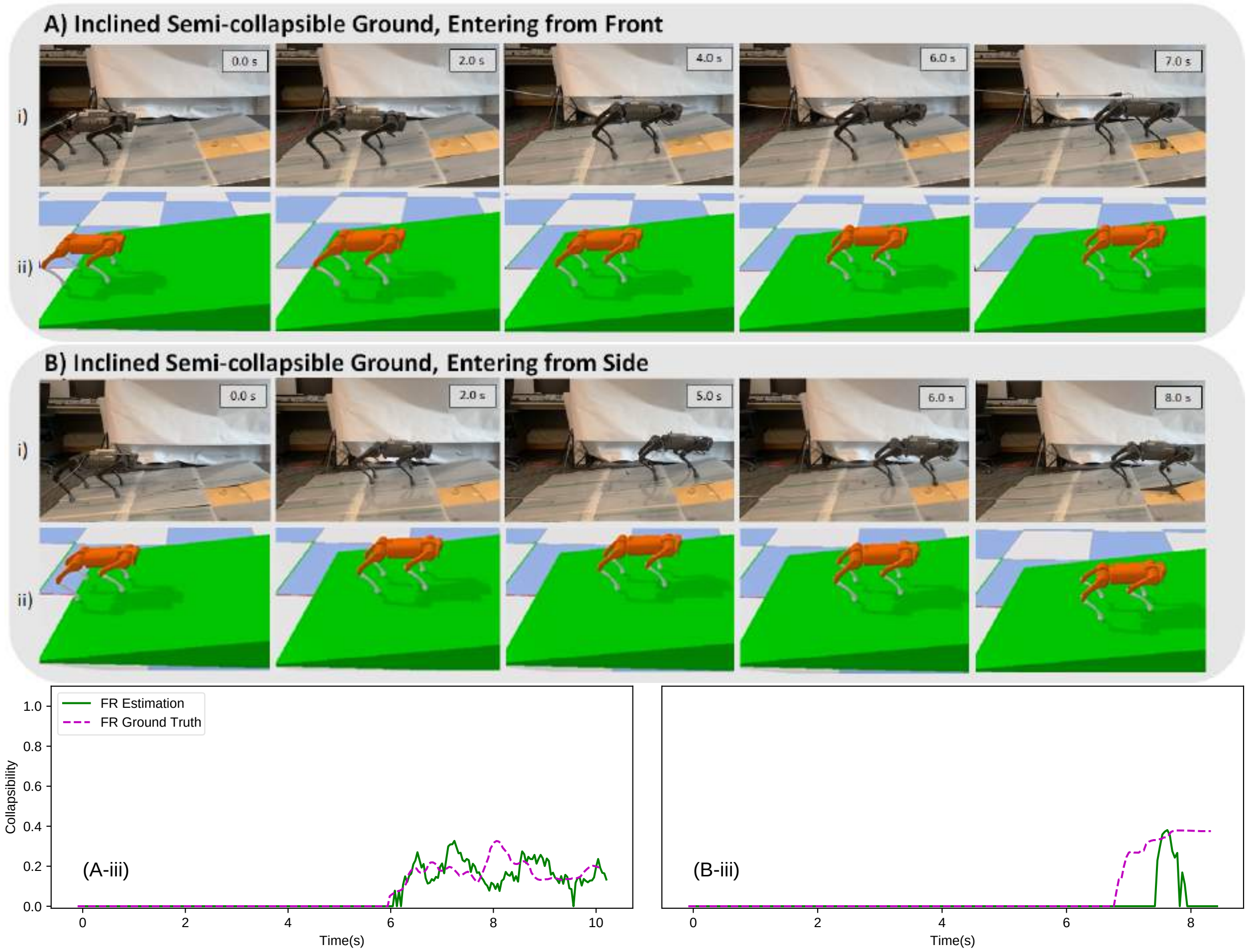

Figure 9: Online collapsibility estimation semi-collapsible experiments on the **inclined ground**: (A) the robot step from the front and (B) from the side. i) Snapshots of the *real robot* at 3 different instants, ii) *digital double* snapshots at the corresponding timestamps, iii) collapsibility estimation over time in green compared to ground truth in dashed purple and yellow.

enter the semi-collapsible tile from the front and right side, as shown in Fig. 9-A and B respectively. The results showed that the FR foot estimated the semi-collapsible tile with an MAE of 2.5%, 4.88% compared to ground truth. This demonstrates that given the NN training dataset, the discrepancy obtained from the digital double framework was sufficient to infer terrain collapsibility in the scenarios of an inclined slop up to 7 degrees, as well as different traveling directions.

3.3.3. Elevated Ground

In the experiment with uneven ground, a $30cm * 30cm * 6cm$ step is placed in both real-

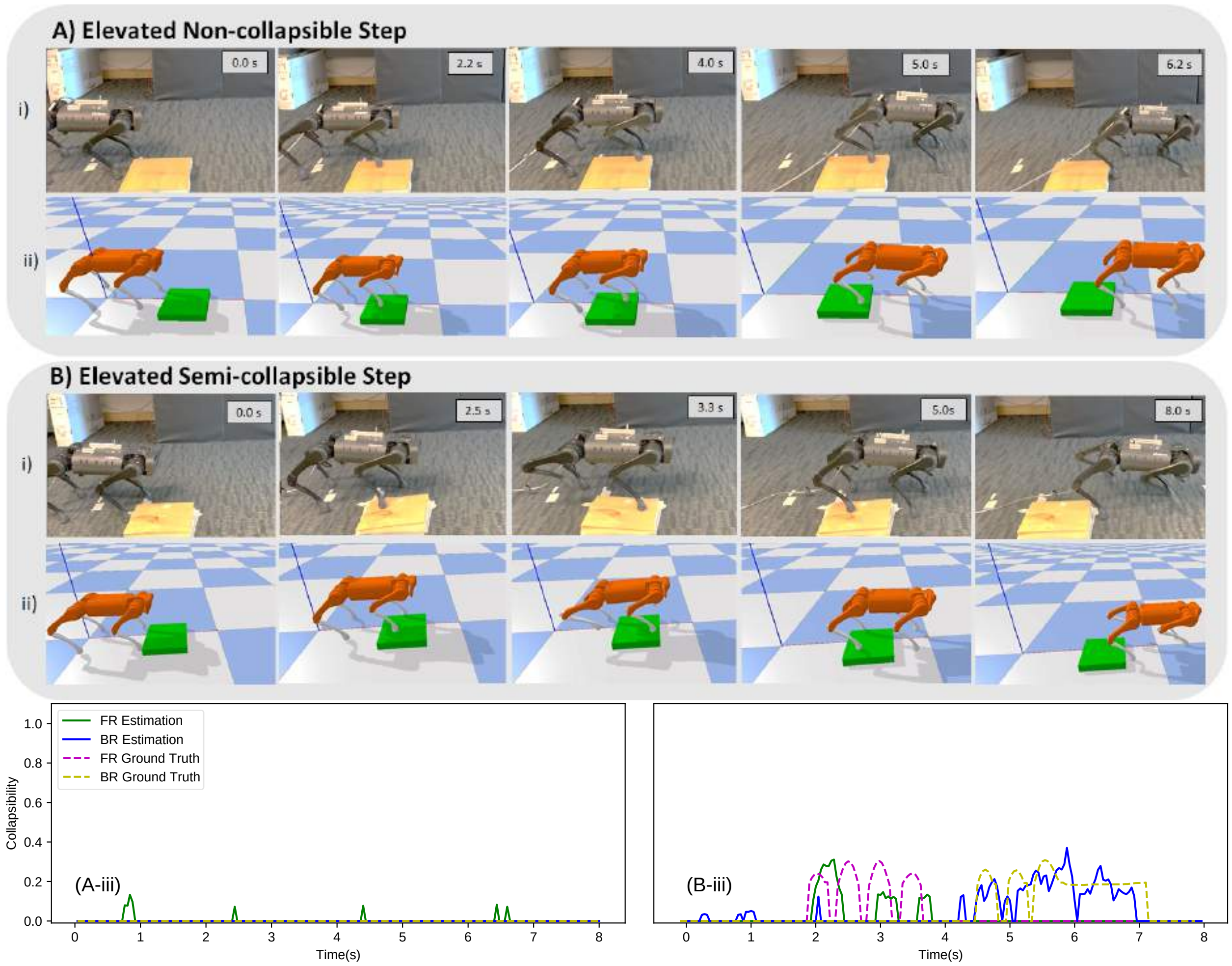

Figure 10: Online collapsibility estimation in **elevated single step**: (A) *real robot*'s FR feet step over a non-collapsible step, (B) *real robot*'s FR feet step over a semi-collapsible step. i) Snapshots of the *real robot* at 3 different instants, ii) *digital double* steps over rigid step and its snapshots at the corresponding timestamps, iii) collapsibility estimation over time compared to ground truth.

world and simulated environments to test the proposed framework's ability to differentiate between hard, non-collapsible steps vs semi-collapsible steps. The robot is commanded to step over the non-collapsible step in the first case and the semi-collapsible step in the second case shown as Fig. 10-A, B, respectively. The FR feet estimate the non-collapsible step with a MAE of 0.34% compared to the hard ground ($C = 0$). The FR and BR feet estimate the semi-collapsible step with MAE of 3.76% and 3.08%, respectively. The results show that proposed method is able to distinguish and estimate semi-collapsible and hard

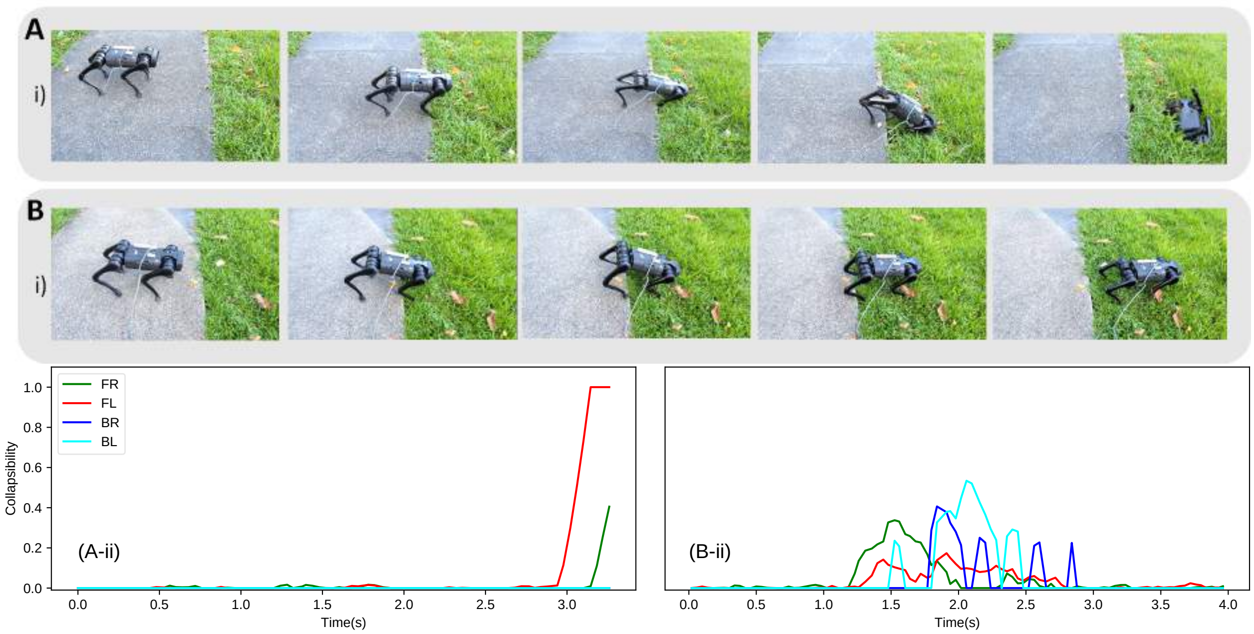

Figure 11: Online collapsibility estimation in **outdoor terrain** where no reaction was used for the validation purposes: Pitfalls covered with grass that represent A) collapsible and B) semi-collapsible terrains. i) snapshot of front-left (FL) and front-right (FR) feet steps over the grass; ii) shows collapsibility estimation of all four legs over time.

step.

3.3.4. Outdoor Scenario

In this experiment, the aim is to show that the proposed framework can detect and distinguish between terrains that seem traversable visually but actually collapse while being physically interacted with. The experiment is designed to represent scenarios where dense vegetation (long grasses) can obstruct the visual perception of the actual terrain depth in realistic real-world applications.

Fig. 11 displays outdoor experiments with robots navigating collapsible and semi-collapsible terrains, and shows the collapsibility estimation of each leg over time. In Fig. 11-A, the robot is unable to cross a patch of grass deeper than $10cm$, while in Fig. 11-B, it is able to successfully traverse the terrain when the grass height is less than $10cm$. Our collapsibility estimation can successfully distinguish between collapsible and semi-collapsible terrains.

3.3.5. Reactive Motion upon Detected Collapsibility

This experiment was designed as an extension to demonstrate the use of our digital double-based collapsibility estimation in a representative application – using it for early detection of non-rigid collapsible terrain and

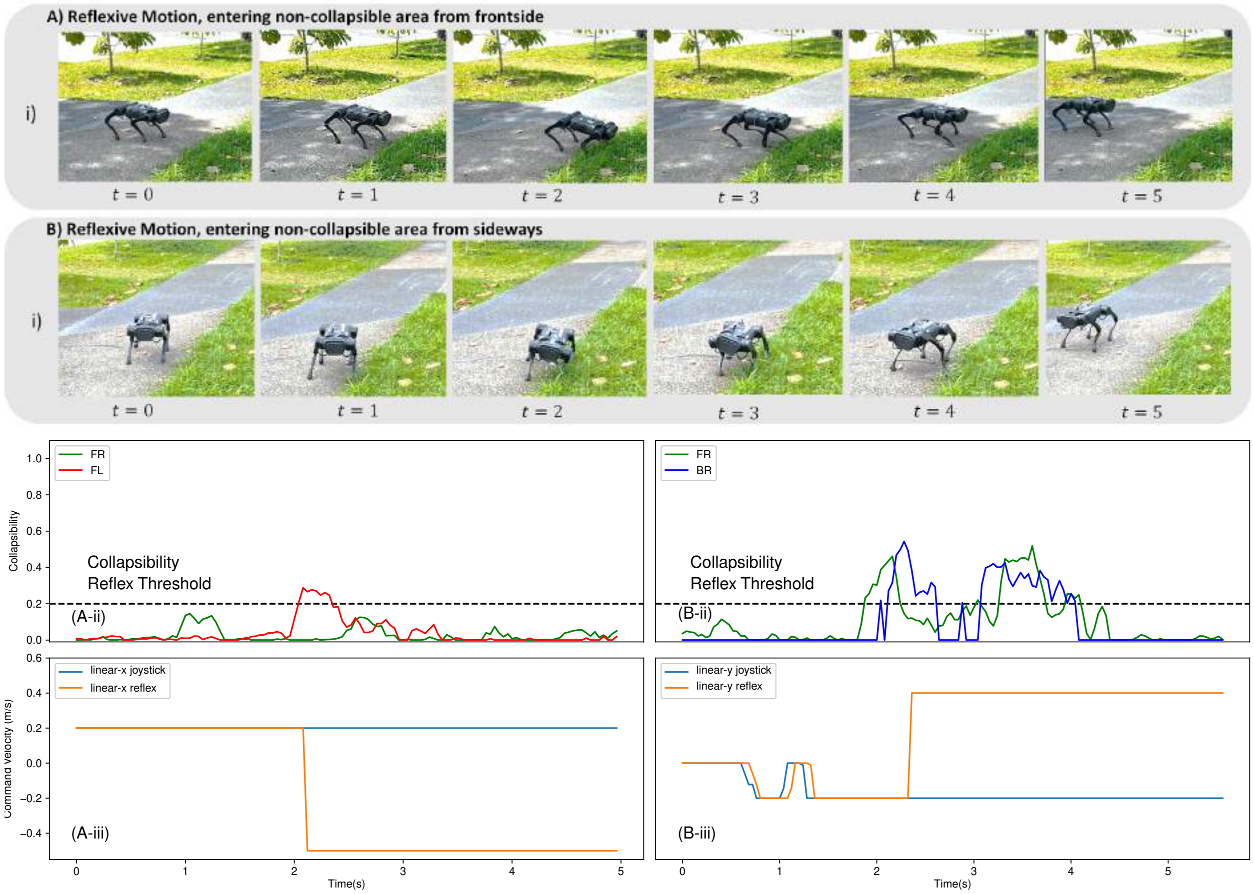

Figure 12: Safe robot locomotion using the proposed method for fast terrain collapsibility detection. The real robot was walking towards the semi-collapsible area from A) front side B) sideways while **reflexive motion** was activated upon the detection of collapsibility: i) snapshots showing robot entering the edge of the non-rigid terrain and quickly retrieving back.ii) collapsibility estimation over time, iii) linear forward (-x) and side-way (-y) directional velocities are given by joystick and by reflex commands.

integrating it with a reflex-based safety response. This enables the robot to avoid potentially unsafe terrains upon detection. Our design includes a high-level velocity multiplexer for the robot, which uses velocity input based on a hierarchy of commands. Specifically, the reflexive motion detection system has a 'reflex' command with higher priority than the 'joystick' command from the human operator.

The experiment consists of two cases in which a robot executes reflexive motion: in the first scenario, the robot enters a non-collapsible area (specifically, a pitfall covered by grass) from the front, while in the second scenario, it enters from the side. In both cases, the reflexive motion detection system overrides the 'joystick' command and utilizes 'reflex' command to avoid navigating potentially dangerous terrain.

Fig 12 shows snapshots of the robot walking towards and avoiding a semi-collapsible area, along with graphs of the robot's collapsibility estimation over time and the linear x and linear y-directional command velocities for both the 'joystick' and 'reflex' commands.

On both cases, the collapsibility estimation exceeds a threshold of $C_t \geq 0.2$, causing the high-level velocity multiplexer to override the 'joystick' command with a larger 'reflex' command that is opposite in direction and 2-3 times higher. This was done to minimize the impact on the robot and keep it away from potentially dangerous terrain.

3.4. Comparison Study of Ground Stiffness and Collapsibility Estimation

In this comparison study, the proposed method of terrain estimation is compared to the traditional ground stiffness estimation method in a dynamic walking scenario as a baseline. The study evaluated contact force readings, foot vertical displacement, and stiffness calculations while walking on rigid and semi-collapsible (non-rigid) terrains.

Terrain stiffness can be estimated by dividing the foot contact forces by the vertical displacement of the terrain ($k = \frac{F}{\Delta x}$). Foot contact forces can be estimated with a force sensor at the foot or by computing it from the joint torques τ as follows $F = (J(q)^T)^{-1}\tau$, where the foot Jacobian is $J(q)$. However, the estimation of vertical terrain displacement (Δx) of a floating-base system, the robot, is more challenging when measured with respect to the ground as an inertia frame. Δx is correlated to foot vertical displacement ($\Delta x \approx \Delta x_{foot}$) and is thus calculated as the difference between the estimated foot height at contact and the initial foot height at contact with respect to the world frame.

A semi-collapsible ground experiment is performed to compare the stiffness and collapsibility estimates. The ground truth stiffness of the semi-collapsible terrain is calculated by placing a 5kg weight and observing a 0.01 m vertical displacement. The ground is considered rigid for stiffness values above $10000Nm^{-1}$. The mean absolute error (MAE) percentage for stiffness estimation is calculated as $MAE(\%) = \frac{100}{10000}\frac{1}{n}\sum_{t=0}^{n}|k_{gt}(t) - \hat{k}(t)|$. The ground truth collapsibility and MAE for collapsibility estimation are calculated the same as in previous experiments.

The results of the experiment where the robot walks on the semi-collapsible ground are shown in Fig.13. The FR feet were able to estimate the ground stiffness with a 10.25% MAE compared to the ground truth. For collapsibility estimation, the FR feet achieved a 1.74% MAE compared to the ground truth. However, the stiffness estimation was only accurate for the first step, with large errors observed when the robot steps out of the semi-collapsible area, making it increasingly unreliable.

The traditional stiffness identification approach is limited due to the noisy and spiky estimation and limited accuracy, and the detailed reasons are as follows. First, the pre-

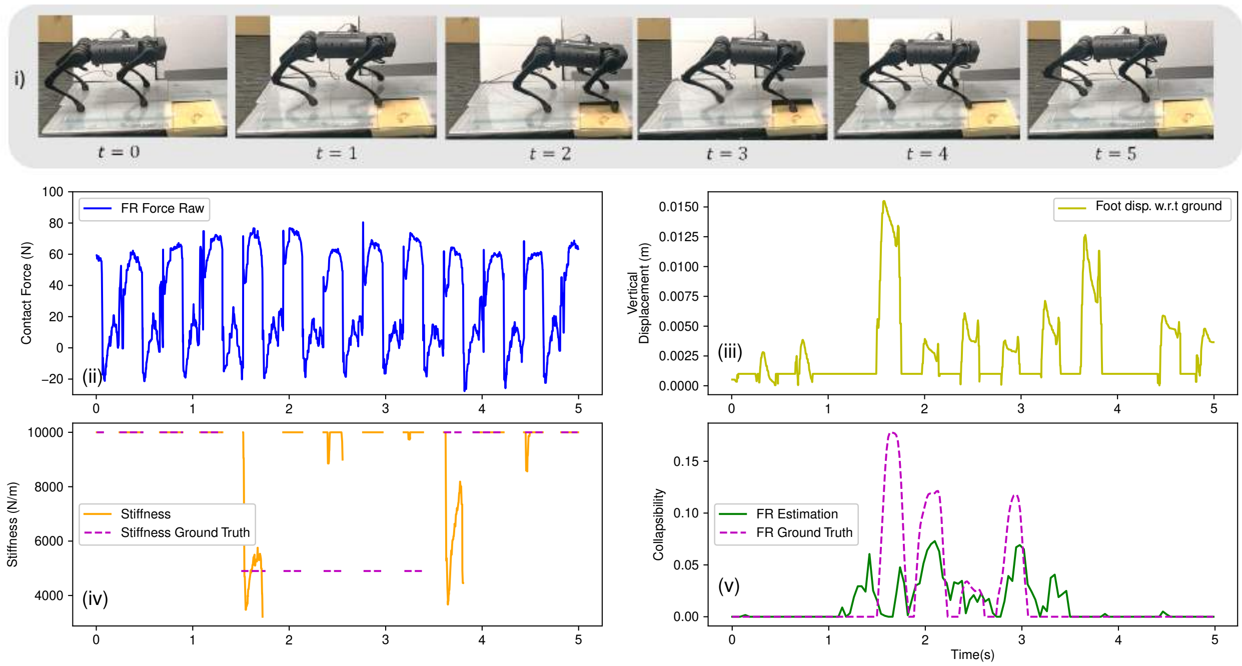

Figure 13: Comparison experiment of traditional stiffness calculation and collapsibility estimation during real-robot dynamic walking: i) snapshots of robot entering non-rigid ground over time, ii) contact foot force in a vertical direction, iii) vertical foot displacement with respect to the ground frame, iv) stiffness estimation and stiffness ground truth, v) the collapsibility estimation and collapsibility ground truth.

cision of calculating the spatial position of the foot tips is difficult by the nature of the floating-base system, which is even worsened due to very fast dynamic locomotion, sensor noises, and delays. Second, the force estimation through joint torque sensors is an indirect approach and limits the measurement sensitivity on non-rigid terrain, and adding a 6-axis force sensor on foot negatively impacts dynamic gaits due to added mass and inertia. Also, due to the cost and calibration issues, such sensors are not equipped on commercially available legged robots. Third, during dynamic walking, the impact forces fluctuate drastically, so the foot-ground contact time is very short and only a few data points can be collected. For example, a stepping frequency around 2 Hz results in a stance duration shorter than 0.125s, making it difficult to collect enough data points for denoising the spiky measurements during short stance phases.

4. Conclusion

We propose a real-time digital double framework to enhance legged robot perception of non-rigid terrain by using motion discrepancy between a digital double and a real robot. Our method achieved real-time behavior similarity between the digital double and the real robot by using time synchronizers on separate locomotion control loops. Anomalies correlated to terrain collapsibility were estimated in real-time by comparing the motion discrepancy between the digital double trotting on ideal hard ground and the real robot trotting on non-rigid terrain surfaces. We showed that our framework can perceive collapsible and semi-collapsible grounds relatively safe for the robot to step on by allowing foot displacement upon ground contact.

We evaluated various neural network architectures and proposed 1-D CNN structures to detect temporal patterns in time series data. Our approach outperformed the conventional method for calculating ground stiffness, achieving accurate and sensible estimation of non-rigid collapsible terrain. We also tested our framework on various unseen terrains, including inclined slopes and outdoor scenarios with dense vegetation obscuring the terrain's true depth. Our estimation results demonstrated generalizability among these unseen scenarios. Finally, we presented an application case for a collapsibility estimate, showing a velocity command-based reflexive motion reacting when non-rigidness is detected.

As for future work, the implementation of this low-latency and high-accuracy collapsibility module for real-time estimation of collapsible terrains will encourage further research in reactive locomotion. Moreover, fully integrated digital twins as described in the literature can be further investigated by simulating real-world terrain properties in the digital world with visual feedback and taking reactive action on detected undesired behaviors from digital systems.The integration of visual feedback and digital double can be used to forecast the future state of the system and anticipate potential failures before they occur in the real world.

5. Acknowledgments

This research is supported by the Singapore government's Research, Innovation and Enterprise 2025 Plan (RIE2025), and administered by the Agency for Science, Technology and Research (A*STAR) 2022 Robotics Horizontal Technology Coordinating Office Seed Fund

References

[1] C. D. Bellicoso, F. Jenelten, C. Gehring, M. Hutter, Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots, IEEE Robotics and Automation Letters 3 (2018) 2261–2268.

[2] D. Kim, J. D. Carlo, B. Katz, G. Bledt, S. Kim, Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control, CoRR abs/1909.06586 (2019). URL: http://arxiv.org/abs/1909.06586. arXiv:1909.06586.

[3] Boston Dynamics, With you, spot can, 2020. URL: https://youtu.be/VRm7oRCTkjE.

[4] J. Sleiman, F. Farshidian, M. V. Minniti, M. Hutter, A unified MPC framework

for whole-body dynamic locomotion and manipulation, CoRR abs/2103.00946 (2021). arXiv:2103.00946.

[5] D. R. Song, C. Yang, C. McGreavy, Z. Li, Recurrent deterministic policy gradient method for bipedal locomotion on rough terrain challenge (2018) 311–318.

[6] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain, Science robotics 5 (2020). doi:10.1126/scirobotics.abc5986.

[7] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild, Science Robotics 7 (2022). doi:10.1126/scirobotics.abk2822.

[8] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, M. Hutter, Robust Rough-Terrain Locomotion with a Quadrupedal Robot, Proceedings - IEEE International Conference on Robotics and Automation (2018) 5761–5768. doi:10.1109/ICRA.2018.8460731.

[9] O. A. V. Magaña, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, C. Semini, Fast and continuous foothold adaptation for dynamic locomotion through cnns, IEEE Robotics and Automation Letters 4 (2019) 2140–2147. doi:10.1109/LRA.2019.2899434.

[10] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, S. Kim, Vision Aided Dynamic Exploration of Unstructured Terrain with a Small-Scale Quadruped Robot, Proceedings - IEEE International Conference on Robotics and Automation (2020) 2464–2470. doi:10.1109/ICRA40945.2020.9196777.

[11] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, C. Semini, Mpc-based controller with terrain insight for dynamic legged locomotion (2020) 2436–2442. doi:10.1109/ICRA40945.2020.9197312.

[12] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, M. Hutter, Perceptive locomotion through nonlinear model predictive control, 2022. doi:10.48550/ARXIV.2208.08373.

[13] E. Coumans, Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning (2016).

[14] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control (2012) 5026–5033. doi:10.1109/IROS.2012.6386109.

[15] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator (2004) 2149–2154.

[16] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, D. Fox,

Gpu-accelerated robotic simulation for distributed reinforcement learning, CoRR abs/1810.05762 (2018). arXiv:1810.05762.

[17] M. Grieves, J. Vickers, Digital Twin : Mitigating Unpredictable , Undesirable Emergent Behavior in Complex Systems, Springer International Publishing, 2017, pp. 85–113. doi:10.1007/978-3-319-38756-7.

[18] P. Staczek, J. Pizon, W. Danilczuk, A. Gola, A digital twin approach for the improvement of an autonomous mobile robot (amr's) operating environment-a case study, Sensors 21 (2021). doi:10.3390/s21237830.

[19] J. Douthwaite, B. Lesage, M. Gleirscher, R. Calinescu, J. M. Aitken, R. Alexander, J. Law, A modular digital twinning framework for safety assurance of collaborative robotics, Frontiers in Robotics and AI 8 (2021). doi:10.3389/frobt.2021.758099.

[20] M. Grzelak, A. Borucka, A. Świderski, Assessment of the influence of selected factors on the punctuality of an urban transport fleet, Transport Problems 15 (2020) 311–323. doi:10.21307/tp-2020-069.

[21] G. Bocewicz, I. Nielsen, A. Gola, Z. Banaszak, Reference model of milk-run traffic systems prototyping, International Journal of Production Research 59 (2020) 1–18. doi:10.1080/00207543.2020.1766717.

[22] P. Sitek, J. Wikarek, K. Rutczynska-Wdowiak, G. Bocewicz, Z. Banaszak, Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: a modified hybrid approach, Neurocomputing 423 (2020). doi:10.1016/j.neucom.2020.02.126.

[23] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: A categorical literature review and classification, IFAC-PapersOnLine 51 (2018) 1016–1022. doi:https://doi.org/10.1016/j.ifacol.2018.08.474, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[24] J. J. Collins, J. B. McVicar, D. Wedlock, R. Brown, D. Howard, J. Leitner, Benchmarking simulated robotic manipulation through a real world dataset, IEEE Robotics and Automation Letters 5 (2020) 250–257.

[25] T. Sun, L. Gong, X. Li, S. Xie, Z. Chen, Q. Hu, D. Filliat, Robotdrlsim: A real time robot simulation platform for reinforcement learning and human interactive demonstration learning, Journal of Physics: Conference Series 1746 (2021) 012035. doi:10.1088/1742-6596/1746/1/012035.

[26] X. Yang, Z. Ji, J. Wu, Y.-K. Lai, An open-source multi-goal reinforcement

learning environment for robotic manipulation with pybullet, 2021. doi:10.48550/ARXIV.2105.05985.

[27] R. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, C. Schmid, Learning to combine primitive skills: A step towards versatile robotic manipulation (2019). doi:10.48550/ARXIV.1908.00722.

[28] R. Featherstone, The calculation of robot dynamics using articulated-body inertias, The international journal of robotics research 2 (1983) 13–30.

[29] C. Brooks, K. Iagnemma, Vibration-based terrain classification for planetary exploration rovers, Robotics, IEEE Transactions on 21 (2006) 1185 – 1191. doi:10.1109/TRO.2005.855994.

[30] C. A. Brooks, K. Iagnemma, Self-supervised terrain classification for planetary surface exploration rovers, Journal of Field Robotics 29 (2012) 445–468. doi:https://doi.org/10.1002/rob.21408.

[31] P. Giguère, G. Dudek, A simple tactile probe for surface identification by mobile robots, Robotics, IEEE Transactions on 27 (2011) 534 – 544. doi:10.1109/TRO.2011.2119910.

[32] E. Krotkov, Active perception for legged locomotion: every step is an experiment (1990) 227–232 vol.1. doi:10.1109/ISIC.1990.128462.

[33] L. Ding, H. Gao, Z. Deng, J. Song, Y. Liu, G. Liu, K. Iagnemma, Foot–terrain interaction mechanics for legged robots: Modeling and experimental validation, The International Journal of Robotics Research 32 (2013) 1585–1606. doi:10.1177/0278364913498122.

[34] M. A. Hoepflinger, C. D. Remy, M. Hutter, S. Haag, R. Siegwart, Haptic terrain classification on natural terrains for legged robots, Emerging Trends in Mobile Robotics- Proceedings of the 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2010 (2010) 785–792. doi:10.1142/9789814329927_0097.

[35] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, M. Hutter, Where should i walk? predicting terrain properties from images via self-supervised learning, IEEE Robotics and Automation Letters 4 (2019) 1509–1516.

[36] K. Walas, Terrain Classification and Negotiation with a Walking Robot, Journal of Intelligent and Robotic Systems: Theory and Applications 78 (2015) 401–423. doi:10.1007/s10846-014-0067-0.

[37] X. A. Wu, T. M. Huh, A. Sabin, S. A. Suresh, M. R. Cutkosky, Tactile sensing and terrain-based gait control for small legged robots, IEEE Transactions on Robotics 36 (2020) 15–27. doi:10.1109/TRO.2019.2935336.

[38] E. Tennakoon, T. Peynot, J. Roberts, N. Kottege, Probe-before-step walking strategy for multi-legged robots on terrain with risk of collapse, Proceedings - IEEE International Conference on Robotics and Automation (2020) 5530–5536. doi:10.1109/ICRA40945.2020.9197154.

[39] H. Kolvenbach, G. Valsecchi, R. Grandia, A. Ruiz, F. Jenelten, M. Hutter, Tactile Inspection of Concrete Deterioration in Sewers with Legged Robots, Field and Service Robots (FSR) (2019) 14.

[40] H. Kolvenbach, C. Bartschi, L. Wellhausen, R. Grandia, M. Hutter, Haptic inspection of planetary soils with legged robots, IEEE Robotics and Automation Letters 4 (2019) 1626–1632. doi:10.1109/LRA.2019.2896732.

[41] W. Bosworth, J. Whitney, S. Kim, N. Hogan, Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ (2016) 3582–3589. doi:10.1109/ICRA.2016.7487541.

[42] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, C. Semini, Stance: Locomotion adaptation over soft terrain, IEEE Transactions on Robotics 36 (2020) 443–457. doi:10.1109/TRO.2019.2954670.

[43] G. Haddeler, H. P. Palanivelu, Y. C. Ng, F. Colonnier, A. H. Adiwahono, Z. Li, C.-M. Chew, M. Y. Chuah, Real-time digital double framework to predict collapsible terrains for legged robots (2022) 10387–10394. doi:10.1109/IROS47612.2022.9981613.

[44] Unitree Robotics, 2023. URL: http://www.unitree.com.

[45] Kinovea, 2023. URL: https://www.kinovea.org/.