

Kinematic Batch Calibration for Legged Robots

Michael Bloesch, Marco Hutter, Christian Gehring, Mark A. Hoepflinger and Roland Siegwart
Autonomous Systems Lab, ETH Zürich, Switzerland, bloeschm@ethz.ch

Abstract—This paper introduces a novel batch optimization based calibration framework for legged robots. Given a non-degenerate calibration dataset and considering the stochastic models of the sensors, the task is formulated as a maximum likelihood problem. In order to facilitate the derivation of consistent measurement equations, the trajectory of the robot and other auxiliary variables are included into the optimization problem. This formulation can be transformed into a nonlinear least squares problem which can be readily solved. Applied to our legged robot StarLETH, the framework estimates kinematic parameters (segment lengths, body dimensions, angular offsets), accelerometer and gyroscope biases, as well as full inter-sensor calibrations. The generic structure easily allows the inclusion of additional sensor modalities. Based on datasets obtained on the real robot the consistency and performance of the presented approach are successfully evaluated.

I. INTRODUCTION

While the current progress in actuation schemes, sensor setups, and mechanical design allows the construction of increasingly performing legged robots, motion planning, control, and state estimation of such systems still pose a very challenging problem. The present work contributes to the ongoing research by focusing on the calibration of the kinematics and the sensor setup of a legged platform.

Most real robotic devices face the problem of calibration and many different research communities have studied the issue. The simplest approach for calibration is to measure the calibration parameters by hand or with the help of some external tools. However, this is not always possible, can be a very tedious work, and might lack the desired precision. Nowadays, calibration is often done by using sensor models and fitting the model parameters by minimizing the measurement errors for a given dataset. In some cases this can be done without explicitly evaluating the state of the robot (e.g., if one of the sensor models is invertible). Many hand-eye or head-eye calibration problems can be solved explicitly [11, 16]. They are often solved by looking at kinematic chains, where the camera images are transformed into kinematic quantities.

The kinematic chain method is the most common approach in kinematic calibration. For the well-studied hand-eye or head-eye calibration problems the kinematic chains are closed through the visual measurements. For setups without cameras, different means have been used, e.g., Baser et al. [1] measured the distance of the end-effector to a fixed point and used a standard nonlinear least squares solver to find the kinematic calibration parameters. An extensive analysis of the existing methods is given by Khalil et al. [9].

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics.

Often the error equations and corresponding noise specification can be more accurately formulated if the state of the robot is included into the calibration process. This means that not only the calibration parameters need to be estimated but the states of the robot too. This increases the dimension of the problem and makes the computations more expensive. Still, with the emergence of more elaborate filtering and optimization techniques as well as with the increase in computational power of modern hardware, this approach has become more and more common. Two classes of approaches can be distinguished here. Online approaches which filter and refine the estimated parameters at runtime and batch approaches which use a prerecorded dataset over a given time period.

Online approaches have the advantage that the setup can be used at startup and that they can handle parameters that might potentially change, e.g., bias values of an inertial measurement unit (IMU). Mirzaei et al. [12] implemented one of the first online approach for extrinsic IMU-camera calibration and IMU bias estimation. It relied on an external calibration target for easy feature detection and matching. Using nonlinear observability analysis they show that the desired quantities are observable if the platform is rotated about at least two different axes. Scandaroli et al. [14] tackle the same task by implementing a nonlinear observer and proving its convergence. They avoid any calibration pattern by producing camera pose estimates directly based on the pixel intensities. Kelly et al. [8] presented a simultaneous localization and mapping (SLAM) framework which performs IMU-camera calibration (extrinsic and bias) at runtime.

In contrast to the above online calibration frameworks, batch approaches solve the full nonlinear optimization problem and thus yield a higher level of accuracy. In the case of slowly drifting calibration parameters the batch calibration can be performed repeatedly. Pradeep et al. [13] presented a general calibration framework for sensors which were able to measure 3D point targets (kinematic, camera, laser rangefinder). They solve the corresponding nonlinear least squares problem over robot states and calibration parameters by using the Levenberg-Marquardt algorithm. Based on the incremental pose estimates of two 6 DoF pose sensors, Brookshire et al. [4] determined the corresponding extrinsic calibration by solving the maximum likelihood problem within the special euclidean group $SE(3)$ using unit dual quaternions. The batch optimization problems that arise here can also be solved in continuous time. To this end, Fleps, Mair, and Ruepp et al. [5] proposed a framework where the continuous trajectory is parameterized using B-splines and thus could easily handle asynchronous measurements from IMU and camera. Similarly

to the approach in the present paper, Birbach et al. [2] implemented a calibration routine for a multi-sensorial humanoid upper body. Based on the general calibration framework of Wagner et al. [18] they solve the corresponding nonlinear least squares problem while elegantly handling the fact that 3D rotations are members of the special orthogonal group $SO(3)$.

The proper handling of 3D rotations has troubled robotic researchers for a long time. Consistent approaches emerged with the method of perturbing 3D rotations by local error rotation vectors [17, 12, 8]. This was formalized by using Lie group/algebra theory: Rotations are members of the special orthogonal group (Lie group) and can be locally mapped to the tangential vector space (Lie algebra) where operations required for optimizations and filtering can be applied [15, 4]. Further in this context, Hertzberg et al. [6] introduced the boxplus (\boxplus) and boxminus (\boxminus) operators which are getting increasingly used when working with 3D rotations. In the past two years, very efficient and convenient nonlinear least squares optimization methods have been made available which implement the above findings [10, 18].

The present work employs a key idea of our earlier work on state estimation for legged robots [3]. The foothold locations are co-estimated with the rest of the robot states. This builds a link to computer vision research where the footholds are interpreted as world-fixed features and where the leg kinematics measurements represent range-bearing measurements between robot body and footholds. In this paper, however, we use this idea to present a calibration framework which estimates the kinematic parameters of the legged platform as well as calibration parameters of the robot's sensor setup. The problem is formulated as a maximum likelihood problem, where the corresponding nonlinear least squares problem can be readily solved.

The approach is evaluated on a dataset collected on our legged robot platform StarLETH [7] including kinematic measurements, inertial measurements, and 6 DoF pose measurements from a motion capture system. For this setup, the calibration framework is motivated by the unknown mounting of the pose sensor, by the joint angle offsets, by the bias affected IMU measurements, as well as by the complex effects of the ball-shaped compliant foot which shall be captured by a simple and more convenient model. The results show that, the trajectory of the robot as well as the set of calibration parameters can be accurately estimated.

The structure of the paper is as follows. In Section II the notation, the handling of rotational quantities, and the general methodology are introduced. Subsequently, Section III discusses the more specific problem of calibrating a legged robot and takes a look at the sensor modalities. Experimental setup and obtained results are presented in Section IV.

II. METHODOLOGY

A. Notation

Given a coordinate frame A and two points P and Q , ${}^A\mathbf{r}_{PQ}$ stands for the coordinates of the vector from P to Q expressed in frame A . If B is a second coordinate frame, then C_{BA}

is the rotation matrix which *transforms the coordinates* of a vector expressed in A to the corresponding coordinates in B . The same rotation can also be represented by using the unit quaternion \mathbf{q}_{BA} . For the sake of readability, A can also stand for the point associated with the coordinate frame A , e.g., we will use ${}^W\mathbf{r}_{WI}$ in order to represent the coordinates, expressed in W , of a vector between world coordinate frame W and IMU-fixed coordinate frame I . Throughout the paper, we always add a subscript k to a quantity v , if we are talking about its value at a time t_k , i.e., $v_k = v(t_k)$.

B. Handling the Special Orthogonal Group

The representation and handling of 3D rotations is still a very common source of confusion. In order to avoid this we give a brief overview on the convention employed in this paper. In order to represent 3D rotations, which are members of the special orthogonal group $SO(3)$, we employ unit quaternions. This choice is mainly motivated by numerical reasons. The set of unit quaternions is a 3D manifold and thus is locally homeomorphic to a 3D vector space. This can be exploited by looking at the exponential mapping between a 3D rotation vector, $\boldsymbol{\theta} \in \mathbb{R}^3$, which lies in the tangential space at the identity \mathbf{q}_0 , and the corresponding quaternion $\mathbf{q} \in SO(3)$:

$$\mathbf{q} = \exp(\boldsymbol{\theta}). \quad (1)$$

This mapping is surjective and thus an inverse exists, which is called the logarithm:

$$\boldsymbol{\theta} = \log(\mathbf{q}). \quad (2)$$

Both mappings have relative simple analytical representations.

Analogously to Hertzberg et al. [6], the boxplus and boxminus operators are introduced as follows:

$$\begin{aligned} \boxplus : SO(3) \times \mathbb{R}^3 &\rightarrow SO(3), \\ \mathbf{q}, \boldsymbol{\theta} &\mapsto \exp(\boldsymbol{\theta}) \otimes \mathbf{q}, \end{aligned} \quad (3)$$

and

$$\begin{aligned} \boxminus : SO(3) \times SO(3) &\rightarrow \mathbb{R}^3, \\ \mathbf{q}_1, \mathbf{q}_2 &\mapsto \log(\mathbf{q}_1 \otimes \mathbf{q}_2^{-1}). \end{aligned} \quad (4)$$

In short, the boxminus operator expresses the difference between two quaternions by returning the error rotation vector between both. On the other hand, the boxplus operator applies a small rotation, expressed by a rotation vector, onto a unit quaternion.

In a very informal and engineering like manner we introduce special differentials on unit quaternions. Given a function $\mathbf{q} : \mathbf{x} \mapsto \mathbf{q}(\mathbf{x})$ which maps from some real vector space \mathbb{R}^N to the set of unit quaternions, we define the differential

$$\left(\frac{\partial \mathbf{q}}{\partial \mathbf{x}}\right)_i := \lim_{\epsilon \rightarrow 0} \frac{\mathbf{q}(\mathbf{x} + \epsilon \mathbf{e}_i) \boxminus \mathbf{q}(\mathbf{x})}{\epsilon}, \quad i = 1, \dots, N, \quad (5)$$

and if $\mathbf{f} : \mathbf{q} \mapsto \mathbf{f}(\mathbf{q})$ is a function which maps from the set of unit quaternions to some real vector space (or to the set of unit quaternions) we define the special differential

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)_i := \lim_{\epsilon \rightarrow 0} \frac{\mathbf{f}(\mathbf{q} \boxplus \epsilon \mathbf{e}_i) - \mathbf{f}(\mathbf{q})}{\epsilon}, \quad i = 1, \dots, 3. \quad (6)$$

Let $C(\cdot)$ be the mapping between unit quaternions and corresponding rotation matrices, then following identities hold:

$$\frac{\partial}{\partial \mathbf{q}} (\mathbf{C}(\mathbf{q})\mathbf{v}) = -(\mathbf{C}(\mathbf{q})\mathbf{v})^\times, \quad (7)$$

$$\frac{\partial}{\partial \mathbf{q}} (\mathbf{q}^{-1}) = -\mathbf{C}^T(\mathbf{q}), \quad (8)$$

$$\frac{\partial}{\partial \mathbf{q}_1} (\mathbf{q}_1 \otimes \mathbf{q}_2) = \mathbf{I}, \quad (9)$$

$$\frac{\partial}{\partial \mathbf{q}_2} (\mathbf{q}_1 \otimes \mathbf{q}_2) = \mathbf{C}(\mathbf{q}_1), \quad (10)$$

$$\frac{\partial}{\partial \mathbf{q}} \log(\mathbf{q}) = \mathbf{A}(\mathbf{q}), \quad (11)$$

where the subscript \times is used to denote the skew-symmetric matrix of a vector and where, similarly to Rodrigues' formula for rotation matrices, there exists a simple expression for $\mathbf{A}(\cdot)$. The above identities strongly simplify the evaluation of analytical Jacobians. It can be proved that the chain rule is valid. Please note that the precise formulation of the identities can vary slightly depending on the employed conventions.

C. Formulation of the Problem

The proposed calibration approach is based on batch optimization. In this sense, model fitting is performed over a dataset of all sensor readings for a given time period. We search the set of model parameters that best explains the observed sensor outputs by stating a maximum likelihood problem. By assuming independent discrete Gaussian noise this is equivalent to a nonlinear least squares problem. In order to allow for general sensor measurement equations we include the trajectory of the robot's states into the optimization problem. This is done by introducing N_S discrete states $\{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{N_S}\}$ at times $\{t_1, \dots, t_{N_S}\}$ distributed over the length of the dataset.

Based on the model of each sensor, a corresponding measurement equation can be defined. It is a function of the robot's states and of the sensor's calibration parameters. In some cases, additional quantities will be co-estimated (see Section III-B). For the sake of readability, quantities that are included as variable into the optimization problem are denoted with a *hat*, and measurement quantities with a *tilde*. Also, we stack all optimization variables (robot states, calibration parameters, and others) into one element $\hat{\mathbf{X}} \in \mathcal{X}$, where \mathcal{X} is the Cartesian product of unit quaternion sets and real vector spaces of the corresponding variables. The N_M measurement equations are now formulated explicitly for the measurement noise, i.e., for the j^{th} measurement we model

$$\mathbf{n}_j = \boldsymbol{\epsilon}_j(\hat{\mathbf{X}}, \tilde{\mathbf{z}}_j), \quad (12)$$

$$\mathbf{n}_j \sim \mathcal{N}(0, \mathbf{R}_j), \quad (13)$$

with the measurement $\tilde{\mathbf{z}}_j$ and the symmetric positive definite covariance matrix \mathbf{R}_j . In most cases, the later can be obtained from the sensors' specification. This represents a more general form of the common additive noise model

$$\tilde{\mathbf{z}}_j = \mathbf{f}_j(\hat{\mathbf{X}}) + \mathbf{n}_j, \quad (14)$$

and allows a direct formulation of the corresponding nonlinear least squares problem resulting from the likelihood maximization:

$$\hat{\mathbf{X}}^* = \arg \min_{\hat{\mathbf{X}} \in \mathcal{X}} \sum_{j=1}^{N_M} \boldsymbol{\epsilon}_j^T(\hat{\mathbf{X}}, \tilde{\mathbf{z}}_j) \mathbf{R}_j^{-1} \boldsymbol{\epsilon}_j(\hat{\mathbf{X}}, \tilde{\mathbf{z}}_j). \quad (15)$$

If we stack all measurements $\tilde{\mathbf{z}}_j$ into one quantity $\tilde{\mathbf{Z}}$ and decompose the symmetric positive definite matrix $\mathbf{R}_j^{-1} = \mathbf{Q}_j^T \mathbf{Q}_j$ we can write

$$\mathbf{E}(\hat{\mathbf{X}}, \tilde{\mathbf{Z}}) := \begin{bmatrix} \mathbf{Q}_1 \boldsymbol{\epsilon}_1(\hat{\mathbf{X}}, \tilde{\mathbf{z}}_1) \\ \vdots \\ \mathbf{Q}_{N_M} \boldsymbol{\epsilon}_{N_M}(\hat{\mathbf{X}}, \tilde{\mathbf{z}}_{N_M}) \end{bmatrix}. \quad (16)$$

With this the Levenberg-Marquardt algorithm can be applied directly on $\mathbf{E}(\hat{\mathbf{X}}, \tilde{\mathbf{Z}})$. If considering the special differentials of Section II-B and using the boxplus operator (3) for the iterative updates, no further care needs to be taken for calculating the Jacobians and solving the optimization problem.

III. CALIBRATION OF A LEGGED ROBOT

A. Overview of the Legged Robot

In this section we take a closer look at the different sensor modalities which are integrated in the calibration process. While different legged platforms may have differing sensor setups, we will present the methodology based on the quadruped robot StarIETH [7] and pay special attention to the formulation of the error function for the kinematics of the leg. Many existing legged platforms are equipped with IMU and kinematic sensors and thus the presented approach should be directly applicable. In addition to this, we include a 6 DoF pose sensor into the calibration process (referred to as *pose sensor*). This could be data from some vision or laser setup, or, like in our case, the measurements of an external motion capture system. Exactly the same procedure could be applied on a platform with an on-board camera tracking some visual target.

In Fig. 1, the employed coordinate frames are illustrated. Each state $\hat{\mathbf{x}}_k$ at time t_k is composed of the IMU's position $w\hat{\mathbf{r}}_{WI}(t_k)$ and attitude $\hat{\mathbf{q}}_{IW}(t_k)$. Shorter we define

$$\hat{\mathbf{x}}_k = (\hat{\mathbf{r}}_k, \hat{\mathbf{q}}_k) := (w\hat{\mathbf{r}}_{WI}(t_k), \hat{\mathbf{q}}_{IW}(t_k)). \quad (17)$$

B. Kinematic Measurements

A special aspect of legged robots is that they interact with their surrounding through intermittent ground contacts. This is exploited for the robot's locomotion and represents a valuable source of information for state estimation [3]. However, this relies on a precise calibration of the kinematic model of the robotic platform.

For a given robot with N_L legs, we assume that all internal kinematic variables $\boldsymbol{\alpha}$ can be measured and that we can write the forward kinematics of the i^{th} leg in the following form:

$${}^B \mathbf{r}_{BF_i} = \mathbf{f}_i(\tilde{\boldsymbol{\alpha}}, \hat{\mathbf{p}}_{kin}) + \mathbf{n}_{s,i}, \quad (18)$$

$$\mathbf{n}_{s,i} \sim \mathcal{N}(0, \mathbf{R}_s), \quad (19)$$

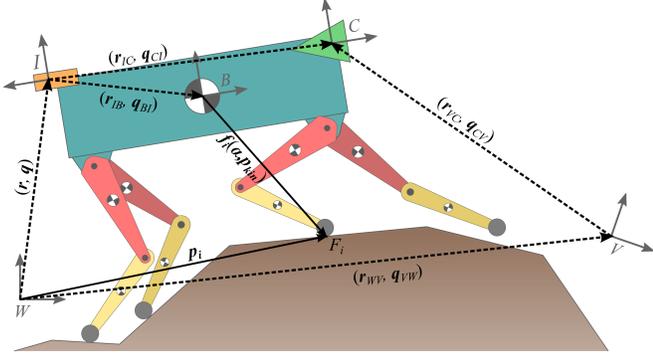


Fig. 1. Overview of coordinate frames. W : World fixed coordinate frame, I : IMU fixed coordinate frame, B : Coordinate frame fixed to the center of the body, C : Coordinate frame fixed to the pose sensor, V : Inertial coordinate frame of pose sensor

where F_i is the contact point of the i^{th} leg. The covariance matrix \mathbf{R}_s models the noise on the kinematic measurements $\tilde{\alpha}$ as well as from other inaccuracies like slippage or unknown exact contact point. Its value cannot be directly obtained from the sensors' specifications and represent the main tuning parameter of the calibration process. The vector $\hat{\mathbf{p}}_{kin}$ is the stacked vector of kinematic calibration parameters. By introducing $\hat{\mathbf{p}}_i = {}_W\hat{\mathbf{r}}_{WF_i}$ as an additional optimization variable, we can formulate the kinematic error equation as follows:

$$\epsilon_{s,i,k} = \mathbf{C}(\hat{\mathbf{q}}_{BI})(\mathbf{C}(\hat{\mathbf{q}}_k)(\hat{\mathbf{p}}_i - \hat{\mathbf{r}}_k) - I\hat{\mathbf{r}}_{IB}) - \mathbf{f}_i(\tilde{\alpha}_k, \hat{\mathbf{p}}_{kin}), \quad (20)$$

where $I\hat{\mathbf{r}}_{IB}$ and $\hat{\mathbf{q}}_{BI}$ parametrize the transformation between IMU frame and body frame.

It is important to note that the contact point of the i^{th} leg $\hat{\mathbf{p}}_i$ is introduced as an optimization variable and assumed to be stationary with respect to the world frame. The leg kinematics measurements represent range-bearing measurements between the contact point and the robot's main body. If the robot is stepping during the calibration procedure, a new contact point is introduced for every new foothold. Additional optimization variables are the calibration parameters $I\hat{\mathbf{r}}_{IB}$, $\hat{\mathbf{q}}_{BI}$, and $\hat{\mathbf{p}}_{kin}$, whereby the later includes most design parameters of a four legged robot with 3 DoF per leg and specified rotational joint sequence. If considering the symmetries of the robot, the parametrized forward kinematics of the X-Y-Y sequence of rotational angles (hip abduction, hip flexion, knee flexion) can be written as follows:

$$\mathbf{f}_i(\tilde{\alpha}, \hat{\mathbf{p}}_{kin}) = \begin{pmatrix} c_i \hat{b}_x + l_1 \\ d_i \hat{b}_y + l_2 \sin(\tilde{\alpha}_i + \Delta\hat{\alpha}_i) \\ l_2 \cos(\tilde{\alpha}_i + \Delta\hat{\alpha}_i) \end{pmatrix}, \quad (21)$$

with

$$l_1 = \hat{l}_T \sin(\tilde{\beta}_i + \Delta\hat{\beta}_i) + \hat{l}_S \sin(\tilde{\beta}_i + \Delta\hat{\beta}_i + \tilde{\gamma}_i + \Delta\hat{\gamma}_i), \\ l_2 = \hat{l}_H + \hat{l}_T \cos(\tilde{\beta}_i + \Delta\hat{\beta}_i) + \hat{l}_S \cos(\tilde{\beta}_i + \Delta\hat{\beta}_i + \tilde{\gamma}_i + \Delta\hat{\gamma}_i).$$

In the above equations, \hat{l}_H , \hat{l}_T , \hat{l}_S , are the lengths of the hip, thigh and shank segments, \hat{b}_x , \hat{b}_y are half the length and width

of the main body, $\Delta\hat{\alpha}_i$, $\Delta\hat{\beta}_i$, $\Delta\hat{\gamma}_i$ are the offset on the angular measurements of hip abduction, hip flexion and knee flexion, and $c_i, d_i \in \{1, -1\}$ are constants. We have

$$\mathbf{p}_{kin} = [\hat{l}_H, \hat{l}_T, \hat{l}_S, \hat{b}_x, \hat{b}_y, \Delta\hat{\alpha}_1, \dots, \Delta\hat{\alpha}_{N_L}, \quad (22)$$

$$\Delta\hat{\beta}_1, \dots, \Delta\hat{\beta}_{N_L}, \Delta\hat{\gamma}_1, \dots, \Delta\hat{\gamma}_{N_L}]^T. \quad (23)$$

C. Inertial Measurements

The mounted IMU contains an accelerometer and a gyroscope. The delivered measurements, $\hat{\mathbf{f}}_k$ (proper acceleration) and $\tilde{\omega}_k$ (rotational rate), are assumed to be affected by constant biases, $\hat{\mathbf{b}}_f$ and $\hat{\mathbf{b}}_\omega$ (calibration parameters), and some additive discrete Gaussian noise. Using simple finite differences the corresponding error equations can be written as

$$\epsilon_{f,k} = \frac{2\hat{\mathbf{r}}_{k-1}}{\Delta t_k(\Delta t_k + \Delta t_{k+1})} + \frac{2\hat{\mathbf{r}}_{k+1}}{\Delta t_{k+1}(\Delta t_k + \Delta t_{k+1})} - \frac{2\hat{\mathbf{r}}_k}{\Delta t_k \Delta t_{k+1}} - \mathbf{C}^T(\hat{\mathbf{q}}_k)(\hat{\mathbf{f}}_k - \hat{\mathbf{b}}_f) - \mathbf{g}, \quad (24)$$

$$\epsilon_{\omega,k} = (\hat{\mathbf{q}}_k \boxtimes \hat{\mathbf{q}}_{k-1}) / \Delta t_k + \tilde{\omega}_k - \hat{\mathbf{b}}_\omega, \quad (25)$$

with $\Delta t_k = t_k - t_{k-1}$ and

$$\epsilon_{f,k} \sim \mathcal{N}(0, \mathbf{R}_f), \quad (26)$$

$$\epsilon_{\omega,k} \sim \mathcal{N}(0, \mathbf{R}_\omega). \quad (27)$$

The convenience of the special derivatives introduced in section II-B can be illustrated here. For example the Jacobian of the error $\epsilon_{\omega,k}$ w.r.t. the unit quaternion $\hat{\mathbf{q}}_{k-1}$ can be evaluated by using equations (8), (10), and (11) together with the chain rule:

$$\frac{\partial \epsilon_{\omega,k}}{\partial \hat{\mathbf{q}}_{k-1}} = \frac{1}{\Delta t_k} \frac{\partial}{\partial \hat{\mathbf{q}}_{k-1}} \log(\hat{\mathbf{q}}_k \otimes \hat{\mathbf{q}}_{k-1}^{-1}), \quad (28)$$

$$= \frac{-1}{\Delta t_k} \mathbf{A}(\hat{\mathbf{q}}_k \otimes \hat{\mathbf{q}}_{k-1}^{-1}) \mathbf{C}(\hat{\mathbf{q}}_k) \mathbf{C}^T(\hat{\mathbf{q}}_{k-1}). \quad (29)$$

D. Pose Measurements

The pose measurements can be obtained from different kind of sensors, whereas the approach could also be extended to include other type of measurements. Here the pose sensor is measuring the translation and rotation between an inertial frame V (which in general is different from W) and the sensor coordinate frame C . Looking at the closed sequence of coordinate frame transformations $W-I-C-V-W$, we can derive the following error equations:

$$\epsilon_{q,k} = (\hat{\mathbf{q}}_{CI} \otimes \hat{\mathbf{q}}_k \otimes \hat{\mathbf{q}}_{WV}) \boxtimes \tilde{\mathbf{q}}_{CV,k}, \quad (30)$$

$$\epsilon_{r,k} = \mathbf{C}(\hat{\mathbf{q}}_{WV})_V \tilde{\mathbf{r}}_{VC,k} + {}_W\hat{\mathbf{r}}_{WV} - \hat{\mathbf{r}}_k - \mathbf{C}^T(\hat{\mathbf{q}}_k) I\hat{\mathbf{r}}_{IC}. \quad (31)$$

Based on the sensor's specifications we have:

$$\epsilon_{q,k} \sim \mathcal{N}(0, \mathbf{R}_q), \quad (32)$$

$$\epsilon_{r,k} \sim \mathcal{N}(0, \mathbf{R}_r). \quad (33)$$

The calibration quantities are the transformations between the IMU coordinate frame I and the pose sensor coordinate frame C , as well as between the world coordinate frame W and the pose sensors' reference inertial frame V . This is parametrized with the calibration parameters ${}_W\hat{\mathbf{r}}_{WV}$, $I\hat{\mathbf{r}}_{IC}$, $\hat{\mathbf{q}}_{WV}$, and $\hat{\mathbf{q}}_{CI}$.

IV. RESULTS AND DISCUSSION

The presented approach is evaluated on real datasets obtained on the quadruped robot StarLETH [7]. The motion of the robot during the recording is chosen heuristically: the robot is alternatively rolling and pitching its main body (see Fig. 3). This is mainly motivated by the experimental observation that large magnitudes of rolling and pitching make the accelerometer bias more accurate, as well as by the statement that the IMU to pose sensor calibration is only observable if the robot is rotated about at least two different axes [12]. The length of the sequences (30 s) is a compromise between accuracy of the estimates and computational costs. Potential time delay between the sensor modalities is estimated by cross-correlating rough estimates of the angular rate norms.

For the corresponding batch optimization, the Levenberg-Marquardt algorithm exhibits a good rate of convergence and a very high robustness against bad initial guesses. For example, the sequence of state estimates depicted in Fig. 2 and Fig. 3 was obtained starting with all optimization variables set to zero (or the unit quaternion). Starting with some other initial guess does in general not affect the final result. Further, if starting with a proper initial guess, which can be obtained from the robot's on-board state estimation [3], the algorithm converges in 3–4 iterations. Using covariance back-propagation an estimate of the covariances of the optimization variables can be obtained. For the robot's states the 3σ bounds are plotted in both figures. The estimated standard deviations for the calibration parameters are listed in table I.

In Fig. 4 the error residuals of the forward kinematics of one leg are displayed. While the overall error remains small, some strong autocorrelation can be observed, which violates the independent discrete Gaussian noise assumption. This originates from the compliance and the rolling of the ball-shaped foot. So far, this is handled by choosing a conservative covariance matrix \mathbf{R}_s for the kinematic measurements.

In table I the estimated parameters for two datasets based on the same calibration motion are listed (DS1 and DS2). For the calibration parameters, which could be measured or estimated by some other means (e.g. CAD drawing), the corresponding reference values are given in the last column (values with a * are only rough estimates and less accurate than the calibration results). The kinematic calibration accuracy is in the range of 1 cm for translation, 0.005 rad for 3D rotations, and about 0.02 rad for the angular calibration of the joints. If comparing both datasets with the reference values it can be observed that some estimates are biased. In particular, the parameter ${}^I\hat{r}_{IB,x}$ exhibits a constant offset of roughly 3 cm. This is probably caused by the violation of the independent Gaussian noise assumption and inaccuracies of the forward kinematics. The shank length, \hat{l}_S , is also of interest: while the reference value 0.235 is measured to the center of the ball-shaped foot, the estimated value is constantly longer by about 1 cm, which could include the rolling effects of the foot. The remaining IMU and pose sensor calibration parameters all exhibit a very good level of precision.

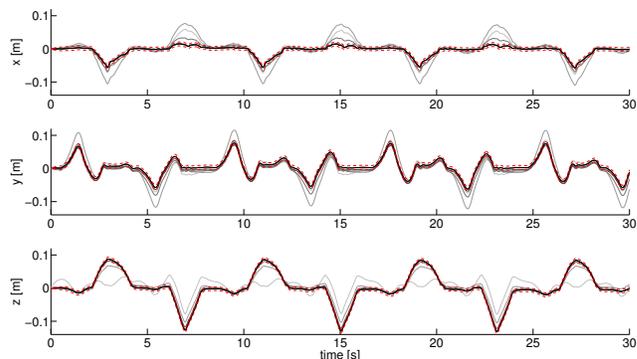


Fig. 2. From light gray to black: sequence of estimated robot positions, whereas the initial guess (all zeros) is not plotted. Dotted red: 3σ bounds of the final estimate.

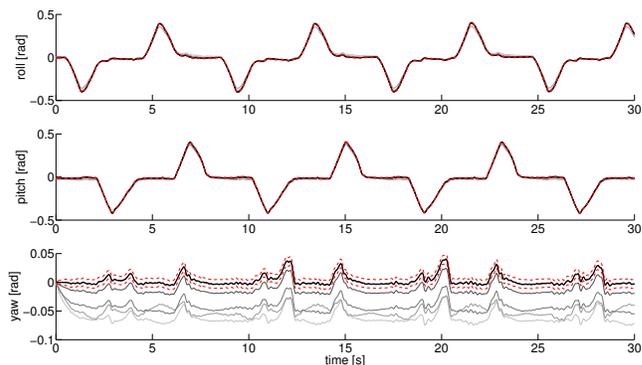


Fig. 3. From light gray to black: sequence of estimated robot attitudes (transformed to Euler angles), whereas the initial guess (all zeros) is not plotted. Dotted red: 3σ bounds of the final estimate. The alternating rolling and pitching motion can be nicely perceived.

V. CONCLUSION AND FUTURE WORK

This paper presents a framework for calibrating the different sensor modalities of a legged robot. By solving the batch optimization problem resulting from the maximum likelihood formulation, the approach yields accurate estimates for most of the calibration parameters. Notably, the framework is able to estimate many of the model parameters of the robot's forward kinematics. While additional sensors could be easily included, this procedure represents a fast and convenient way of getting a legged robot's calibration parameters. In a reduced implementation the batch optimization could also be solved repeatedly for bias estimation and thereby contribute to the performance of online state estimation. Future work will include the investigation of different kinematic models in order to more accurately capture the effects of the ball-shaped and compliant foot.

REFERENCES

- [1] O. Baser and E. I. Konukseven. Kinematic calibration of a 7 DoF haptic device. In *Advanced Robotics, Int. Conf. on*, Jun. 2011.
- [2] O. Birbach, B. Bauml, and U. Frese. Automatic and self-contained calibration of a multi-sensorial humanoid's

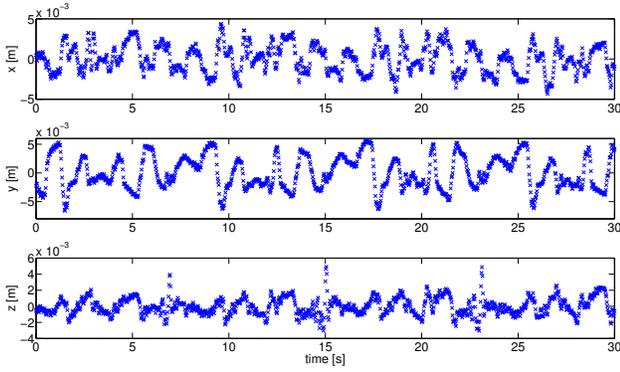


Fig. 4. Error residuals of the forward kinematics of one leg (expressed in the body frame). The RMS values for the different coordinates are: 0.0017 m, 0.0029 m, 0.0011 m. Strong autocorrelation can be observed.

TABLE I
ESTIMATED CALIBRATION PARAMETERS

	Par.	DS1	DS2	Std	Ref.
Kinematic calibration					
Body translation (m)	\hat{I}^{IB},x	-0.2675	-0.2673	0.0040	-0.2334
	\hat{I}^{IB},y	-0.0111	-0.0108	0.0035	0.0060
	\hat{I}^{IB},z	-0.0941	-0.0938	0.0032	-0.0905
Body rotation (rad)	\hat{q}_{BI},x	0.0025	0.0001	0.0104	0*
	\hat{q}_{BI},y	0.0060	0.0066	0.0077	0*
	\hat{q}_{BI},z	0.0216	0.0241	0.0041	0*
Segment lengths (m)	\hat{l}_H	-0.0644	-0.0687	0.0037	-0.0685
	\hat{l}_T	-0.1995	-0.2025	0.0057	-0.2
	\hat{l}_S	-0.2450	-0.2469	0.0054	-0.235*
Body dimensions (m)	\hat{b}_x	0.2455	0.2481	0.0078	0.2525
	\hat{b}_y	0.1811	0.1827	0.0061	0.185
Angular offsets (rad)	$\Delta\hat{\alpha}_1$	-0.0432	-0.0428	0.0147	0*
	$\Delta\hat{\alpha}_2$	0.0358	0.0381	0.0147	0*
	$\Delta\hat{\alpha}_3$	-0.0236	-0.0208	0.0135	0*
	$\Delta\hat{\alpha}_4$	0.0312	0.0298	0.0144	0*
	$\Delta\hat{\beta}_1$	0.0521	0.0691	0.0275	0*
	$\Delta\hat{\beta}_2$	0.0157	0.0198	0.0273	0*
	$\Delta\hat{\beta}_3$	-0.0381	-0.0429	0.0286	0*
	$\Delta\hat{\beta}_4$	-0.0597	-0.0710	0.0292	0*
	$\Delta\hat{\gamma}_1$	-0.0490	-0.0537	0.0326	0*
	$\Delta\hat{\gamma}_2$	-0.0016	-0.0009	0.0307	0*
	$\Delta\hat{\gamma}_3$	0.0124	0.0096	0.0330	0*
	$\Delta\hat{\gamma}_4$	0.0155	0.0140	0.0339	0*
IMU calibration					
Accelerometer bias (m/s ²)	$\hat{b}_{f,x}$	0.045	0.0492	0.0100	n/a
	$\hat{b}_{f,y}$	-0.0357	-0.0311	0.0103	n/a
	$\hat{b}_{f,z}$	0.0019	0.0018	0.0012	n/a
Gyroscope bias (rad/s)	$\hat{b}_{\omega,x}$	-0.0010	-0.0010	0.0005	n/a
	$\hat{b}_{\omega,y}$	0.0045	0.0044	0.0005	n/a
	$\hat{b}_{\omega,z}$	0.0032	0.0030	0.0005	n/a
Pose sensor calibration					
Sensor translation (m)	\hat{I}^{IC},x	-0.3435	-0.3443	0.0030	n/a
	\hat{I}^{IC},y	-0.0141	-0.0136	0.0029	n/a
	\hat{I}^{IC},z	-0.0066	-0.0074	0.0021	n/a
Sensor rotation (rad)	\hat{q}_{CI},x	-0.0085	-0.0085	0.0029	n/a
	\hat{q}_{CI},y	-0.0075	-0.0052	0.0029	n/a
	\hat{q}_{CI},z	-0.0371	-0.0360	0.0013	n/a

upper body. In *Robotics and Automation, IEEE Int. Conf. on*, May 2012.

[3] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Proc. of Robotics: Science and*

Systems, Jul. 2012.

[4] J. Brookshire and S. Teller. Extrinsic Calibration from Per-Sensor Egomotion. In *Proc. of Robotics: Science and Systems*, Jul. 2012.

[5] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka. Optimization based IMU camera calibration. In *Intelligent Robots and Systems, IEEE/RSJ Int. Conf. on*, Sep. 2011.

[6] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2011.

[7] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart. StarLETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion. In *Climbing and Walking Robots, Int. Conf. on*, Jul. 2012.

[8] J. Kelly and G. S. Sukhatme. Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *Int. J. of Robotics Research*, 30(1):56–79, Nov. 2011.

[9] W. Khalil, S. Besnard, and P. Lemoine. Comparison study of the geometric parameters calibration methods. *Int. J. of Robotics and Automation*, 15(2):56–67, 2000.

[10] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *Robotics and Automation, IEEE Int. Conf. on*, May 2011.

[11] M. Li. Kinematic calibration of an active head-eye system. *Robotics and Automation, IEEE Trans. on*, 14(1):153–158, 1998.

[12] F. M. Mirzaei and S. I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Trans. on*, 24(5):1143–1156, 2008.

[13] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Exp. Robotics, Int. Symp. on*, Dec. 2010.

[14] G. G. Scandaroli, P. Morin, and G. Silveira. A nonlinear observer approach for concurrent estimation of pose, IMU bias and camera-to-IMU rotation. In *Intelligent Robots and Systems, IEEE/RSJ Int. Conf. on*, Sep. 2011.

[15] H. Strasdat, J. M. M. Montiel, and A. Davison. Scale Drift-Aware Large Scale Monocular SLAM. In *Proc. of Robotics: Science and Systems*, Jun. 2010.

[16] K. H. Strobl and G. Hirzinger. Optimal hand-eye calibration. In *Intelligent Robots and Systems, IEEE/RSJ Int. Conf. on*, Oct. 2006.

[17] A. Ude. Nonlinear least squares optimisation of unit quaternion functions for pose estimation from corresponding features. In *Pattern Recognition, Int. Conf. on*, Aug. 1998.

[18] R. Wagner, O. Birbach, and U. Frese. Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM. In *Intelligent Robots and Systems, IEEE/RSJ Int. Conf. on*, Sep. 2011.