

# Simultaneous State Estimation and Contact Detection for Legged Robots by Multiple-Model Kalman Filtering

Menner, Marcel; Berntorp, Karl

TR2024-085 June 27, 2024

## Abstract

This paper proposes an algorithm for combined contact detection and state estimation for legged robots. The algorithm models the robot's movement as a switched system, where different modes relate to different feet being in contact with the ground. The key element of our algorithm is an interacting multiple-model Kalman filter, which identifies the currently-active mode defining contacts, while estimating the state. The rationale for the proposed estimation framework is that contacts (and contact forces) impact the robot's state, and vice versa. We present validation studies with a quadruped using (i) the high-fidelity simulator Gazebo for a comparison with ground truth values and a baseline estimator, and (ii) hardware experiments with the Unitree A1 robot. The simulation study shows that the proposed algorithm outperforms the baseline estimator, which does not simultaneously detect contacts. The hardware experiments showcase the applicability of the proposed algorithm and highlights the ability to detect contacts.

*European Control Conference (ECC) 2024*

© 2024 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Simultaneous State Estimation and Contact Detection for Legged Robots by Multiple-Model Kalman Filtering

Marcel Menner and Karl Berntorp

**Abstract**—This paper proposes an algorithm for combined contact detection and state estimation for legged robots. The algorithm models the robot’s movement as a switched system, where different modes relate to different feet being in contact with the ground. The key element of our algorithm is an interacting multiple-model Kalman filter, which identifies the currently-active mode defining contacts, while estimating the state. The rationale for the proposed estimation framework is that contacts (and contact forces) impact the robot’s state, and vice versa. We present validation studies with a quadruped using (i) the high-fidelity simulator Gazebo for a comparison with ground truth values and a baseline estimator, and (ii) hardware experiments with the Unitree A1 robot. The simulation study shows that the proposed algorithm outperforms the baseline estimator, which does not simultaneously detect contacts. The hardware experiments showcase the applicability of the proposed algorithm and highlights the ability to detect contacts.

## I. INTRODUCTION

Mobile robots are useful for a variety of tasks to increase automation or provide assistance, e.g., robots have been employed for warehouse logistics, search and rescue, load carrying, etc. Wheeled robots have proven to be effective in more structured environments such as warehouses, but mobile robots that move by using legs provide a higher level of versatility and the potential to maneuver in more unstructured environments. However, this higher level of versatility comes at the expense of greater difficulty to achieve accurate and safe movements, i.e., the legs have to be moved to simultaneously traverse an area, stabilize the robot, and avoid unwanted contacts with the environment. To fully reap the benefits of a legged robot’s versatility in unstructured environments, fast control and estimation algorithms are needed that target such environments.

This paper focuses on state estimation and contact detection. One possible path to achieving accurate state estimation for control is to introduce additional sensors in the robot’s design, but this is costly and more sensitive to failures in the additional sensors. Instead, we present a software-based solution with an algorithm for state estimation that only uses inertial measurement unit (IMU) sensor measurements (Euler angles, rotations, linear accelerations) and motor measurements (joint angles, joint accelerations, motor torques). Such sensor measurements are commonly available on robotic platforms such as legged robots.

We propose a method for simultaneous contact detection and state estimation for legged robots based on a physics-based motion model to reason about the contact forces’

impact on the robot. The proposed method uses an interacting multiple-model Kalman filter (IMM-KF), where the robot’s movement is modeled as a switched system. The system relates the multiple models to different feet being in contact with the ground. The IMM-KF identifies the currently-active mode defining contacts, while estimating the state.

In [1], [2], a stochastic filtering approach with outlier rejection is presented, which fuses leg kinematics with inertial measurements, and [3] develops a probabilistic framework for detecting contacts. In [4], a disturbance observer-based approach is proposed for contact detection with an event-based finite state machine. A factor graph optimization method is presented in [5] for state estimation fusing inertial navigation, leg odometry, and visual odometry, and [6] fuses IMU and leg odometry sensing for pose and velocity estimation by relying on force/torque sensors at the feet using the Schmitt trigger mechanism for contact classification. In [7], IMU measurements are fused with leg odometry and studies the impact of soft terrain on state estimation, and [8] proposes a state estimator for slippery environments, which fuses inertial and velocity measurements from a tracking camera and leg kinematics; [9] proposes a contact-aided invariant extended Kalman filter; [10] uses body speed measurements obtained from a stereo camera without the need for a non-slip assumption; and [11] proposes a state estimator for legged robots based on learned displacement measurement using convolutional neural networks. While current solutions often rely on a camera, which is susceptible to external conditions such as environmental/lighting, or the availability of contact sensors, our proposed method only uses IMU measurements for simultaneous state estimation and contact detection.

*Notation.* This paper uses boldface  $\mathbf{x}$  and boldface  $\mathbf{X}$  to indicate a vector and matrix, respectively. This paper uses parenthesis,  $\mathbf{x}(t)$ , and sub-scripts,  $\mathbf{x}_t$ , to distinguish the state in continuous time and discrete time  $t$ , respectively. Let  $\Pr(\mathcal{X})$  be the probability of an event  $\mathcal{X}$ , e.g.,  $\Pr(\delta = 1)$  is the probability that  $\delta = 1$ .  $\mathbf{P}_{\text{skew}}(\mathbf{p}) \in \mathbf{R}^{3 \times 3}$  is the skew matrix of a vector  $\mathbf{p} \in \mathbf{R}^3$  such that  $\mathbf{P}_{\text{skew}}(\mathbf{p})\mathbf{f} = \mathbf{p} \times \mathbf{f}$  with the cross product  $\times$ . The notation  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  means  $\mathbf{x}$  sampled from a normal distribution,  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ . Throughout, super-scripts  $(\cdot)^{\text{bf}}$  and  $(\cdot)^{\text{wf}}$  indicate body-frame and world-frame, respectively. Further,  $\bar{x}$  defines a measurement and  $\hat{x}$  defines an estimate or computed value of a quantity  $x$ . The rotation matrix from world-frame to body-frame is denoted as  $\mathbf{R}_{\text{wf} \rightarrow \text{bf}}(\boldsymbol{\Theta}) \in \mathbf{R}^{3 \times 3}$  as a function of the Euler angles  $\boldsymbol{\Theta} \in \mathbf{R}^3$ . Let  $\mathbf{0}$  denote a zero matrix of zero vector of appropriate dimension. Let  $\mathbf{I}$  denote an identity matrix of appropriate dimension.

## II. INTERACTING MULTIPLE-MODEL KALMAN FILTER

An IMM-KF is a state estimator for a switched system

$$\mathbf{x}_{t+1} = \mathbf{A}_t^{(k)} \mathbf{x}_t + \mathbf{b}_t^{(k)} + \mathbf{w}_t^{(k)} \quad (1a)$$

$$\mathbf{y}_t = \mathbf{C}_t^{(k)} \mathbf{x}_t + \mathbf{v}_t^{(k)}, \quad (1b)$$

where  $k \in \{1, \dots, M\}$  denotes the  $k^{\text{th}}$  of the  $M$  modes. The switched system's modes are modeled as a Markov chain with with transition probabilities

$$\Pr(m_{t+1} = m^{(j)} | m_t = m^{(i)}) = \pi_{ij}, \quad (1c)$$

where  $m_t \in \mathcal{M}$  is the mode at time  $t$ , and the transition from mode  $m^{(i)}$  to mode  $m^{(j)}$  being  $\pi_{ij}$ . The IMM-KF involves an interaction, a filtering, a probability update, and a combination step [12].

1) *Interaction step*: In the interaction step, the estimates of  $M$  KFs are mixed and used to initialize each filter:

$$c^{(k)} = \sum_{j=1}^M \pi_{jk} \mu_{t-1}^{(j)} \quad (2a)$$

$$\mu_{t-1|t-1}^{(j|k)} = \frac{\pi_{jk} \mu_{t-1}^{(j)}}{c^{(k)}} \quad (2b)$$

$$\bar{\mathbf{x}}_{t-1|t-1}^{(k)} = \sum_{j=1}^M \mu_{t-1|t-1}^{(j|k)} \hat{\mathbf{x}}_{t-1|t-1}^{(j)} \quad (2c)$$

$$\bar{\mathbf{P}}_{t-1|t-1}^{(k)} = \sum_{j=1}^M \mu_{t-1|t-1}^{(j|k)} \left( \mathbf{P}_{t-1|t-1}^{(j)} + \mathbf{X}_{t-1|t-1}^{(k,j)} \right) \quad (2d)$$

with  $\mathbf{X}_{t|t}^{(k,j)} = (\bar{\mathbf{x}}_{t|t}^{(k)} - \hat{\mathbf{x}}_{t|t}^{(j)}) (\bar{\mathbf{x}}_{t|t}^{(k)} - \hat{\mathbf{x}}_{t|t}^{(j)})^T$ , where  $\hat{\mathbf{x}}_{t|t}^{(k)}$  is the state estimate of filter  $k$  at time  $t$ , and  $\mu_t^{(k)}$  is the probability of filter  $k$  being active.

2) *Filtering step*: Each KF is executed separately using  $\bar{\mathbf{x}}_{t-1|t-1}^{(k)}$  and  $\bar{\mathbf{P}}_{t-1|t-1}^{(k)}$  to initialize filter  $k$ .

3) *Probability update step*: The filters' innovation residuals are used to update the filters' model probabilities:

$$L_t^{(k)} = \frac{\exp\left(-\frac{1}{2}(\tilde{\mathbf{y}}_t^{(k)})^T (\mathbf{S}_t^{(k)})^{-1} \tilde{\mathbf{y}}_t^{(k)}\right)}{\left|2\pi \mathbf{S}_t^{(k)}\right|^{0.5}} \quad (2e)$$

$$\mu_t^{(k)} = \frac{c^{(k)} L_t^{(k)}}{\sum_{i=1}^M c^{(i)} L_t^{(i)}} \quad (2f)$$

with the likelihood  $L_t^{(k)}$  and the model probability  $\mu_t^{(k)}$ .

4) *Combination step*: Lastly, the filters' state estimates are combined as a weighted sum using the model probabilities:

$$\hat{\mathbf{x}}_{t|t} = \sum_{k=1}^M \mu_t^{(k)} \hat{\mathbf{x}}_{t|t}^{(k)} \quad (2g)$$

$$\mathbf{P}_{t|t} = \sum_{k=1}^M \mu_t^{(k)} \left( \mathbf{P}_{t|t}^{(k)} + (\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t}^{(k)}) (\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t}^{(k)})^T \right). \quad (2h)$$

## III. MODE-DEPENDENT MOTION MODEL

The motion of the robot's trunk can be modeled as

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\Theta(t)) \mathbf{x}(t) + \mathbf{g} + \mathbf{b}(t) \quad (3a)$$

with the state  $\mathbf{x}(t) \in \mathbf{R}^{12}$  at time  $t$ , the gravity vector  $\mathbf{g}$ , and  $\mathbf{b}(t)$  resulting from external forces. The state consists of the orientation  $\Theta$ , the center of mass (CoM) position  $\mathbf{d}_{\text{CoM}}^{\text{wf}}$ , the rotation  $\boldsymbol{\omega}^{\text{wf}}$ , and the linear velocity  $\mathbf{v}_{\text{CoM}}^{\text{wf}}$ :

$$\mathbf{x}(t) = [\Theta(t)^\top \quad \mathbf{d}_{\text{CoM}}^{\text{wf}}(t)^\top \quad \boldsymbol{\omega}^{\text{wf}}(t)^\top \quad \mathbf{v}_{\text{CoM}}^{\text{wf}}(t)^\top]^\top. \quad (3b)$$

Hence,

$$\mathbf{A}(\Theta(t)) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{R}_{\text{wf} \rightarrow \text{bf}}(\Theta(t)) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (3c)$$

The external forces result from contacts of the robot's feet with the ground/environment, i.e.,

$$\mathbf{b}(t) = \mathbf{B}(t) \mathbf{f}^{\text{bf}}(t) \quad (4a)$$

with the force vector  $\mathbf{f}^{\text{bf}}(t)$ . For a four-legged robot,

$$\mathbf{f}^{\text{bf}}(t) = [\mathbf{f}_{\text{FL}}^{\text{bf}}(t) \quad \mathbf{f}_{\text{FR}}^{\text{bf}}(t) \quad \mathbf{f}_{\text{RL}}^{\text{bf}}(t) \quad \mathbf{f}_{\text{RR}}^{\text{bf}}(t)]^\top, \quad (4b)$$

where  $\mathbf{f}_i^{\text{bf}} = [f_{x,i}^{\text{bf}} \quad f_{y,i}^{\text{bf}} \quad f_{z,i}^{\text{bf}}]^\top \in \mathbf{R}^3$  is the foot-contact force of leg  $i$ . The reason for representing the contact forces in body-frame will become apparent in Section IV. Hence,

$$\mathbf{B}(t) = \mathbf{B}_1(t) \mathbf{B}_2(t)$$

$$\mathbf{B}_1(t) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{\text{bf} \rightarrow \text{wf}}(\Theta(t)) (\mathbf{I}^{\text{bf}})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\text{bf} \rightarrow \text{wf}}(\Theta(t)) \frac{1}{m} \end{bmatrix}$$

$$\mathbf{B}_2(t) = \begin{bmatrix} \mathbf{P}_{\text{FL}}^{\text{bf}}(t) & \mathbf{P}_{\text{FR}}^{\text{bf}}(t) & \mathbf{P}_{\text{RL}}^{\text{bf}}(t) & \mathbf{P}_{\text{RR}}^{\text{bf}}(t) \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix},$$

where  $\mathbf{B}_1(t)$  includes rotations and inertia values of the robot's trunk, and  $\mathbf{B}_2(t)$  computes linear and angular accelerations from the contact forces with

$$\mathbf{P}_i^{\text{bf}}(t) = \mathbf{P}_{\text{skew}}(\mathbf{p}_i^{\text{bf}}(t)) \in \mathbf{R}^{3 \times 3}, \quad (4c)$$

where  $\mathbf{p}_i^{\text{bf}} \in \mathbf{R}^3$  is the relative position of leg  $i$  from the trunk's CoM, which can be calculated using forward kinematics and the joint angles  $\mathbf{q}_i$ , i.e.,  $\mathbf{p}_i^{\text{bf}}(t) = \mathbf{FK}_i(\mathbf{q}_i)$ . Hence,  $\mathbf{B}_2(t) \mathbf{f}^{\text{bf}}(t)$  computes the resulting moments and forces at the CoM (in body-frame) as a combination of the forces at the feet, which are transformed to angular and linear accelerations (in world-frame) via  $\mathbf{B}_1(t)$ .

### A. Model as Switched System

As the contact forces  $\mathbf{f}_i^{\text{bf}}$  are often difficult to measure, this paper proposes to use estimated "hypothetical" forces in combination with binary contact-detection variables,  $\delta_i^k$ , with mode index  $k$ :

$$\mathbf{f}_i^{\text{bf}}(t) = \delta_i^k \hat{\mathbf{f}}_i^{\text{bf}}(t).$$

The estimated ‘‘hypothetical’’ forces,  $\hat{\mathbf{f}}_i^{\text{bf}}$ , indicate a contact force acting on the robot if the robot’s leg *would be* in contact with the ground. Hence, if the leg  $i$  is actually in contact with ground, the binary variable has to be one,  $\delta_i^k = 1$ , for the dynamics to be accurate. In turn, this observation can be leveraged to detect feet that are in contact with the ground. Hence, simultaneous contact detection and state estimation is achieved by jointly estimating  $\mathbf{x}(t)$  and  $\delta_i^k$ . Thus, the dynamics in (3a) are represented by a switched system with

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\Theta(t))\mathbf{x}(t) + \mathbf{g} + \mathbf{B}^k(t)\hat{\mathbf{f}}^{\text{bf}}(t), \quad (5)$$

$$\mathbf{B}^k(t) = \mathbf{B}_1(t)\mathbf{B}_2^k(t), \quad (6)$$

$$\mathbf{B}_2^k = \begin{bmatrix} \delta_{\text{FL}}^k \mathbf{P}_{\text{FL}}^{\text{bf}} & \delta_{\text{FR}}^k \mathbf{P}_{\text{FR}}^{\text{bf}} & \delta_{\text{RL}}^k \mathbf{P}_{\text{RL}}^{\text{bf}} & \delta_{\text{RR}}^k \mathbf{P}_{\text{RR}}^{\text{bf}} \\ \delta_{\text{FL}}^k \mathbf{I} & \delta_{\text{FR}}^k \mathbf{I} & \delta_{\text{RL}}^k \mathbf{I} & \delta_{\text{RR}}^k \mathbf{I} \end{bmatrix}. \quad (7)$$

### B. Contact Modes and Contact Probabilities

An enumeration of the modes is

$$k = 1, m^{(1)} \rightarrow \delta_{\text{FL}}^1 = 0, \delta_{\text{FR}}^1 = 0, \delta_{\text{RL}}^1 = 0, \delta_{\text{RR}}^1 = 0$$

$$k = 2, m^{(2)} \rightarrow \delta_{\text{FL}}^2 = 0, \delta_{\text{FR}}^2 = 0, \delta_{\text{RL}}^2 = 0, \delta_{\text{RR}}^2 = 1$$

⋮

$$k = 15, m^{(15)} \rightarrow \delta_{\text{FL}}^{15} = 1, \delta_{\text{FR}}^{15} = 1, \delta_{\text{RL}}^{15} = 1, \delta_{\text{RR}}^{15} = 0$$

$$k = 16, m^{(16)} \rightarrow \delta_{\text{FL}}^{16} = 1, \delta_{\text{FR}}^{16} = 1, \delta_{\text{RL}}^{16} = 1, \delta_{\text{RR}}^{16} = 1$$

with the total number of modes  $M$ . For example,  $m^{(1)}$  and  $m^{(16)}$  indicates none and all four legs are in contact with the ground, respectively. Note that for computational reasons, some less likely modes can be neglected, e.g.,  $k \in \{2, 3, 5, 9\}$  relate to only one leg being in contact with the ground, which is less likely during standard operation.

Using a probabilistic state estimator with the switched system in (5), it is further possible to indicate probabilities of leg  $i$  being in contact with the ground. Let  $\mu^k$  be the likelihood that mode  $m^{(k)}$  is active. Then, the likelihood of leg  $i$  being in contact with the ground can be calculated as

$$p_i = \Pr(\delta_i = 1) = \sum_{k=1}^M \mu^k \delta_i^k. \quad (8)$$

Our method leverages these probabilities in combination with joint angles and joint velocities to estimate the trunk’s CoM position and velocity, discussed in the following section.

## IV. ESTIMATION AND FILTER DESIGN

For state estimation, this paper assumes the availability of the following sensor measurements

- Euler angles  $\Theta$ , e.g., provided by the IMU
- Acceleration at CoM  $\bar{\mathbf{a}}^{\text{bf}}$ , e.g., provided by the IMU
- Rotations at CoM  $\bar{\boldsymbol{\omega}}^{\text{bf}}$ , e.g., provided by the IMU
- Joint angles  $\bar{\mathbf{q}}$ , e.g., provided by motors
- Joint angle velocities  $\bar{\dot{\mathbf{q}}}$ , e.g., provided by motors
- Joint torques  $\boldsymbol{\tau}$ , e.g., provided by motors

Further, this paper uses first-order Euler discretization with sampling rate  $T_s$  of the switched system in (5),

$$\mathbf{x}_{t+1} = \mathbf{x}_t + T_s \left( \mathbf{A}(\Theta_t)\mathbf{x}_t + \mathbf{g} + \mathbf{B}_t^k \hat{\mathbf{f}}_t^{\text{bf}} \right) + \mathbf{w}_t \quad (9)$$

with the process noise  $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ , where sub-script  $t$  is used to indicate discrete time.

*Remark 1:* The joint angle velocities can also be estimated if they are not directly measured.

### A. Pseudo Measurements

Without a positioning system, the robot does not provide a measurement of the trunk’s position or the velocity. However, if the feet being in contact with the ground are known, then kinematic relationships can be leveraged to provide a ‘‘pseudo measurement’’ for the trunk’s position and velocity. Hence, detecting contacts while estimating the state enables the use of such pseudo measurements.

In order to obtain pseudo measurements for the robot’s trunk velocities, the mode-probabilities in (8) can be leveraged. Let  $\mathbf{v}_i^{\text{bf}} = \mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i$  be the foot-velocity of leg  $i$  in reference to the robot’s CoM (in body-frame). Then, if leg  $i$  is in contact with the ground (and not slipping),  $\mathbf{v}_i^{\text{wf}} = \mathbf{0}$ , which implies  $\mathbf{v}_{\text{CoM}}^{\text{wf}} = -\mathbf{R}_{\text{bf} \rightarrow \text{wf}} \mathbf{v}_i^{\text{bf}}$ . Hence, pseudo measurements can be obtained by considering contacts and footstep velocities. In this paper, the probabilities in (8) are used to obtain such pseudo measurements as

$$\hat{\mathbf{v}}_{\text{CoM}}^{\text{wf}} = -\frac{1}{\sum_{i=1}^4 h(p_i)} \sum_{i=1}^4 h(p_i) \mathbf{R}_{\text{bf} \rightarrow \text{wf}}(\hat{\Theta}) \hat{\mathbf{v}}_i^{\text{bf}} \quad (10a)$$

with  $\hat{\mathbf{v}}_i^{\text{bf}} = \mathbf{J}_i(\hat{\mathbf{q}}_i)\hat{\dot{\mathbf{q}}}_i$ , where  $h$  is a function that computes a weight based on the contact probability  $p_i$ . This paper uses  $h(p_i) = \max(p_i - \bar{p}, 0)$  with the threshold value  $\bar{p} = 60\%$ . The reason for choosing such a threshold value is the exclusion of legs that are likely not in contact with the ground for computing the pseudo measurements.

Similarly, for obtaining a position measurement,

$$\hat{\mathbf{p}}_{\text{CoM}}^{\text{wf}} = -\frac{1}{\sum_{i=1}^4 h(p_i)} \sum_{i=1}^4 h(p_i) \mathbf{R}_{\text{bf} \rightarrow \text{wf}}(\hat{\Theta}) \hat{\mathbf{p}}_i^{\text{bf}} \quad (10b)$$

with  $\hat{\mathbf{p}}_i^{\text{bf}} = \mathbf{FK}_i(\hat{\mathbf{q}}_i)$ .

*Remark 2:* The advantage of using (10b) is that the robot’s vertical position,  $p_{z, \text{CoM}}^{\text{wf}}$ , is estimated in reference to the ground rather than in global coordinates, which is useful for control algorithms. The horizontal position defined by  $p_{x, \text{CoM}}^{\text{wf}}$  and  $p_{y, \text{CoM}}^{\text{wf}}$  is estimated in reference to the footstep positions, i.e., if  $p_{x, \text{CoM}}^{\text{wf}} = p_{y, \text{CoM}}^{\text{wf}} = 0$ , then the robot’s CoM is perfectly in the geometric center of the feet, which is also useful for control.

### B. Hypothetical Forces

For the four-legged robot configuration considered in this paper, the hypothetical forces  $\hat{\mathbf{f}}_i^{\text{bf}}$  can be approximated using using the foot Jacobian  $\mathbf{J}_i$  with the joint angles  $\mathbf{q}_i$  and the motor torques  $\boldsymbol{\tau}_i$  of leg  $i$ :

$$\hat{\mathbf{f}}_{i,t}^{\text{bf}} = (\mathbf{J}_i(\mathbf{q}_{i,t}))^{-1} \boldsymbol{\tau}_{i,t}, \quad (11)$$

which is a common practice for legged robots, see, e.g., [13], [14] where this relationship is used for a stance controller. Note that during standard operation,  $\mathbf{J}_i$  is invertible for a leg with three joints/motors.

### C. Measurement Equation

The estimation algorithm in this paper uses the measurements  $\mathbf{y}_t \in \mathbf{R}^{15}$  with

$$\mathbf{y}_t = \mathbf{C}(\Theta_t)\mathbf{x}_t + \mathbf{D}^k \hat{\mathbf{f}}_t^{\text{bf}} + \mathbf{v}_t \quad (12)$$

with the sensor noise  $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ ,

$$\mathbf{y}_t = [\hat{\Theta}_t \quad \hat{\mathbf{p}}_t^{\text{wf}} \quad \hat{\omega}_t^{\text{bf}} \quad \hat{\mathbf{v}}_t^{\text{wf}} \quad \hat{\mathbf{a}}_t^{\text{bf}}]^\top, \\ \mathbf{C}(\Theta_t) = \begin{bmatrix} \text{blkdiag}(\mathbf{I}, \mathbf{I}, \mathbf{R}_{\text{wf} \rightarrow \text{bf}}(\Theta_t), \mathbf{I}) & \\ & \mathbf{0} & & \dots \end{bmatrix}$$

and

$$\mathbf{D}^k = \begin{bmatrix} \mathbf{0} & \dots \\ \vdots & \ddots \\ \delta_{\text{FL}}^k \frac{1}{m} \mathbf{I} & \delta_{\text{FR}}^k \frac{1}{m} \mathbf{I} & \delta_{\text{RL}}^k \frac{1}{m} \mathbf{I} & \delta_{\text{RR}}^k \frac{1}{m} \mathbf{I} \end{bmatrix}$$

The reason for  $\mathbf{D}^k$  being mode dependent is that measured accelerations at the CoM are the sum of the contact forces with  $\delta_i^k = 1$ , i.e., of feet in contact with the ground. While the acceleration measurements in this setting do not directly impact the state estimate, they help in determining the ‘‘correct’’ mode of operation. Hence, an advantage of the proposed algorithm is that it can leverage acceleration measurements,  $\hat{\mathbf{a}}_t^{\text{bf}}$ , for state estimation by means of altering the modes’ probabilities.

### D. Overall Filter Design

Overall, the process model and measurement model for each Kalman filter related to mode  $m^{(k)}$  are given by

$$\hat{\mathbf{x}}_{t+1} = (\mathbf{I}_{12} + T_s \mathbf{A}(\hat{\Theta}_t)) \hat{\mathbf{x}}_t + \mathbf{g} + \mathbf{B}_t^k \hat{\mathbf{f}}_t^{\text{bf}} + \mathbf{w}_t \quad (13a)$$

$$\mathbf{y}_t = \mathbf{C}(\hat{\Theta}_t) \hat{\mathbf{x}}_t + \mathbf{D}^k \hat{\mathbf{f}}_t^{\text{bf}} + \mathbf{v}_t. \quad (13b)$$

*Remark 3 (Confidence in Pseudo Measurements):* The algorithm leverages time-varying covariance matrices  $\mathbf{Q}_t, \mathbf{R}_t$  in order to adjust the confidence in the pseudo-measurements. This paper scales the elements in  $\mathbf{R}_t$  that relate to the pseudo measurements with

$$r_{\text{scale}} = \frac{1}{1 + 100 \sum_{i=1}^4 h(p_i)}, \quad (14)$$

i.e., for higher confidence in a foot being in contact with the ground, the algorithm decreases the uncertainty associated with the measurement. Hence, the contact probabilities in (8) are useful not only to compute a pseudo measurement but also to reason about its confidence.

### E. Exploiting Physical Limits of Contact Forces for Biasing Mode Probabilities

There are two physical limits that can be exploited easily for more accurate contact detection and state estimation. First, contact forces need to be positive, because feet can push the robot up but cannot pull the robot down. Hence, if  $\hat{f}_{z,i}^{\text{wf}} < 0$ , then all modes  $k$  with  $\delta_i^k = 1$  are less likely. Second, horizontal forces are constrained by friction to be smaller (in absolute value) than  $\nu \hat{f}_{z,i}^{\text{wf}}$  with the friction coefficient  $0 \leq \nu \leq 1$ , i.e.,  $((\hat{f}_{x,i}^{\text{wf}})^2 + (\hat{f}_{y,i}^{\text{wf}})^2)^{0.5} < \nu \hat{f}_{z,i}^{\text{wf}}$ .

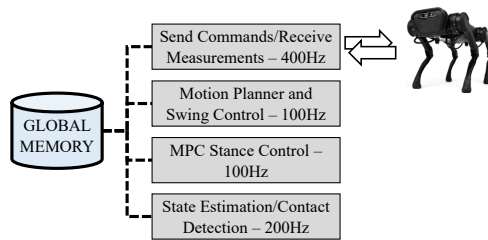


Fig. 1. Controller architecture. For the Gazebo validations, the controller publishes ROS topics that the robot uses. For the hardware experiments, the controller communicates with the robot via Ethernet.

An IMM-KF allows to incorporate such prior knowledge/physical limits by means of biasing the modes’ probabilities. Instead of (2e), this paper uses

$$L_t^{(k)} = \frac{\exp\left(-\frac{1}{2}(\hat{\mathbf{y}}_t^{(k)})^T (\mathbf{S}_t^{(k)})^{-1} \hat{\mathbf{y}}_t^{(k)}\right)}{|2\pi \mathbf{S}_t^{(k)}|^{0.5}} \exp\left(-h_f(\hat{\mathbf{f}}_t^{\text{wf}})\right) \\ h_f(\hat{\mathbf{f}}_t^{\text{bf}}) = c_{\text{force}} \sum_{i=1}^4 \delta_i^k \left(\min\left(0, \hat{f}_{z,i}^{\text{wf}}\right)\right)^2$$

with some weight  $c_{\text{force}}$ .

*Remark 4 (Alternative Formulation):* An alternative to using forces as inputs to the dynamical system in (13a) is to augment the state by contact forces. This may lead to more accurate estimation of contact forces, but comes at the expense of higher computation cost. Such a formulation with ‘‘forces as pseudo states’’ may be useful in cases, where (11) does not provide a good estimate for the forces.

## V. RESULTS

Fig. 1 summarizes the controller architecture and communication with the robot. The controller executes four modules in parallel. The four modules share a global memory, which they use to communicate their respective results. First, one module links the controller with the robot. This module sends motor commands from the global memory to the robot and receives measurements from the robot, which are subsequently written into the memory. The second module is used for motion planning and the coordination of the robot’s legs. The third module uses the state estimate stored in the global memory to compute motor commands that stabilize the robot. The state estimation and contact detection module pulls the robot’s measurements from the global memory, processes the measurements, and pushes the updated state estimate to the global memory. Note that the proposed algorithm can be utilized for (almost) any controller structure. For both simulation and hardware experiments, torque commands are sent using the robotic operating system (ROS).

For validation, we implement eight different modes. The modes  $k = 2, 4$  relate to a trot gait. The modes  $k = 3, 5, 6, 7$  relate to three feet being in contact with the ground, which are important for walking gait and for early contact detection during trot gait. The transition probabilities of the IMM-KF are chosen for trot gait being the standard mode operation. Walking gait, e.g., can be considered by changing the last row of the transition probability matrix.

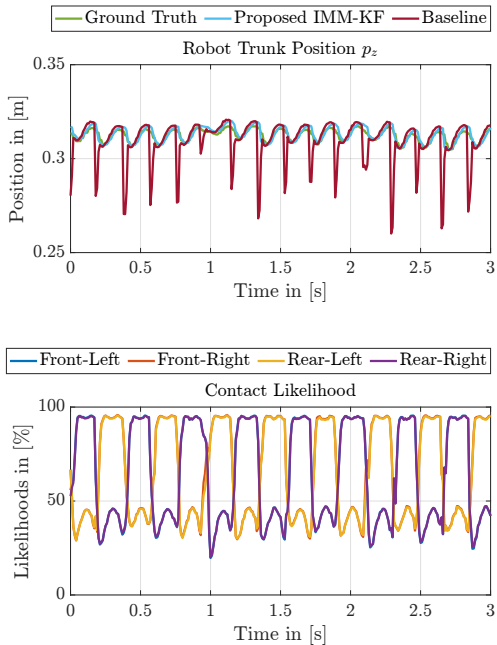


Fig. 2. Gazebo simulation results. Top: Vertical position tracking, with ground truth (green), the estimate provided by the IMM-KF (light blue), and the baseline (dark red). Bottom: Contact likelihoods of the four feet.

### A. Gazebo Simulation Results

The IMM-KF was implemented for the Unitree A1 robot [15] using Gazebo with the bullet physics engine. The Unitree A1 simulator is a realistic high-fidelity environment. The Gazebo simulation environment is executed in a parallel thread, and hence the software needs to be executed during run-time of the simulation. The proposed IMM-KF estimate is compared with a baseline Kalman filter, which uses the switched dynamics in (13a) with the modes given by the controller instead of being estimated.

Fig. 2 shows Gazebo simulation results for a 3s window. The robot is traversing with a constant forward velocity of 1m/s in trot gait. The contact likelihoods of the four feet calculated as in (8) indicate correctly that the front-left and the rear-right leg are swinging together, switching off with the front-right and rear-left. Further, the algorithm is able to quickly identify switching contact, which is highlighted in more detail in Section V-B. Fig. 2 also shows that the proposed IMM-KF solution tracks the robot’s vertical position closely, whereas the baseline spikes in between gait transitions. These spikes are due to the baseline filter not actively identifying modes, but using the controller modes in open loop. Table I quantifies the tracking error for 1min of operation for the full state, the vertical position of the robot’s CoM, and the robot’s velocities. As displayed in Fig. 2, the robot’s vertical position above ground is crucial to stabilize the robot, especially for controllers that use torque commands. Table I shows that the state tracking error is reduced by a factor of 2.5. For the vertical position, the tracking error is reduced by 7.5, and the velocity tracking error is reduced by 3.75. This highlights the benefit of simultaneously estimating the state and detecting contacts.

TABLE I  
STATE TRACKING: ROOT MEAN SQUARE ERROR

		IMM-KF	Baseline
Full State	RMSE	0.0952	0.2438
Vertical Position	RMSE	0.17cm	1.25cm
	Maximum	0.88cm	6.36cm
Velocities	RMSE	0.1195m/s	0.4395m/s

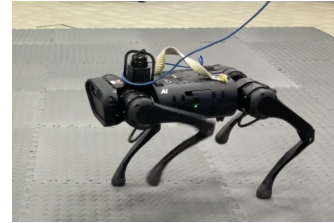


Fig. 3. Hardware experiments with the Unitree A1.

### B. Hardware Experiments

Fig. 3 shows the Unitree A1 robot. Fig. 4 shows the contact likelihoods,  $\Pr(\delta_i = 1)$  computed as in (8) for the four legs. It shows the robot taking four steps, which can be identified by the vertical foot positions in the bottom plot, which were computed using forward kinematics. First, the front-right and rear-left legs swing together around 0.2s. Second, the front-left and rear-right legs swing together around 1.2s. At 2.5s, two trotting motions are executed back-to-back. During a swing, the contact likelihoods of the swinging legs are below 40%, whereas the likelihoods of the stance legs are around 95%. When all four legs are in contact with the ground, all contact likelihoods are around 75%. The reasons for higher likelihoods of the stance legs during a swing execution are (i) higher contact forces associated with the two stance legs compared to a situation with all four legs being in contact with the ground and (ii) the physics-based biasing of probabilities of swinging legs (see Sec. IV-E).

Fig. 5 highlights a moment in time with an early contact. Here, the front-right foot is touching down earlier than the rear-left foot by around 2cm or around 20ms, which can best be seen by the vertical position plot. Consequently, the contact likelihood associated with the front-right leg increase faster than that of the rear-left leg. Fig. 6 illustrates the mode likelihoods,  $\mu^{(k)}$ , which are used within the IMM-KF. It shows that mode  $m^{(6)}$  peaks briefly at around 2.7–2.75s identifying correctly that the front-right foot has touched down early. It can be seen that contacts are detected using the proposed IMM-KF in less than 20ms, i.e., four filter recursions at an execution rate of 200Hz.

1) *Computation Times:* Computation times average 2.1ms (mean), 2.0ms (median), and showed a maximum computation time of 3.5ms, which is below the sampling frequency of 5ms (200Hz). Such high frequencies are required for contact detection in order to quickly identify feet that have touched down. The algorithm was executed on a desktop computer using ROS in Python 2.7 with AMD 5950X, DDR4 2933 32GB RAM, and NVIDIA GeForce GTX TITAN X.

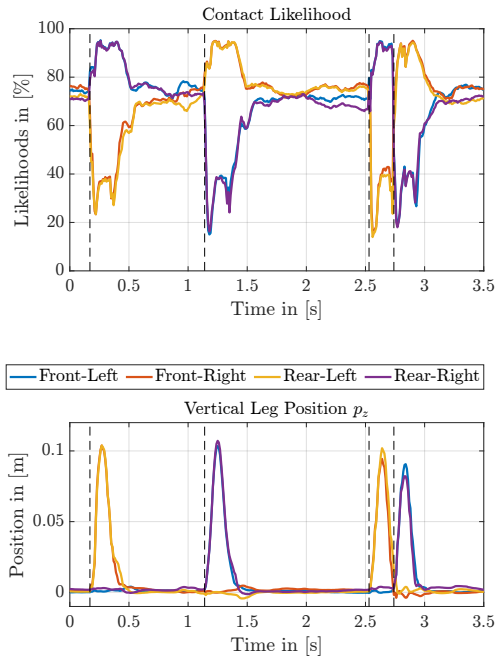


Fig. 4. Hardware experiment results. Top: Contact likelihoods of the four feet. Bottom: Feet positions calculated using forward kinematics.

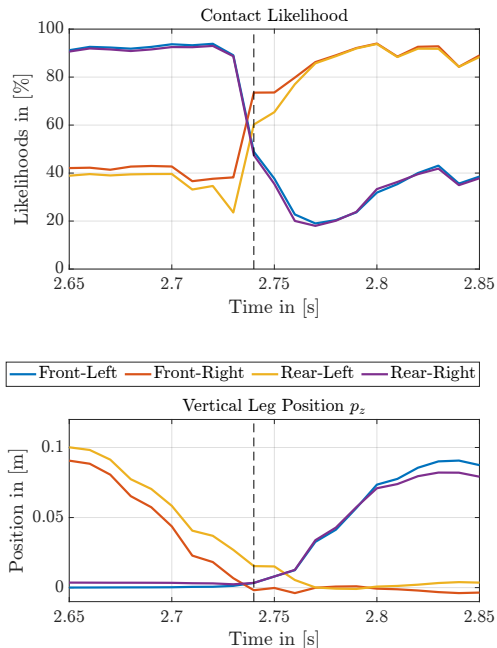


Fig. 5. Early contact detection in hardware experiment. Top: Contact likelihoods. Bottom: Feet positions calculated using forward kinematics.

## VI. CONCLUSION

We proposed a model-based algorithm for simultaneous contact detection and state estimation. The algorithm used an interacting multiple-model KF, where each filter model relates to a configuration of feet being in contact with the ground. Simulation results using the high-fidelity environment Gazebo showed an improved accuracy of state estimation by a factor of 2.5 compared to a KF without simultaneous contact detection. Further, hardware experiments showed that the algorithm is able to estimate the state accurately and in real time. In particular, it showed that contacts can correctly be identified with only a few KF

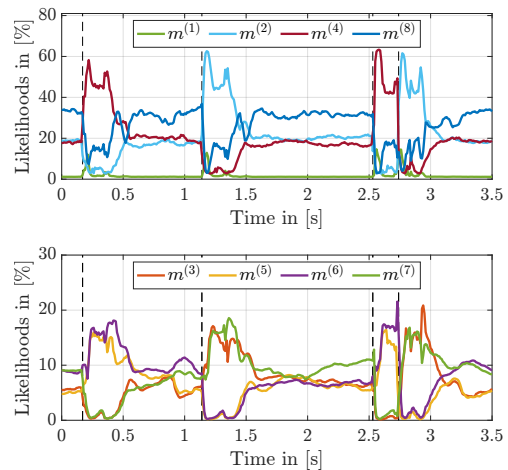


Fig. 6. Likelihoods of modes. Top: Mode  $m^{(1)}$  (no feet in contact), mode  $m^{(8)}$  (four feet in contact), and trot modes  $m^{(2)}$ ,  $m^{(4)}$  (two feet in contact). Bottom: Modes  $m^{(3)}$ ,  $m^{(5)}$ ,  $m^{(6)}$ ,  $m^{(7)}$  (three feet in contact).

recursions in less than 20ms.

## REFERENCES

- [1] M. Bloesch, M. Hutter, M. A. Hoepfinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013.
- [2] M. Bloesch, *State estimation for legged robots-kinematics, inertial sensing, and computer vision*. PhD thesis, ETH Zurich, 2017.
- [3] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, 2017.
- [4] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE Int. Conf. Robot. Autom.*, pp. 4399–4406, 2018.
- [5] D. Wisht, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.
- [6] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Frontiers in Robotics and AI*, vol. 7, p. 68, 2020.
- [7] S. Fahmi, G. Fink, and C. Semini, "On state estimation for legged locomotion over soft terrain," *IEEE Sensors Letters*, vol. 5, no. 1, pp. 1–4, 2021.
- [8] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended Kalman filter with velocity update," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3104–3110, 2021.
- [9] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Legged robot state estimation using invariant Kalman filtering and learned contact events," *arXiv preprint arXiv:2106.15713*, 2021.
- [10] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "STEP: state estimator for legged robots using a preintegrated foot velocity factor," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4456–4463, 2022.
- [11] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," in *Conference on Robot Learning*, pp. 1575–1584, 2022.
- [12] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger, "Interaction-aware motion prediction for autonomous driving: A multiple model Kalman filtering scheme," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 80–87, 2020.
- [13] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1–9, 2018.
- [14] A. Schperberg, S. Di Cairano, and M. Menner, "Auto-tuning of controller and online trajectory planner for legged robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7802–7809, 2022.
- [15] Uniret Robotics, "GitHub repository for Uniret A1," <https://github.com/uniretrobotics>, accessed 2021-09.