A Flexible Framework for Mobile Robot Pose Estimation and Multi-Sensor Self-Calibration

Davide Antonio Cucci and Matteo Matteucci
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

Keywords: Sensor Fusion, Mobile Robots, Odometry, Tracking, Calibration.

Abstract: The design and the development of the position and orientation tracking system of a mobile robot, and the calibration of its sensor parameters (e.g., displacement, misalignment and iron distortions), are challenging and time consuming tasks in every autonomous robotics project. The ROAMFREE framework delivers turn-on-and-go multi-sensors pose tracking and self-calibration modules and it is designed to be flexible and to adapt to every kind of mobile robotic platform. In ROAMFREE, the sensor data fusion problem is formulated as a hyper-graph optimization where nodes represent poses and calibration parameters and edges the non-linear measurement constraints. This formulation allows us to solve both the on-line pose tracking and the off-line sensor self-calibration problems. In this paper, we introduce the approach and we discuss a real platform case study, along with experimental results.

1 INTRODUCTION

In every project involving autonomous robots the design and development of the pose tracking subsystem is one of the first and most important milestones (Borenstein et al., 1996). This task is often time consuming and the performance of higher level control and navigation modules are tightly dependent on the localization accuracy. Moreover, to determine an estimate of the robot position and attitude with respect to the world, multiple sensors are usually employed. This often requires the calibration of intrinsic parameters for which direct measurement is impractical or impossible and the calibration task is usually performed by mean of time consuming ad-hoc, hand-tuned procedures especially designed to target the platform in use.

The ROAMFREE project aims at developing a general framework for robust sensor fusion and self-calibration to provide a turn-on-and-go 6-DOF pose tracking system for autonomous mobile robots. It includes: (i) a library of sensor families which are handled by the framework out-of-the box; (ii) an on-line tracking module based on Gauss-Newton minimization of the error functions associated to sensor readings; (iii) an off-line calibration suite which allows to estimate unknown sensor parameters such as the displacement and the misalignment with respect to the robot odometric center, scale factors and biases, magnetometer soft and hard iron distortion, and so on.

Figure 1: The Quadrivio ATV.

In the mobile robots literature the position and orientation tracking problem has often been addressed as a problem of multi-sensor data fusion and solved by means of Bayesian filters, such as Kalman and particle filters (Chen, 2003). Bayesian filtering has also been employed to the problem of on-line estimation of calibration parameters, such as the systematic and non-systematic components of the odometry error (Martinelli et al., 2007), or quantities which affects the tracking performances, such as the GPS latency (Bouvet and Garcia, 2000). If we restrict to the off-line calibration of single sensor displacement and misalignment, the problem is commonly referred as hand-eye clibration (Horaud and Dornaika, 1995).

Antonio Cucci D. and Matteucci M..
A Flexible Framework for Mobile Robot Pose Estimation and Multi-Sensor Self-Calibration.
DOI: 10.5220/0004484703610368
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2013), pages 361-368
ISBN: 978-989-8565-71-6

Although the effectiveness of previous approaches has been proven and are now well established in the literature, researchers are still required to write their own on ad-hoc implementations to adapt state-of-the-art techniques to the particular application or platform. More recently, in (Weiss et al., 2012), a general framework for position tracking and sensor displacement calibration is presented which relies on an Extended Kalman Filter to perform sensor fusion tasks.

Similarly, our work focuses on the development of a very general framework suited to be employed in every kind of autonomous robot platform. In this paper we describe the first deployment of the system to target the Quadrivio ATV, (see Figure 1), an autonomous off-road vehicle equipped with a wide variety of sensors, such as laser rangefinders, GPS, wheel and handlebar encoders and an Inertial Measurement Unit, which have to be integrated and calibrated to deliver 6-DOF position and orientation estimate to the higher level trajectory control and navigation modules. Differently from (Weiss et al., 2012), we base our framework on the use of a Gauss-Newton graph based estimation techniques as the ones applied in (Kümmerle et al., 2012) to the problem of 3-DOF simultaneous localization, mapping and calibration.

In the next section we give a high level description of the ROAMFREE framework; in Section 3 and Section 4 we go more in depth in the techniques employed and we describe the mathematical background that allows to perform the data fusion task in a sensor independent manner. In Section 5 we present the first deployment of the framework on the Quadrivio ATV and discuss preliminary experimental results. Conclusions and further work appear in Section 6.

2 FRAMEWORK DESCRIPTION

The ROAMFREE framework is designed to fuse displacement measurements coming from an arbitrary number of odometric sensors. Here, we focus on the generality of the approach trying to abstract as much as possible from the nature of the information sources. To this ends, we choose not to work directly with physical hardware, but with logical sensors, which are characterized only by the type of the proprioceptive measurement they produce. To clarify this point, consider a SLAM system built on the top of wheel encoders and multiple cameras. We call *logical sensor* the set of the encoders, the cameras and the SLAM algorithm, since, ultimately, it estimates the absolute pose of the robot with respect to a fixed reference frame. Moreover, the wheel encoders, taken alone, build up another logical sensor which measures the

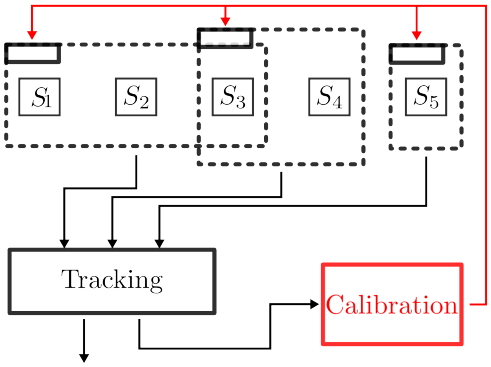


Figure 2: The ROAMFREE logical structure. Eventually overlapping groups of physical sensors, S_i, build up logical information sources whose output is fused by the tracking module. The position and attitude output is employed offline by the calibration module to estimate intrinsic parameters of the logical sensors.

roto-translation between poses at time t and $t + \Delta t$.

ROAMFREE allows the end-user to define logical sensors building up on separate modules and automatically integrate their measurements to both obtain a robust estimate of the robot pose and to calibrate unknown sensor parameters (see Figure 2). In this way, the developer can focus only on the internals of each logical sensor (e.g., signal processing, visual odometry algorithms), which extract information from the raw sensor data and then deliver it to ROAMFREE, e.g. through ROS (Quigley et al., 2009).

To achieve the abstraction form the physical hardware and be able to deal with very general, user-defined, information sources we observe that, despite the wide variety of proprioceptive sensors available for the design of mobile robots, all the useful logical sensor output for pose tracking fit into one of the following categories:

- absolute position and/or velocity
- angular and tangential speed in sensor frame
- acceleration in sensor frame
- vector field in sensor frame (e.g., magnetic field, gravitational acceleration)

For each of these categories ROAMFREE provides an error model which relates the state estimate with the measurement data, taking into account all the common sources of distortion, bias and noise (e.g., hard and soft magnetic distortion, sensor displacement or misalignment, etc.). These error models are then employed in the on-line estimation process to handle the logical sensor readings, according to their type. Moreover, the final user can choose a set of the predefined calibration parameters for which he can either provide values or let the framework estimate them us-

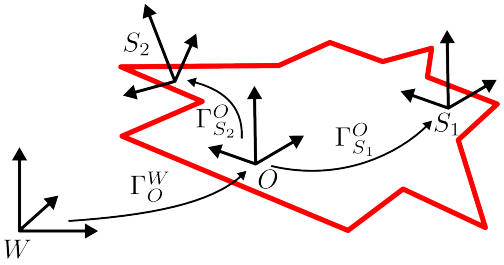


Figure 3: The reference frames employed in ROAMFREE. To represent the odometric reference frame O we store $\Gamma^W_O = [O^{(W)}, q^W_O]$. The sensor displacements and misalignment parameters of sensor S_i define the roto-translation from the odometric reference frame to the S_i reference frame $\Gamma^O_{S_i} = [\mathbf{S}_i^{(O)}, q^O_{S_i}]$.

ing an off-line formulation of the tracking problem.

In the following, a reference frame A will be represented with the roto-translation Γ^B_A which takes a point $p^{(A)}$, whose coordinates are expressed in reference frame A, to reference frame B. To represent such transformations, we use the origin of A expressed in reference frame B, $A^{(B)}$, and the quaternion q^B_A, i.e., the rotation taking points from frame A to frame B. So $\Gamma^B_A = [A^{(B)}, q^B_A]$.

ROAMFREE makes use of three reference frames (see Figure 3): W, the fixed world frame, O, the moving reference frame, placed at the odometric center of the robot, and the i-th sensor frame, S_i, whose origin and orientation are defined with respect to O in terms of displacement and misalignment. The tracking module estimates the position and orientation of O with respect to W, i.e., Γ^W_O. If absolute positioning sensors (e.g., GPS) are not available, we assume that $O \equiv W$ at time $t = 0$, and W becomes arbitrary. Otherwise, the origin of W is placed at given coordinates, with the y axis aligned with the geographic north and the z axis aligned with g and pointing up.

Currently we assume that all the logical sensor readings are precisely timestamped and that all the clocks in the system are synchronized. The importance of accurate measurement timestamps has been often stressed in the literature and a number of clock synchronization algorithms designed for robotics applications exist, i.e., (Harrison and Newman, 2011). If accurate timestamps are available, the framework is robust with respect to out-of-order measurements, as it will become clear in the next section.

The ROAMFREE framework is open-source and it is available at `http://roamfree.dei.polimi.it`.

3 POSE TRACKING AND SENSOR SELF-CALIBRATION

We formulate the tracking problem as a maximum likelihood optimization on an hyper graph in which the nodes represent poses and sensor parameters (to be eventually estimated) and hyper-edges correspond to measurement constraints. An error function is associated to each hyper-edge to measure how well the values of the nodes involved in the hyper-edge fit the sensor observations. The goal of the max-likelihood approach is to find a configuration for the poses and sensor parameters which minimizes the negative log-likelihood of the graph given all the measurements.

Let us discuss more precisely the problem formulation and techniques employed to solve it. Let $e_i(x_i, \eta)$ be the error function associated to the i-th edge in the hyper graph, where x_i is a vector containing the variables appearing in any of the connected nodes and η is a zero-mean Gaussian noise. Thus $e_i(x_i, \eta)$ is a random vector and its expected value is computed as $\overline{e}_i(x_i) = e_i(x_i, \eta)|_{\eta=0}$. Since e_i can involve non-linear dependencies with respect to the noise vector, we compute the covariance of the error through linearization. Let Σ_η be the covariance matrix of η and J_i the Jacobian of e_i with respect to η:

$$\Sigma_{e_i} = J_i \Sigma_\eta J_i^T \Big|_{\eta=0}. \tag{1}$$

The optimization problem is defined as follows:

$$\mathcal{P}: \quad \arg\min_x \sum_{i=1}^{N} \overline{e}_i(x_i)^T \Omega_{e_i} \overline{e}_i(x_i), \tag{2}$$

where $\Omega_{e_i} = \Sigma^{-1}_{e_i}$ is the i-th edge information matrix and N is the total number of edges. If a reasonable initial guess for x is known, a numerical solution for $\mathcal{P}$ can be found by means of the poular Gauss-Newton (GN) or Levenberg-Marquardt (LM) algorithms, see for example (Gill and Murray, 1978): first the error functions are approximated with their first order Taylor expansion around the current estimate $\check{x}$:

$$e_i(\check{x} \boxplus \Delta x) \simeq e_i(\check{x}) + J_i \Delta x, \tag{3}$$

where J_i is the Jacobian of e_i with respect to Δx evaluated in $\Delta x = 0$ and $\boxplus$ is an operator which applies an increment Δx to $\check{x}$. Substituting the error functions $e_i(x_i)$ in (2) with their first order expansion yields a quadratic form which can be solved in Δx. Finally we set $\check{x} \leftarrow \check{x} \boxplus \Delta x$ and we update the Ω_{e_i} as in (1). The process is iterated till termination criteria are met.

Note that this formulation of the optimization problem, thanks to the $\boxplus$ operator, allows us to consistently work with variables which span over non-Euclidean spaces, such as $SO(3)$ and $SE(3)$.

In these cases, over-parametrized representations are often employed, and substituting the $\boxplus$ operator with regular $+$ in (3) would yield solutions for Δx which could break the constraints induced by over-parametrization. Instead, the $\boxplus$ operator consistently applies a local variation Δx, belonging to an Euclidean space, to the original variable. Consider for instance x being a quaternion: a 4D representation is employed to represent 3-DOF rotations.

$$x \boxplus \Delta x = x \otimes \left[1 - \sqrt{\Delta x_1^2 + \Delta x_2^2 + \Delta x_3^2}, \Delta x_1, \Delta x_2, \Delta x_3\right], \quad (4)$$

where $\otimes$ is the quaternion product operator and $\Delta x \in \Re^3$. This ensures that $||x \boxplus \Delta x|| = 1$ and that quaternions are consistently updated once Δx has been determined. This technique is called *manifold encapsulation*, for details see (Hertzberg et al., 2013).

We next discuss how an instance of the hyper-graph optimization problem for the tracking and/or sensor self-calibration problem is constructed (see Figure 4). We first need to instantiate pose and parameter nodes and measurement edges and then determine an initial guess for all the variables in the problem. We choose a high frequency sensor from which it is possible to predict $\Gamma_O^W(t+\Delta t)$ given $\Gamma_O^W(t)$ and its measurement $z(t)$. Each time a new reading for this sensor is available, we instantiate a new node $\Gamma_O^W(t+\Delta t)$ using the last pose estimate available, $\check{\Gamma}_O^W(t)$, and $z(t)$ to compute an initial guess for this node. $z(t)$ is also employed to initialize an odometry edge between poses at time t and $t+\Delta t$.

When other measurements from different sensors are available, their corresponding edge is inserted into the graph relying on the existing nodes. More precisely, the pose node (or the pair of poses) nearest to the measurement timestamp is employed. For instance, if a new gyroscope reading is available with timestamp t we search in the graph for the pair of subsequent poses with timestamps nearest to t and we insert an angular speed constraint between the selected nodes. We assume here that, if the sensor employed to initialize the pose nodes has a sufficiently high sampling rate, the error introduced by this approximation is negligible. Note that, if the timestamps satisfy the assumptions mentioned in the previous section, this approach natively allows to deal with out-of-order measurements since constraints can be instantiated in the past between proper pose nodes an contribute to the refinement of the newest pose estimate.

To solve the optimization problem we employ g^{2}o (Kümmerle et al., 2011), a general framework for graph optimization which relies on very efficient implementations and it is reported to solve graph problems with thousands of nodes in fractions of a second, allowing us to perform the tracking task on-line.

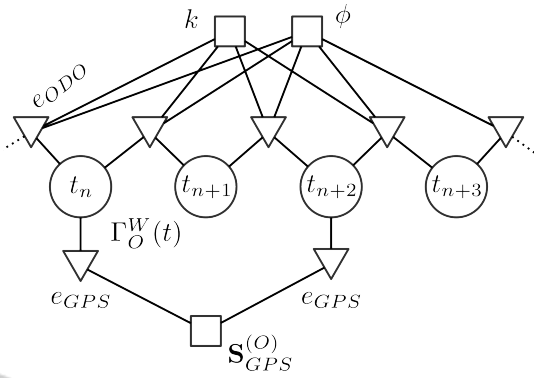


Figure 4: An instance of the tracking optimization problem with four pose vertices $\Gamma_O^W(t)$ (circles), odometry edges e_{ODO} (triangles), two shared calibration parameters vertices k and ϕ (squares), two GPS edges e_{GPS} and the GPS displacement $\mathbf{S}_{GPS}^{(O)}$.

The graph optimization approach is suited to both the on-line position tracking problem, in which the requirements are substantially related to precision and robustness of the estimate and real-time frequency of operation, and the off-line calibration problem, in which the goal is to determine the sensor parameters assignments which globally maximize the likelihood of all the measurements collected during a long robot roam. Indeed, in case of on-line tracking the parameter values are known and the corresponding nodes are excluded from the optimization. Moreover, to allow real time operation, we assume that the process memory extends up to N time slots in the past and we discard older nodes and edges. The optimization is run at constant frequency and we iterate the GN algorithm till timing deadlines approach or when the error function improvement drops below the chosen threshold. Conversely, during off-line calibration, a set of the parameter nodes is chosen for estimation and the graph containing all available measurements is considered. Note that with the same problem formulation it is also possible to track on-line a limited set of calibration parameters, along with the robot pose, which are subject to change, such as slippage coefficients.

4 ERROR MODELS

In this section we introduce the sensor models employed in ROAMFREE and the corresponding error functions, which measure how well the sensor reading fit the corresponding poses. Since error functions are evaluated multiple times during GN optimization, when we refer to state variables, e.g., $\Gamma_O^W(t)$, we al-

ways intend their current estimate, e.g., $\check{\Gamma}^W_O(t)$. Sensor parameters are highlighted by a bold font.

4.1 General Odometer

Every kind of odometer can be seen as a sensor which measures the roto-translation between two subsequent poses, i.e., $\Gamma^{O(t)}_{O(t+\Delta t)}$, from which, assuming constant speed between t and $t+\Delta t$, it is possible to obtain estimates of the robot tangential speed and turn rate $\hat{v}^{(S)}(t)$ and $\hat{\omega}^{(S)}(t)$ in sensor frame. Integrating these quantities over Δt yields a predictor for $\Gamma^W_{O(t+\Delta t)}$:

$$\begin{aligned}\hat{O}^{(W)}(t+\Delta t) &= \left(q^W_O(t)\mathbf{q}^O_S\right)\hat{v}^{(S)}(t)\Delta t + O^{(W)}(t)\\ \hat{q}^W_O(t+\Delta t) &= \Phi\left(\mathbf{q}^O_S\hat{\omega}^{(S)}(t),\,\Delta t\right)q^W_O(t).\end{aligned} \tag{5}$$

Here $\hat{q}^W_O(t+\Delta t)$ it is obtained by means of the well known Rodriguez formula:

$$\Phi(\omega,\Delta t) = \left[\cos(|\omega|\frac{\Delta t}{2})I + 2\frac{\sin(|\omega|\frac{\Delta t}{2})}{|\omega|}W(\omega)\right], \tag{6}$$

where $W(\omega)$ is the 4×4 skew-symmetric matrix constructed from the augmented ω vector $[0,\omega_x,\omega_y,\omega_z]$. Note that $\hat{v}^{(S)}(t)$ and $\hat{\omega}^{(S)}(t)$ are random vectors.

The difference between state estimate and predictors yields the odometry error:

$$e_{ODO}(t) = \begin{bmatrix} \hat{O}^{(W)}(t+\Delta t) - O^{(W)}(t+\Delta t) \\ \hat{q}^W_O(t+\Delta t)\otimes[q^W_O(t+\Delta t)]^{-1}\end{bmatrix}. \tag{7}$$

Note that the Gaussian noise corrupting $\hat{\omega}^{(S)}(t)$, because of the constraints involved in the quaternion representation of 3-DOF rotations (i.e., $||q||=1$), yields a singular covariance matrix for the full quaternion error vector. To overcome this issue, we assume the orientation error being small (i.e., the quaternion part of the error vector is near to the identity, $q\simeq[1,0,0,0]$) and we discard the first component, yielding a well defined information matrix Ω_{e_i}.

Note that a wide variety of sensors can be treated in this way, e.g., visual odometry systems, scan-matching algorithms, but also cinematic models, since a prediction $\hat{\Gamma}^{O(t)}_{O(t+\Delta t)}$ can be computed applying the forward cinematic to the control inputs. For the sake of clarity we develop here an example of cinematic odometry based on the platform of Figure 1.

Four wheel vehicles usually employ the Ackermann steering geometry. The odometry for these vehicles is computed placing a reference frame S_{ACK} in the middle point of the rear wheels axis. A first order approximation of the dynamic equations governing the motion of such vehicles yields the predictors $\hat{v}^{(S_{ACK})}(t)$ and $\hat{\omega}^{(S_{ACK})}(t)$ as a function of rear wheels speed $v_{ACK}(t)$ and steering angle $\theta_{ACK}(t)$, i.e.:

$$\begin{aligned}\hat{v}^{(S_{ACK})}(t) &= \begin{bmatrix}\mathbf{k}_v v_{ACK}(t)\\ 0\\ 0\end{bmatrix} + \eta_v\\ \hat{\omega}^{(S_{ACK})}(t) &= \begin{bmatrix}0\\ 0\\ \frac{\mathbf{k}_v v_{ACK}(t)}{\mathbf{L}}\tan\left(\mathbf{k}_\theta\theta_{ACK}(t)+\mathbf{k}_\gamma\right)\end{bmatrix} + \eta_\omega,\end{aligned} \tag{8}$$

where $[\mathbf{k}_v,\mathbf{k}_\theta,\mathbf{k}_\gamma]$ is a vector of gains and biases and $\mathbf{L}$ is the distance between front and rear wheel axes (Abe and Manning, 2009). Note that the vehicle motion is constrained on a 2D surface, which is not necessarily flat. The *predicted* pitch and roll speed are zero, although these components speed can not be computed with the information provided by $v_{ACK}(t)$ and $\theta_{ACK}(t)$. Thus we have to consider large variances of the corresponding components of η.

4.2 GPS

The GPS sensor measures its own latitude, longitude and elevation with respect to a standard, Earth fixed, reference frame. The pose of the robot can be employed to predict the GPS reading, once this has been projected on an Euclidean ENU (East, Nord, Up) three-dimensional space centered at W

$$\hat{z}_{GPS}(t) = O^{(W)}(t) + q^W_O(t)\mathbf{S}^{(O)}_{GPS} + \eta, \tag{9}$$

where $\mathbf{S}^{(O)}_{GPS}$ is GPS sensor displacement. The same equation can also be employed with other absolute positioning logical sensors, such as SLAM systems. The error associated to the GPS measure is simply $e_{GPS}(t) = \hat{z}_{GPS}(t) - z_{GPS}(t)$ and $\mathbf{e}_{GPS}(t) \in \Re^3$.

4.3 Magnetometer

The three-axis magnetometer reading is a measure of the the Earth's magnetic field $h^{(W)}$ in sensor frame, affected by various sources of distortion, bias and noise. Here we employ an error model that keeps into account hard and soft iron distortion, non-orthogonality of the sensor axes, scaling, bias and misalignment with respect to the odometric reference frame, as discussed in (Vasconcelos et al., 2011):

$$\hat{\mathbf{z}}_{MAG}(t) = \mathbf{A}\left(q^W_O(t)\mathbf{q}^O_{S_{MAG}}h^{(W)}\right) + \mathbf{b} + \eta, \tag{10}$$

where $\mathbf{A}$ is an unconstrained 3×3 matrix, $\mathbf{b}\in\Re^3$ is a bias vector, and $h^{(W)}$ is the Earth's magnetic field vector in the robot operation area. The error associated to the magnetometer measurement is then $e_{MAG}(t) = \hat{z}_{MAG}(t) - z_{MAG}(t)$ and $e_{MAG}(t)\in\Re^3$.

4.4 Gyroscope

The gyroscope sensor measures its own angular velocity in sensor frame. Here we assume the robot to be a rigid body and thus experiencing the same angular velocity at every point. Under this assumption, a gyroscope provides a direct measure for the angular velocity of the odometric reference frame. As in (5):

$$\hat{q}_O^W(t+\Delta t) = \Phi\big(\mathbf{q}_{S_{GYRO}}^O(z_{GYRO}+\eta), \Delta t\big) q_O^W(t), \quad (11)$$

where the gyroscope reading is moved in the odometric reference frame by means of the misalignment rotation $\mathbf{q}_S^O$ and then employed to predict robot orientation at time $t+\Delta t$. Thus e_{GYRO} can be computed as the difference between $\hat{q}_O^W(t+\Delta t)$ in (11) and the current estimate of $q_O^W(t+\Delta t)$:

$$e_{GYRO}(t) = \hat{q}_O^W(t+\Delta t) \otimes [q_O^W(t+\Delta t)]^{-1}. \quad (12)$$

Again we assume that e_{GYRO} represents a small rotation and we discard the first component of the quaternion so that $e_{GYRO} \in \Re^3$.

5 EXPERIMENTAL RESULTS

In this section we discuss preliminary experimental results for the first deployment of the ROAMFREE framework, targeting the Quadrivio ATV (Bascetta et al., 2009)(Bascetta et al., 2012), an all-terrain autonomous vehicle being developed with the aim of off-road autonomous navigation on rough terrains. In these environments, high level navigation and trajectory control modules could benefit from the ROAMFREE 6-DOF pose tracking capabilities.

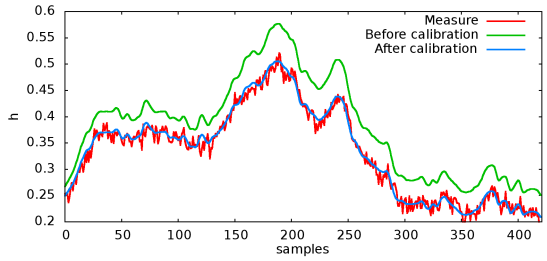
The ATV is equipped with a wide variety of sensors which make it an interesting experimental platform for multi-sensor fusion techniques. These consist in a Xsens MTi, a MEMS inertial measurement unit with integrated 3D magnetometer, a low cost Yuan10 L1 C/A code USB GPS receiver and two Sick laser rangefinders, which were not employed in this preliminary experiments. Moreover, two encoders read the handlebar steering angle and wheel speed.

On this platform, we tested the tracking and calibration capabilities of the ROAMFREE framework. Indeed, multiple unknown parameters have to be determined in order to achieve maximum pose tracking accuracy and their direct measurement is not always simple or practical. Here we focus on the following set of parameters: the GPS displacement, $\mathbf{S}_{GPS}^{(O)}$ and the Xsens misalignment, $\mathbf{q}_{S_{IMU}}^O$, with respect to the odometric reference frame, the magnetometer hard and soft iron distortion compensation matrix and bias, $\mathbf{A}$ and $\mathbf{b}$, the Ackermann steering geometry and encoders parameters, $[\mathbf{k}_v, \mathbf{k}_\theta, \mathbf{k}_\gamma, \mathbf{L}]$. We start from raw sensors data collected while driving the ATV and then we perform the off-line optimization to find the parameter values which maximize the likelihood of the collected measurements.

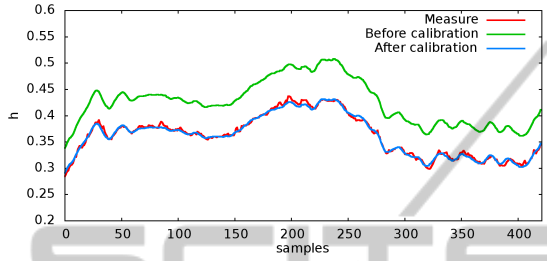
Note that it is not guaranteed that all the calibration parameters are observable. For example the z component of $\mathbf{S}_{GPS}^{(O)}$ can not be estimated considering only planar trajectories. Fortunately, these phenomena can be detected looking at the approximated Hessian of the error functions computed during the Gauss-Newton optimization process, from which we can extract an estimation of the marginal covariance of the estimates. In case of non-observable parameters given the trajectory considered, their covariance will be high, implying that perturbations in their values have limited effect on the error functions. It is difficult to give theoretical results on the conditions which have to be satisfied by the trajectory to be able to observe all of a given set of calibration parameters in the general case. A detailed analysis of the differential drive case can be found in (Censi et al., 2008). Morover, multiple solutions to the calibration problem may exist. Again, these situations can be detected looking at the marginal cross covariance of the parameter variables, in which high values suggest the existence of dependencies between the involved parameters. In these cases one possible strategy is to first obtain a coarse-grained estimate of one set of parameters while keeping the dependent ones fixed. In ROAMFREE this can be done by means of a graphical calibration tool which allows the user to track the convergence of the optimization, displays the current estimates of the calibration parameters and their covariance matrix, and let him fix and unfix nodes or modify noise covariances.

In our experiments we considered a calibration trajectory consisting in 303.5 m of planar roaming. We built the optimization graph instantiating a new pose node at a frequency of 20 Hz, every time a new reading from the Ackermann odometry is available. Magnetometer (100 Hz) and GPS (1 Hz) readings constraints the pose node with nearest timestamp. Moreover, each pair of subsequent pose nodes is constrained by one Ackermann odometry and one gyroscope edge. The ATV was also equipped with a Trimble 5700 GPS in RTK configuration to provide accurate position measures at 5 Hz that were used as a ground truth reference for pose tracking.

We considered two sets of parameters and estimated them in the following order, while keeping the others fixed: $[\mathbf{S}_{GPS}^{(O)} \mathbf{q}_{S_{IMU}}^O]$ and $[\mathbf{A}, \mathbf{b}, \mathbf{k}_v/\mathbf{L}, \mathbf{k}_\theta, \mathbf{k}_\gamma]$. We initialized parameter nodes with rough estimates ob-



(a) X axis



(b) Y axis

Figure 5: A portion of the X-Y magnetometer readings compared with predictions computed from $q_O^W(t)$, before and after calibration for $\mathbf{A}$ and $\mathbf{b}$.

tained by direct measurements on the vehicle (i.e., tape measure), or with default assignments such as $\mathbf{A} = I$ and $\mathbf{b} = 0$. The following covariance matrices were assumed for the noise vectors η: $\Sigma_{GPS} = 0.69\,I$, $\Sigma_{ACK} = \text{diag}(0.027, 10, 100, 10, 100, 0.003)$, $\Sigma_{MAG} = 0.004\,I$ and $\Sigma_{GYRO} = 0.003\,I$. We collected data by means of an on-board PC, employing the system clock to provide rough sensor reading timestamps.

The results of the calibration procedure were:

$$\mathbf{S}_{GPS}^{(O)} = \begin{bmatrix} -0.52 \\ 1.14 \\ 8.30 \end{bmatrix} \quad \mathbf{q}_{S_{IMU}}^{O} = \begin{bmatrix} 0.997 \\ 0.003 \\ -0.004 \\ -0.079 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0.87 & -0.02 & -0.02 \\ 0.00 & 0.88 & 0.00 \\ -0.02 & -0.04 & 0.85 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -0.008 \\ -0.007 \\ -0.039 \end{bmatrix}$$

$$\mathbf{k}_v/\mathbf{L} = 0.797 \quad \mathbf{k}_\theta = 1.043 \quad \mathbf{k}_\gamma = 0.005 \ .$$

With the only exception of $\mathbf{S}_{GPS}^{(O)}$, all the estimated values, along with their covariance matrices, are consistent with the direct measures. Moreover, the magnetometer calibration parameters $\mathbf{A}$ and $\mathbf{b}$ are able to accommodate for the soft and iron distortion. Indeed, given the estimated trajectory of the robot, from the orientation at time t we can compute a prediction of the magnetometer readings; in Figure 5 we compare these predictions with the actual measurements before and after the calibration of $\mathbf{A}$ and $\mathbf{b}$.

Regarding $\mathbf{S}_{GPS}^{(O)}$, its estimate does not reflect the direct measurements performed on the robot. Nevertheless, we note that the direct measurement of this parameter is within the estimate confidence interval, given its covariance matrix at the end of the Guass-Newton optimization, which was:

$$\Sigma_{\mathbf{S}_{GPS}^{(O)}} = \begin{bmatrix} 0.118 & 0.001 & -0.107 \\ 0.001 & 0.015 & 0.146 \\ -0.107 & 0.146 & 8.049 \end{bmatrix}.$$

Note that the z component is not observable given the trajectory considered and this results in large variance for the z component. We argue that the high uncertainty and the lack of physical meaning of the estimated $\mathbf{S}_{GPS}^{(O)}$ is caused by limited timestamps accuracy, high noise, and limited frequency of the available data. To confirm this hypothesis, we repeated the calibration employing the high accuracy RTK GPS readings instead of the low cost Yuan10. This time the result was $\mathbf{S}_{RTK}^{(O)} = [0.71, -0.44, 1.75]$, which is very accurate with respect to direct measurements.

Finally we tested the on-line pose tracking system. In the tracking benchmarks we consider windows of 60 poses. Again each new pose node is added every time a new Ackermann odometry reading is available, i.e., every 0.05s. We compared the trajectory, estimated from on-board sensors data, with the RTK GPS measurements. The displacement of the reference RTK GPS with respect to the Yuan10 GPS sensor used for tracking was carefully measured and can be assumed to be $\mathbf{S}_{RTK}^{(S_{GPS})} = [-0.29, -1.10, 0.00]$. In Figure 6 we plot the position of the RTK GPS computed from the estimate of $O^{(W)}$ and the ground truth. It is possible to see that we are able to smoothly track position and orientation of the robot despite the limited precision and sample rate of the Yuan10 GPS.

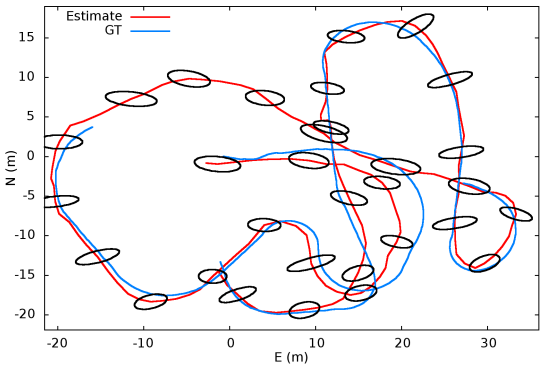


Figure 6: Ground truth RTK GPS readings and its position $\mathbf{S}_{RTK}^{(W)}$ computed from the estimate of $O^{(W)}$, along with 3σ uncertainties.

6 CONCLUSIONS AND FURTHER WORK

In this work we have presented the ROAMFREE mobile robot pose tracking and sensor self-calibration framework along with experimental results about the very first deployment of the framework to target the Quadrivio ATV. ROAMFREE is being continuously developed and our work focuses on the framework structure, enriching the sensor library and improving the sensor fusion techniques.

ROAMFREE is going to be extended to handle acceleration measures. To do this, we plan to employ Lie Groups operators to compute the finite difference derivative of the pose nodes in the graph, consistently working on the $SE(3)$ manifold. This will also allow us to interpolate the pose nodes in the graph assuming constant acceleration and thus obtain virtual pose nodes placed at the precise measurement timestamps.

We are also working on the extension of the calibration suite to allow the estimation of latencies possibly compromising measurement timestamps, and we are considering to feedback the output of the on-line tracking module to the logical sensors so that they can take advantage of the accurate estimate of $\Gamma_O^W(t)$ to update their internal state (i.e., update the tracked image feature positions in a visual odometry system).

ACKNOWLEDGEMENTS

This work has been supported by the Italian Ministry of University and Research (MIUR) through the PRIN 2009 grant "ROAMFREE: Robust Odometry Applying Multi-sensor Fusion to Reduce Estimation Errors".

REFERENCES

Abe, M. and Manning, W. (2009). *Vehicle handling dynamics: theory and application*. Butterworth-Heinemann.

Bascetta, L., Cucci, D., Magnani, G., Matteucci, M., Osmankovic, D., and Tahirovic, A. (2012). Towards the implementation of a mpc-based planner on an autonomous all-terrain vehicle. In *In Proceedings of Workshop on Robot Motion Planning: Online, Reactive, and in Real-time (IEEE/RJS IROS 2012)*, pages 1–7.

Bascetta, L., Magnani, G., Rocco, P., and Zanchettin, A. (2009). Design and implementation of the low-level control system of an all-terrain mobile robot. In *Advanced Robotics (ICAR), 2009 International Conference on*, pages 1–6. IEEE.

Borenstein, J., Everett, H., and Feng, L. (1996). Where am i? sensors and methods for mobile robot positioning. *University of Michigan*, 119:120.

Bouvet, D. and Garcia, G. (2000). Improving the accuracy of dynamic localization systems using rtk gps by identifying the gps latency. In *Robotics and Automation (ICRA), 2000 IEEE International Conference on*, pages 2525–2530. IEEE.

Censi, A., Marchionni, L., and Oriolo, G. (2008). Simultaneous maximum-likelihood calibration of odometry and sensor parameters. In *Robotics and Automation (ICRA), 2008 IEEE International Conference on*, pages 2098–2103. IEEE.

Chen, Z. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69.

Gill, P. E. and Murray, W. (1978). Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal on Numerical Analysis*, 15(5):977–992.

Harrison, A. and Newman, P. (2011). Ticsync: Knowing when things happened. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 356–363. IEEE.

Hertzberg, C., Wagner, R., Frese, U., and SchröDer, L. (2013). Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77.

Horaud, R. and Dornaika, F. (1995). Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210.

Kümmerle, R., Grisetti, G., and Burgard, W. (2012). Simultaneous parameter calibration, localization, and mapping. *Advanced Robotics*, 26(17):2021–2041.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g^2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE.

Martinelli, A., Tomatis, N., and Siegwart, R. (2007). Simultaneous localization and odometry self calibration for mobile robot. *Autonomous Robots*, 22(1):75–85.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3.

Vasconcelos, J., Elkaim, G., Silvestre, C., Oliveira, P., and Cardeira, B. (2011). Geometric approach to strapdown magnetometer calibration in sensor frame. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(2):1293–1306.

Weiss, S., Achtelik, M., Chli, M., and Siegwart, R. (2012). Versatile distributed pose estimation and sensor self-calibration for an autonomous mav. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 31–38. IEEE.