

Multisensor Data Fusion for Robust Pose Estimation of a Six-Legged Walking Robot

Annett Chilian, Heiko Hirschmüller, Martin Görner

Abstract—For autonomous navigation tasks it is important that the robot always has a good estimate of its current pose with respect to its starting position and – in terms of orientation – with respect to the gravity vector. For this, the robot should make use of all available information and be robust against the failure of single sensors. In this paper a multisensor data fusion algorithm for the six-legged walking robot DLR Crawler is presented. The algorithm is based on an indirect feedback information filter that fuses measurements from an inertial measurement unit (IMU) with relative 3D leg odometry measurements and relative 3D visual odometry measurements from a stereo camera. Errors of the visual odometry are computed and considered in the filtering process in order to achieve accurate pose estimates which are robust against visual odometry failure. The algorithm was successfully tested and results are presented.

I. INTRODUCTION

Mobile robots are built with the goal that they will be able to perform tasks in environments which are dangerous for humans. For example, they should explore the surface of other planets, climb into caves or provide information from disaster sites. While for most of these tasks a certain degree of autonomy is beneficial, it is absolutely necessary for robots on planetary surfaces because long signal round trip times forbid remote control of the robot. A basic skill of an autonomous robot is the robust estimation of its current pose. For this, the robot should not depend on external information because in some applications GPS or a priori maps are not available. Thus, the robot should be able to estimate its pose using only its own sensors, for example wheel encoders, joint sensors, visual and inertial sensors. However, when GPS or other additional information become available, the robot should use them.

The DLR Crawler [1] (ref. Fig. 1) is a six-legged actively compliant walking robot which was developed as a prototype of an autonomous robot for rough terrain. Its legs are equipped with joint angle sensors that allow the computation of a leg odometry in six degrees of freedom (DOF). Furthermore, it has a stereo camera head for perceiving its environment and an inertial measurement unit (IMU) to measure accelerations and angular velocities. For autonomous navigation it needs accurate and robust position and orientation estimates. Since each kind of sensor has disadvantages under certain conditions, the information of all available sensors must be combined so that robust pose estimation is possible.

A. Chilian, M. Görner, H. Hirschmüller are with the Institute of Robotics and Mechatronics, DLR German Aerospace Center, Oberpfaffenhofen, Germany annett.chilian@dlr.de

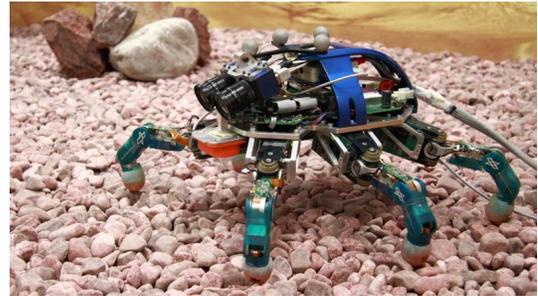


Fig. 1. DLR Crawler in the gravel testbed

Many authors have addressed the problem of sensor data fusion. Often IMU data is combined with GPS readings because of their complementary properties. A multisensor Kalman filter is presented by Caron et al. for fusing IMU and GPS data [2]. Dissanayake et al. presented an indirect information filter for fusing GPS, IMU and wheel encoder data [3].

Other authors do not use GPS data but combine inertial measurements with visual and/or odometry information. Konolige et al. show a data fusion method for correcting visual odometry measurements by IMU roll and pitch angle measurements with respect to the gravity vector and by IMU angular rate measurements for the yaw angle using an extended Kalman filter [4]. Helmick et al. use the indirect extended Kalman filter formulation to fuse IMU measurements with relative measurements from visual odometry and vehicle odometry [5]. Lamon and Siegwart used an extended information filter for fusing data from an inertial navigation system and a three-dimensional rover odometry [6].

In this paper, a multisensor data fusion algorithm is presented which uses an indirect information filter for fusing inertial measurements of an IMU with relative translation and rotation measurements from a 3D visual odometry and 3D leg odometry. An error estimate for each visual odometry measurement is computed and used in the fusion process to assign a lower weight to the measurement when the visual conditions are bad.

II. SENSOR MEASUREMENTS

A. Inertial Measurements

The DLR Crawler uses an XSens IMU which consists of three accelerometers and three gyroscopes measuring the accelerations in x, y, z direction and the angular velocities around the three axis, respectively. IMU data is received at a rate of 120 Hz. Since the accelerometers also sense the

gravity, absolute roll and pitch angles can be derived. The accelerations and angular velocities can be integrated over time to get the velocity, position and orientation of the IMU. However, since the accelerometer and gyroscope measurements are biased, the errors in position and orientation will grow unbounded due to the integration. For this, an IMU cannot be used solely to give good pose estimates but it needs to be corrected by other sensors with less drift.

The advantage of the IMU is that it only depends on the present gravity and apart from that is independent of environmental conditions.

B. Visual Odometry

1) *Computation of Visual Odometry*: The stereo camera permits the ego-motion estimation of a robot from the images, which is known as visual odometry. The advantage of it in mobile robotics is, that it is independent of wheel or leg slip. The used stereo visual odometry [7] is based on depth images that are available, since they are also required for other tasks like obstacle avoidance, mapping, etc. Depth images are computed by the Semi-Global Matching (SGM) method [8], which is very accurate at fine structures and in weakly textured terrain. A GPU implementation runs with 4-5 Hz on VGA sized images with 128 pixel disparity range. Assuming the environment is static, corresponding feature points are detected in subsequent images using a corner detector, followed by feature point matching. The disparity image is used for 3D reconstruction of all feature points. Thereafter, a robust algorithm is used for removing outliers in the correspondences using the rigidity constraint. Finally, the 6 DOF to register the point clouds of the previous and current image frame are determined in closed form as initial ego-motion estimate [7].

2) *Visual Odometry Error*: Regardless of the exact method for determining the initial ego-motion estimate, a visual odometry method typically performs non-linear optimization for minimizing either the reprojection error of reconstructed features or another error model as fast approximation of the reprojection error [9]. We used $\mathbf{x} = [t_x, t_y, t_z, n_1, n_2, n_3]^T$ as parametrization of the transformation with $\mathbf{t} = [t_x, t_y, t_z]^T$ as translation, $\mathbf{n} = [n_1, n_2, n_3]^T$ as rotation axis and $\alpha = |\mathbf{n}|$ as rotation angle. Let $\mathbf{y} = [p'_{1x} - p_{1x}, \dots, p'_{ny} - p_{ny}]^T$ be the reprojection error vector that is the difference between the feature point locations p_i in the image and the corresponding projected locations of their reconstructions p'_i . The function to be minimized (e.g. by Levenberg-Marquardt) is then $\mathbf{y} = f(\mathbf{x})$ with $\mathbf{y}_0 = f(\mathbf{x}_0)$ as solution with the lowest reprojection error.

Assuming that the correspondences are outlier free, $\varepsilon_p = 0.5$ pixel is a typical error in feature point localization. The propagation of this error into the parameters results in the parameter error ε_x . For computing this error, the function $f(\mathbf{x})$ must be inverted. Since it is not invertible in closed form (otherwise a non-linear optimization would not be needed), a linearization is computed at \mathbf{x}_0 as approximation

$$\mathbf{y} = f(\mathbf{x}) \approx \mathbf{J}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{y}_0, \quad (1)$$

with \mathbf{J}_0 as Jacobi matrix at the solution \mathbf{x}_0 . The Levenberg-Marquardt optimization computes the Jacobi matrix internally, which may be reused, or it can be computed from scratch by numerical forward differentiation of $f(\mathbf{x}_0)$. For small values $\mathbf{x} - \mathbf{x}_0$, the linearization is a good approximation of the original function. The function can now be inverted by $\mathbf{x} - \mathbf{x}_0 = \mathbf{J}_0^+(\mathbf{y} - \mathbf{y}_0)$, with \mathbf{J}_0^+ as pseudo inverse of the Jacobian, computed by singular value decomposition. In this formulation, the error ε_p can be propagated individually, corresponding to each element of \mathbf{y} , by $\varepsilon_x^i = \mathbf{J}_0^+ \varepsilon^i$, with ε^i as null vector with only element i set to ε_p . If independent errors are assumed, then the individually propagated errors are simply the square root of the sum of squares according to the rules of error propagation. This is effectively the same as multiplying the pixel error with the L_2 norm over the rows of the inverse Jacobian, i.e.

$$\varepsilon_{xi} = \varepsilon_p |\mathbf{J}_{0i}^+|, \quad (2)$$

with \mathbf{J}_{0i}^+ as the i -th row of \mathbf{J}_0^+ . It is important to understand that the estimation of the visual odometry error ε_x implicitly includes all sources of errors due to bad conditioned scenes with weak texture or low contrast, like low number of correspondences, feature points that are clustered in an image area, etc. Therefore, it is a very good value for judging the quality of visual odometry for fusion with other ego-motion sensors.

C. Leg Odometry

Using joint angle and joint torque measurements, a 6 DOF odometry of the DLR Crawler is computed. It estimates relative pose changes of the robot based on matching point clouds, which are represented by the positions of the supporting feet. The algorithm assumes rigidity of the configurations, which implies a no slip condition for the feet. Hence, the quality of the relative leg odometry measurements depends on the ground conditions. Since the basic odometry is subject to strong drift of the pitch and roll angles, the joint torque sensors are used to compute an estimate of the earth gravity direction which allows to stabilize the absolute roll and pitch angles using an error state Kalman filter.

D. Summary of the Sensors

The IMU is independent from environmental conditions and provides measurements at a high frequency. Since it has a strong drift in position and orientation, it must be aided by other sensors that do not suffer from drift. Aiding measurements are provided by leg odometry and visual odometry. These are relative measurements which are drift-free. Furthermore, leg odometry and visual odometry can be considered as complementary because usually rough terrain, where leg odometry is prone to slip, has good texture and allows accurate visual odometry measurements, and vice versa.

III. FILTER CHOICE FOR MULTISENSOR DATA FUSION

To combine the measurements of IMU, visual and leg odometry, a multisensor data fusion algorithm has been

implemented. Usually probabilistic estimators such as the Kalman filter or its inverse formulation, the information filter, are used for such applications. In this application, an indirect feedback information filter is used. The information filter has the advantage that fusing measurements of multiple sensors at the same time can be achieved very easily. The indirect or error state form works on an error state vector which contains the errors of the actual state rather than the state variables themselves. The advantage is that no model of the usually nonlinear robot dynamics is required but the filter is based on linear equations describing the error propagation in the inertial system. The feedback formulation means that the estimated error is fed back into the IMU navigation equations to correct the current position, velocity and orientation estimates. For this, the estimated error states are kept small and small angle approximations in the filter equations are possible. That also means that the error state can be predicted as zero for each new filter step. Furthermore, the indirect filter formulation allows the filter to be run at a lower frequency than the inertial navigation equations. For a more detailed discussion of the different filter formulations the reader is referred to Roumeliotis et al. [10].

The information filter is numerically equivalent to the Kalman filter but has inverse mathematical properties. To transform the indirect Kalman filter into the information form, the information matrix \mathbf{Y} and the error information vector $\Delta\mathbf{y}$ are defined as

$$\mathbf{Y} = \mathbf{P}^{-1} \quad \text{and} \quad \Delta\mathbf{y} = \mathbf{Y} \cdot \Delta\mathbf{x}, \quad (3)$$

where \mathbf{P} is the estimation covariance matrix and $\Delta\mathbf{x}$ is the error state vector. Transforming the Kalman filter equations so that \mathbf{Y} and $\Delta\mathbf{y}$ are estimated results in the prediction step

$$\mathbf{Y}_t^- = (\mathbf{A}_t \mathbf{Y}_{t-1}^- \mathbf{A}_t^T + \mathbf{Q}_t^p)^{-1} \quad (4)$$

$$\Delta\mathbf{y}_t^- = \mathbf{Y}_t^- (\mathbf{A}_t \mathbf{Y}_{t-1}^- \Delta\mathbf{y}_{t-1}^-), \quad (5)$$

where \mathbf{A}_t is the state transition matrix and \mathbf{Q}_t^p is the process noise matrix. In the feedback form, the prediction (5) can be simplified to $\Delta\mathbf{y}_t^- = 0$ because it is assumed that the error is corrected after each filter step. The update step of the information filter becomes

$$\mathbf{Y}_t = \mathbf{H}_t^T (\mathbf{Q}_t^m)^{-1} \mathbf{H}_t + \mathbf{Y}_t^- \quad (6)$$

$$\Delta\mathbf{y}_t = \mathbf{H}_t^T (\mathbf{Q}_t^m)^{-1} \mathbf{z}_t + \Delta\mathbf{y}_t^-, \quad (7)$$

where \mathbf{H}_t is the measurement matrix and \mathbf{Q}_t^m is the measurement noise matrix. In the indirect formulation, the measurement vector \mathbf{z}_t is the difference between the IMU measurements and the measurements of an aiding sensor. The update step can be written as

$$\mathbf{Y}_t = \mathbf{I}_t + \mathbf{Y}_t^-, \quad \text{with} \quad \mathbf{I}_t = \mathbf{H}_t^T (\mathbf{Q}_t^m)^{-1} \mathbf{H}_t, \quad (8)$$

$$\Delta\mathbf{y}_t = \mathbf{i}_t + \Delta\mathbf{y}_t^-, \quad \text{with} \quad \mathbf{i}_t = \mathbf{H}_t^T (\mathbf{Q}_t^m)^{-1} \mathbf{z}_t. \quad (9)$$

The term \mathbf{I}_t is the amount of information in the measurement and \mathbf{i}_t is the contribution of the measurement \mathbf{z}_t to the state vector [3]. If there are several measurements $\mathbf{z}_{k,t}$ at

a timestep t we get

$$\mathbf{I}_t = \sum_{k=1}^n \mathbf{H}_{k,t}^T (\mathbf{Q}_{k,t}^m)^{-1} \mathbf{H}_{k,t} = \sum_{k=1}^n \mathbf{I}_{k,t} \quad (10)$$

$$\mathbf{i}_t = \sum_{k=1}^n \mathbf{H}_{k,t}^T (\mathbf{Q}_{k,t}^m)^{-1} \mathbf{z}_{k,t} = \sum_{k=1}^n \mathbf{i}_{k,t}. \quad (11)$$

The simplicity of the update stage of the information filter originates from the fact, that the measurements of the single sensors are conditionally independent. Hence, the information form of the Kalman filter has computational advantages for multisensor data fusion. The routines for computing $\mathbf{I}_{k,t}$ and $\mathbf{i}_{k,t}$ for each measurement are independent of each other and independent of \mathbf{Y}_t^- and $\Delta\mathbf{y}_t^-$ and can run in parallel and on distributed systems. The disadvantage is, that a matrix inversion is required to obtain the error state vector $\Delta\mathbf{x}_t$ from the information vector $\Delta\mathbf{y}_t$. However, the more external sensors are used, the higher the profit when using the information filter.

IV. STATE VECTOR AND STATE TRANSITION MODEL

For implementing the information filter we chose to use a state vector consisting of 15 variables: The position \mathbf{p} (3), the velocity \mathbf{v} (3), the orientation Euler angles φ (3), the bias of the gyroscopes \mathbf{b}_g (3) and the bias of the accelerometers \mathbf{b}_a (3). In the indirect formulation the error state vector

$$\Delta\mathbf{x} = (\Delta\mathbf{p}, \Delta\mathbf{v}, \Delta\varphi, \Delta\mathbf{b}_g, \Delta\mathbf{b}_a)^T \quad (12)$$

is used. The position \mathbf{p} and velocity \mathbf{v} variables are given in world coordinates with the origin located at the IMU origin at the beginning of the data fusion process. The Euler angles φ are the angles of the rotation matrix that turns a point from the IMU coordinate system to the world coordinate system. The bias values \mathbf{b}_g and \mathbf{b}_a are given in IMU coordinates.

The discrete time error state propagation originates from the inertial error dynamics [11] as

$$\Delta\mathbf{x}_t^- = \mathbf{A}_t \cdot \Delta\mathbf{x}_{t-1} \quad (13)$$

$$\mathbf{A}_t = \mathbf{I} - \begin{bmatrix} \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_t^- [(a_t - \mathbf{b}_{a,t}^-) \times] & \mathbf{0} & \mathbf{R}_t^- \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_t^- & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \Delta t \quad (14)$$

$$[\mathbf{o} \times] = \begin{bmatrix} 0 & -o_z & o_y \\ o_z & 0 & -o_x \\ -o_y & o_x & 0 \end{bmatrix}, \quad (15)$$

where \mathbf{I} is the identity matrix (not to confuse with the information amount \mathbf{I}_t), $\mathbf{a}_t = (a_{tx}, a_{ty}, a_{tz})^T$ is the acceleration measured by the IMU, $\mathbf{b}_{a,t}^-$ is the predicted accelerometer bias, \mathbf{R}_t^- is the propagated rotation from the IMU coordinate system into the world coordinate system and Δt is the time difference between $t-1$ and t .

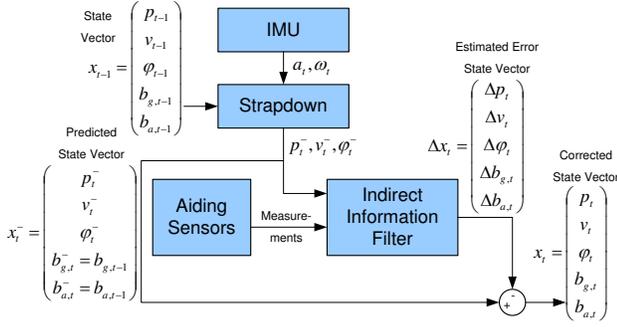


Fig. 2. Overview of the multisensor data fusion process

V. THE MULTISENSOR DATA FUSION PROCESS

An overview of one time step of the data fusion process is given in Fig. 2. First, the accelerations \mathbf{a}_t and angular velocities $\boldsymbol{\omega}_t$ measured by the IMU are fed into a strapdown algorithm. Considering the state vector \mathbf{x}_{t-1} from the previous filter step, this algorithm integrates the IMU measurements to velocity \mathbf{v}_t^- , position \mathbf{p}_t^- and orientation Euler angles $\boldsymbol{\varphi}_t^-$. These values are the predicted state variables. The bias values $\mathbf{b}_{a,t}^-$ and $\mathbf{b}_{g,t}^-$ are predicted to be equal to the bias values of the last filter step. Every time one or more measurements of the aiding sensors are available, the indirect information filter is run and gives an estimated error state vector $\Delta \mathbf{x}_t$. This error state vector is then subtracted from the predicted state vector \mathbf{x}_t^- to feedback the error. The result is the corrected state vector \mathbf{x}_t . If no measurements of the aiding sensors are available, the error state vector is zero and the corrected state will be the predicted state. The strapdown block and the information filter block are described in more detail in the following sections.

A. The Strapdown Algorithm

The accelerations and angular velocities of the IMU are measured in the IMU coordinate system. Since the IMU moves, the accelerations have to be transformed into the world coordinate system before integrating them. For this, the rotation matrix \mathbf{R}_t , which turns a vector from the IMU coordinate system into the world coordinate frame, has to be computed.

The rotation matrix can be propagated using the gyroscope measurements $\boldsymbol{\omega}_t$. Assuming a high sampling rate (Δt is small), the propagation of the rotation matrix can be performed as follows [12]:

$$\mathbf{R}_t^- = \mathbf{R}_{t-1} \mathbf{R}_{\Delta,t} \quad (16)$$

$$\mathbf{R}_{\Delta,t} = \mathbf{I} + \frac{\sin|\boldsymbol{\phi}_t|}{|\boldsymbol{\phi}_t|} [\boldsymbol{\phi}_t \times] + \frac{1 - \cos|\boldsymbol{\phi}_t|}{|\boldsymbol{\phi}_t|^2} [\boldsymbol{\phi}_t \times]^2 \quad (17)$$

$$|\boldsymbol{\phi}_t| = \sqrt{\phi_{x,t}^2 + \phi_{y,t}^2 + \phi_{z,t}^2} \quad (18)$$

$$\boldsymbol{\phi}_t = (\boldsymbol{\omega}_t - \mathbf{b}_{g,t}^-) \Delta t. \quad (19)$$

\mathbf{R}_t^- is the propagated rotation matrix, which is computed from the rotation matrix \mathbf{R}_{t-1} of the last time step and a differential rotation $\mathbf{R}_{\Delta,t}$. The variable $\boldsymbol{\phi}_t$ is the rotation vector.

Knowing the rotation matrix, the IMU velocity \mathbf{v}_t^- and position \mathbf{p}_t^- can be computed. The acceleration measurements \mathbf{a}_t have to be compensated for bias $\mathbf{b}_{a,t}^-$, transformed into the world frame using \mathbf{R}_t^- and the gravity vector $\mathbf{g} = (0, 0, -9.80665)^T$ must be compensated:

$$\mathbf{v}_t^- = \mathbf{v}_{t-1} + (\mathbf{R}_t^- (\mathbf{a}_t - \mathbf{b}_{a,t}^-) + \mathbf{g}) \Delta t \quad (20)$$

$$\mathbf{p}_t^- = \mathbf{p}_{t-1} + \mathbf{v}_{t-1} \Delta t + \frac{1}{2} (\mathbf{R}_t^- (\mathbf{a}_t - \mathbf{b}_{a,t}^-) + \mathbf{g}) \Delta t^2. \quad (21)$$

B. The Indirect Information Filter

Within the indirect information filter relative and absolute measurements are used to compute the estimated error state vector. Relative measurements contain a difference between the current system state and a previous state. Since Kalman filter theory assumes that a measurement only depends on the current state of the system, the state vector and covariance matrix have to be augmented to also contain the previous state which is part of the relative measurement. This approach was described by Roumeliotis et al. [13]. This method introduces the correlations between the current and the previous state and allows to estimate a correct covariance matrix, which grows with time if only relative measurements are available.

To keep the augmented covariances small, we chose to only clone the covariances associated to the states \mathbf{p}_t and $\boldsymbol{\varphi}_t$, because only relative position and rotation measurements are used. At each time $t = t_{\text{Start}}$ when at least one relative measurement starts, the covariance matrix is augmented as follows:

$$\tilde{\mathbf{x}}_t = \begin{bmatrix} \mathbf{p}_t \\ \boldsymbol{\varphi}_t \end{bmatrix} \quad \check{\mathbf{P}}_t = \mathbf{Cov}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_t) \quad (22)$$

$$\mathbf{P}_t^{\text{aug}} = \begin{bmatrix} \check{\mathbf{P}}_{t_{\text{Start}}} & \mathbf{Cov}(\tilde{\mathbf{x}}_{t_{\text{Start}}}, \mathbf{x}_t) \\ \mathbf{Cov}(\mathbf{x}_t, \tilde{\mathbf{x}}_{t_{\text{Start}}}) & \mathbf{P}_t \end{bmatrix}, \quad (23)$$

where $\mathbf{Cov}(\mathbf{x}_t, \tilde{\mathbf{x}}_{t_{\text{Start}}})$ is the covariance between the states at time t and the cloned states at t_{Start} . Since the covariance $\check{\mathbf{P}}_{t_{\text{Start}}}$ must not change during prediction of the filter, the system matrix $\mathbf{A}_t^{\text{aug}}$ and the process noise matrix $\mathbf{Q}_t^{p,\text{aug}}$ become

$$\mathbf{A}_t^{\text{aug}} = \mathbf{blkdiag}[\mathbf{I}, \mathbf{A}_t] \quad (24)$$

$$\mathbf{Q}_t^{p,\text{aug}} = \mathbf{blkdiag}[\mathbf{0}, \mathbf{Q}_t^p], \quad (25)$$

where $\mathbf{blkdiag}[\mathbf{U}, \mathbf{V}]$ stands for a block diagonal matrix with the matrices \mathbf{U}, \mathbf{V} on its main diagonal.

Since in the information filter the inverse covariance is used, it must be ensured that in the prediction step (4) $\mathbf{A}_t^{\text{aug}} \mathbf{P}_t^{\text{aug}} (\mathbf{A}_t^{\text{aug}})^T + \mathbf{Q}_t^{p,\text{aug}}$ is invertible. For that reason, if two different relative measurements start at the same time, cloning is applied only once to keep the covariance matrix invertible. If measurements start at different times, the covariances between the measurements also have to be cloned correctly. At the end of each relative measurement, the corresponding covariances are deleted from the augmented covariance matrix. However, in this application, usually a relative measurement starts at the same time the previous measurement ends.

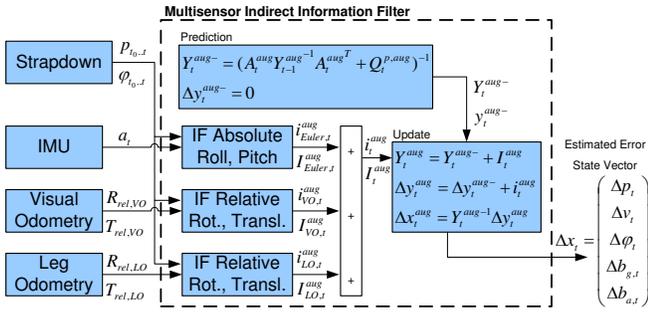


Fig. 3. Overview of the multisensor data fusion information filter

The data flow within the indirect information filter for multisensor data fusion is shown in Fig. 3. First, the augmented information matrix is predicted. Then, for each available sensor measurement k at time step t the values for $i_{k,t}^{aug}$ and $I_{k,t}^{aug}$ are computed using the differences between the strapdown algorithm results and the sensor measurements. Then, all available information amounts $I_{k,t}^{aug}$ and information contributions $i_{k,t}^{aug}$ are summed up and the update equations are performed. In the end, the resulting information vector is transformed into an error state vector from which the cloned states are deleted. These steps are described in more detail in the following sections.

1) *Prediction*: Using the state transition matrix A_t^{aug} as given in (14) and (24), the information matrix Y_t^{aug-} is predicted using (5). The prediction of the information vector simply becomes $\Delta y_t^{aug-} = 0$ because in the indirect feedback information filter the error is corrected after each filter step.

2) *Absolute Roll and Pitch Angle Measurements*: Since the accelerometers of the IMU sense the gravity, which is known in size and direction with respect to the world frame, it is possible to determine the absolute roll and pitch angles γ_{abs} and β_{abs} of the acceleration measurement $\mathbf{a} = [a_x, a_y, a_z]^T$ as follows:

$$\gamma_{abs} = \text{atan2}(a_y, a_z), \quad (26)$$

$$\beta_{abs} = \text{atan2}(-a_x, a_y \sin \gamma_{abs} + a_z \cos \gamma_{abs}). \quad (27)$$

From the absolute roll and pitch angles, an absolute rotation matrix \mathbf{R}_{abs} can be computed using

$$\mathbf{R}_{abs} = \begin{bmatrix} c\beta c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ c\beta s\alpha & s\gamma s\beta s\alpha + c\gamma c\alpha & c\gamma s\beta s\alpha - s\gamma c\alpha \\ -s\beta & s\gamma c\beta & c\gamma c\beta \end{bmatrix} \quad (28)$$

$$s\varphi = \sin \varphi_{abs} \quad c\varphi = \cos \varphi_{abs}$$

For this, the yaw angle α_{abs} is set to be equal to the yaw angle of the propagated rotation matrix \mathbf{R}_t^- because it cannot be determined from the acceleration measurements.

However, if the IMU is in motion, there are additional accelerations which disturb the computation of the absolute orientation. From the propagated rotation matrix \mathbf{R}_t^- as computed in (17)-(19), the roll and pitch angles can also be determined but these values suffer from a drift. By bringing \mathbf{R}_{abs} and \mathbf{R}_t^- together, the roll and pitch Euler

angles can be determined quite accurately. The difference rotation matrix between the propagated rotation \mathbf{R}_t^- and the absolute rotation \mathbf{R}_{abs} is computed as

$$\mathbf{R}_{diff} = \mathbf{R}_t^- \cdot \mathbf{R}_{abs}^T. \quad (29)$$

Using the equations

$$\begin{aligned} \alpha &= \text{atan2}(\mathbf{R}_{(2,1)}, \mathbf{R}_{(1,1)}) \\ \beta &= \text{atan2}(-\mathbf{R}_{(3,1)}, \mathbf{R}_{(2,1)} \sin \alpha + \mathbf{R}_{(1,1)} \cos \alpha) \\ \gamma &= \text{atan2}(\mathbf{R}_{(1,3)} \sin \alpha - \mathbf{R}_{(2,3)} \cos \alpha, \\ &\quad -\mathbf{R}_{(1,2)} \sin \alpha + \mathbf{R}_{(2,2)} \cos \alpha) \end{aligned} \quad (30)$$

to extract Euler angles from the elements $\mathbf{R}_{(i,j)}$ of a rotation matrix, the angle differences γ_{diff} and β_{diff} can be computed from \mathbf{R}_{diff} which give the measurement vector $\mathbf{z}_{Euler,t}$:

$$\mathbf{z}_{Euler,t} = \begin{bmatrix} \gamma_{diff} \\ \beta_{diff} \end{bmatrix} \quad (31)$$

The Euler angle gimbal lock problem can be neglected because such orientations will not be reached by the robot.

The measurement matrix $\mathbf{H}_{Euler,t}$ which projects the state vector \mathbf{x}_t onto the measurement vector $\mathbf{z}_{Euler,t}$ is

$$\mathbf{H}_{Euler,t} = [\mathbf{0}_{2 \times 6} \quad \mathbf{I}_{2 \times 2} \quad \mathbf{0}_{2 \times 7}] \quad (32)$$

For the augmented state vector, the measurement matrix has to be augmented with zeros to

$$\mathbf{H}_{Euler,t}^{aug} = [\mathbf{0} \quad \mathbf{H}_{Euler,t}], \quad (33)$$

because the measurement does not depend on any previous states but is absolute. The measurement noise matrix $\mathbf{Q}_{Euler,t}^m$ contains the variances of the absolute roll and pitch angle measurements and can be found by filter tuning.

Knowing $\mathbf{z}_{Euler,t}$, $\mathbf{H}_{Euler,t}^{aug}$, $\mathbf{Q}_{Euler,t}^m$ the information contribution $i_{Euler,t}^{aug}$ and the information amount $I_{Euler,t}^{aug}$ are computed using (8)-(9).

3) *Relative Translation and Rotation Measurements*: The relative motion measurements provided by visual and leg odometry have to be fused with the relative rotations and translations computed by the strapdown algorithm within the same time period. A relative measurement, thus, has two timestamps t_{start} and t_{end} at the beginning and the end of the relative motion. Furthermore, for fusing relative rotations and translations, all values must be represented in the same coordinate system. That means, the relative measurements of all sensors have to be transformed into relative measurements in the IMU coordinate frame in order to be fused with IMU measurements. That also means, that the transformations between the different sensor coordinate frames must be known, either by design or by calibration.

The differences between the relative motion given by the strapdown algorithm in the time interval from t_{start} to t_{end} and the relative motion measured by the odometry sensor give the measurement vector $\mathbf{z}_{rel,t}$. In order to compute the difference between two relative rotations \mathbf{R}_{rel}^I measured by the IMU and \mathbf{R}_{rel}^S measured by an odometry sensor, an absolute rotation matrix has to be computed. To preserve the relative character of the measurements, both relative rotations

have to be multiplied with the same absolute rotation matrix $\mathbf{R}_{t_{\text{start}}}$ to get pseudo-absolute rotation measurements. This absolute rotation matrix $\mathbf{R}_{t_{\text{start}}}$ should be the best estimate of the rotation from the IMU into the world frame at time step t_{start} :

$$\mathbf{R}_{t_{\text{end}}}^I = \mathbf{R}_{t_{\text{start}}} \mathbf{R}_{\text{rel}}^I, \quad \mathbf{R}_{t_{\text{end}}}^S = \mathbf{R}_{t_{\text{start}}} \mathbf{R}_{\text{rel}}^S. \quad (34)$$

Now the rotational difference matrix can be computed as

$$\mathbf{R}_{\text{diff}} = \mathbf{R}_{t_{\text{end}}}^I \cdot (\mathbf{R}_{t_{\text{end}}}^S)^T. \quad (35)$$

The measurement vector $\mathbf{z}_{\text{rel},t}$ contains the differences \mathbf{p}_{diff} between the two relative translations and the angle differences φ_{diff} computed from \mathbf{R}_{diff} using (30):

$$\mathbf{z}_{\text{rel},t} = [\mathbf{p}_{\text{diff}}, \varphi_{\text{diff}}]^T. \quad (36)$$

The augmented measurement matrix $\mathbf{H}_{\text{rel},t}^{\text{aug}}$ which projects the augmented state vector $\Delta \mathbf{x}_t^{\text{aug}}$ onto the measurement vector $\mathbf{z}_{\text{rel},t}$ is

$$\mathbf{H}_{\text{rel},t}^{\text{aug}} = \begin{bmatrix} -\mathbf{H}_{\text{rel},t_{\text{start}}} & \mathbf{H}_{\text{rel},t} \end{bmatrix}. \quad (37)$$

$$\mathbf{H}_{\text{rel},t} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 6} \end{bmatrix}. \quad (38)$$

The matrix $\mathbf{H}_{\text{rel},t_{\text{start}}}$ contains an identity matrix in the columns corresponding to the location of the cloned covariance of time t_{start} and zeros everywhere else.

The measurement noise matrix $\mathbf{Q}_{\text{rel},t}^m$ is computed from the standard deviations of the relative position and rotation measurements. For leg odometry, the measurements errors depend on how much the feet of the robot slip on the ground. On homogeneous ground the amount of slippage can be assumed constant and found by filter tuning. Using visual odometry, assuming constant standard deviations for the relative motion measurements is not appropriate for environments with changing light or texture conditions. Thus, the estimated standard deviations of each visual odometry measurement (ref. Sec. II-B.2) are transformed into the IMU coordinate system using error propagation and then fed into the measurement noise matrix.

Knowing $\mathbf{z}_{\text{rel},t}$, $\mathbf{H}_{\text{rel},t}^{\text{aug}}$, $\mathbf{Q}_{\text{rel},t}^m$ the information contribution $\mathbf{i}_{\text{rel},t}^{\text{aug}}$ and the information amount $\mathbf{I}_{\text{rel},t}^{\text{aug}}$ are computed using (8)-(9).

4) *Update*: At every time step, $\mathbf{i}_{k,t}^{\text{aug}}$ and $\mathbf{I}_{k,t}^{\text{aug}}$ of each available sensor measurement are computed. In the final step of the multisensor information filter, these values are summed and used to update the predicted information vector and information matrix using (8)-(9). Finally, the resulting information vector $\Delta \mathbf{y}_t^{\text{aug}}$ is transformed into an error state vector $\Delta \mathbf{x}_t^{\text{aug}}$. After deleting the augmented states from this vector, $\Delta \mathbf{x}_t$ is obtained, which contains the estimated errors of the single robot states. By inverting the resulting information matrix $\mathbf{Y}_t^{\text{aug}}$ and deleting the augmented covariances, the covariance matrix \mathbf{P}_t can be computed if required.

C. Error State Feedback

To correct the position, velocity and bias values of the predicted state vector \mathbf{x}_t^- , the corresponding error estimates from the error state vector $\Delta \mathbf{x}_t$ are subtracted. For feeding back the estimated rotation angle error $\Delta \varphi_t$, a rotation matrix \mathbf{R}_{corr} has to be computed from $\Delta \varphi_t$ using (28) and correction is performed as

$$\mathbf{R}_t = \mathbf{R}_{\text{corr}}^T \cdot \mathbf{R}_t^-. \quad (39)$$

From \mathbf{R}_t the corrected Euler angles can be extracted via (30).

VI. FILTER INITIALIZATION

In the beginning of the data fusion process the robot is motionless in its starting position. This phase can be used for filter initialization.

From the very first IMU measurement, the starting orientation \mathbf{R}_{t_0} with respect to the gravity vector is estimated from the acceleration measurements \mathbf{a}_{t_0} as shown in (26)-(27). Furthermore, the bias estimates \mathbf{b}_{a,t_0} and \mathbf{b}_{g,t_0} are initialized using the starting orientation, the known gravity vector \mathbf{g} and the gyroscope measurements $\boldsymbol{\omega}_{t_0}$. Good starting values for the sensor biases are

$$\mathbf{b}_{a,t_0} = \mathbf{a}_{t_0} + \mathbf{R}_{t_0}^T \cdot \mathbf{g} \quad (40)$$

$$\mathbf{b}_{g,t_0} = \boldsymbol{\omega}_{t_0}. \quad (41)$$

From the following IMU measurements, the estimates of the bias values and the starting orientation can be refined exploiting the fact that the robot does not move. Hence, position, velocity and orientation measurements with the value of zero and small noise matrices are fed into the information filter. As a result the bias value estimation stabilizes. Furthermore, the absolute roll and pitch angle measurements from the accelerations are fused with the orientation measurements from the gyroscopes as described in section V-B.2. The initialization phase is finished when the change in the bias estimates drops below a threshold. This process usually takes a few seconds. Once the information filter is initialized, the robot can start moving and visual odometry and leg odometry measurements are used.

VII. EXPERIMENTAL RESULTS

For evaluating the performance of the multisensor data fusion filter, the Crawler was steered along a rectangular path through a 2×2 m testbed filled with gravel. It was controlled manually via a 6 DOF space mouse generating the commands “walk forward”, “turn left/right” and “walk sideways to the left/right”. Its walking speed was approximately 0.04 m/s. The estimated trajectories measured by visual odometry and leg odometry, as well as the trajectory estimated by fusing inertial, visual and leg odometry data were recorded. Additionally, a reflecting target body was mounted on the Crawler and tracked by an infrared tracking system. The trajectory of the target body provided a ground truth measurement.

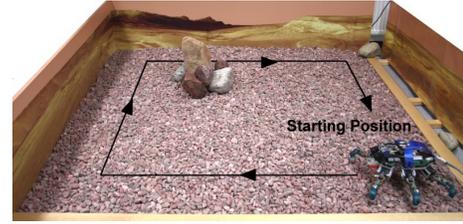
In one run, the lighting conditions and the texture of the gravel were very good. Fig. 4(a) shows the test setup with

the robot in its starting pose and the approximate steered path. Fig. 4(b) shows the ground truth trajectory measured by the tracking system, the fusion result and the different odometry trajectories which were obtained by summing the relative measurements of the respective sensors. The trajectory computed using only the IMU measurements is not shown here because its enormous drift leads to an error of more than 100 m after 60 s runtime. The visual odometry trajectory is quite accurate apart from a small drift of the yaw angle. The leg odometry trajectory shows that yaw angles are overestimated because of slip in the gravel (ref. Fig. 4(d)). The fusion trajectory is very close to the ground truth path. Fig. 4(c) shows plots of the z-coordinates. While visual odometry and leg odometry drift due to roll and pitch angle errors, the estimated z-coordinate of the fusion result remains close to the ground truth curve because of absolute roll and pitch angle measurements from the accelerometers. Fig. 4(e) shows the standard deviations of the position estimates computed from the estimation covariance matrix. As can be seen, the standard deviations grow with time since no absolute position measurements are available. The detailed plot in this figure illustrates the influence of the relative measurements on the covariance: Visual odometry measurements usually have lower uncertainty than leg odometry measurements and, thus, reduce the estimation covariance more than the leg odometry measurements. However, during turning in the corners of the testbed, the errors of the visual odometry measurements are higher. The reason for that is the texture of the testbed walls which is worse than the texture of the gravel. Hence, the covariances increase stronger during these periods.

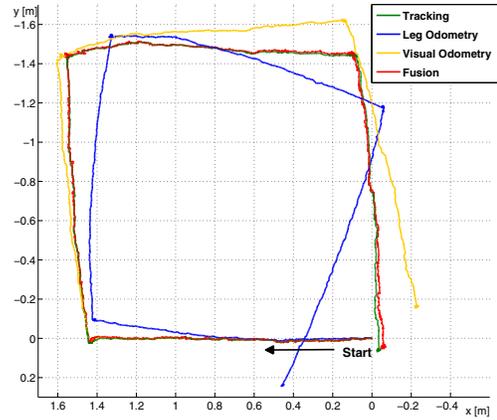
In another test, poor visual conditions were simulated. For this, one corner of the testbed was illuminated by a very bright light source (ref. Fig. 5(a)) and the camera was set to fixed exposure so that images taken from the illuminated area were overexposed and nearly white while all other images were well exposed. Sample images taken by the Crawler's camera are shown in Fig. 5(b). Fig. 5(c)-(d) show the recorded trajectories. Leg odometry suffers from a yaw angle error (ref. Fig. 5(e)) and overestimates forward translation. Visual odometry is very accurate in areas with good lighting. However, large visual odometry errors occur in the illuminated corner. These errors cause the visual odometry trajectory to continue in a wrong direction because of the summation of relative measurements. Since the error of the visual odometry measurements was estimated to be very high in this region, the data fusion filter put a lower weight on these measurements. Hence, the fusion result is not affected by the large visual odometry errors and is very accurate.

To achieve more general information about the performance of the data fusion filter, we computed the end point errors of the recorded trajectories as the distances to the ground truth trajectory end points. The results for 9 runs along a rectangular path in the gravel testbed with different lighting conditions are shown in Fig. 6. As can be seen, the fusion result is always significantly better than using only

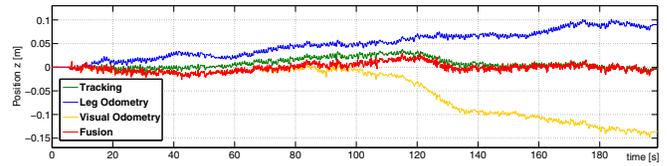
visual odometry or leg odometry. The average endpoint error of the fusion trajectory is 1.1 % in relation to the average path length of 5.6 m.



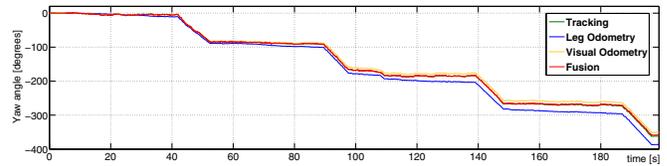
(a) Test setup and steered trajectory



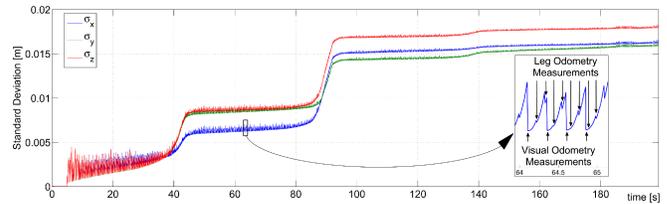
(b) Recorded trajectories



(c) z-coordinates



(d) Yaw angles

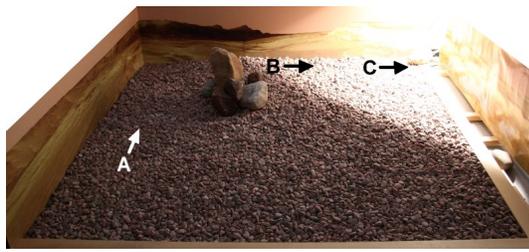


(e) Standard deviations computed from the estimation covariance matrix

Fig. 4. Test run with good visual conditions

VIII. CONCLUSION AND FUTURE WORK

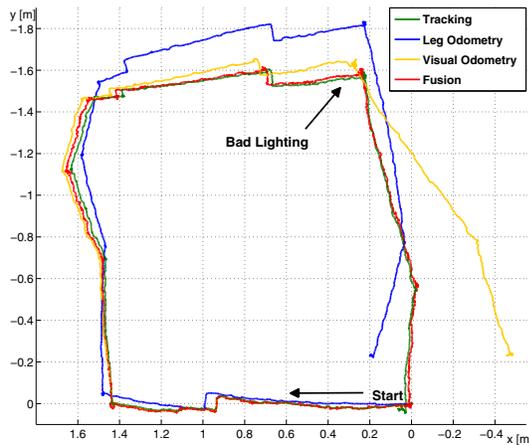
In this paper, a multisensor data fusion algorithm for combining inertial data with relative leg odometry and visual odometry measurements has been presented. It is based on an indirect feedback information filter whose special properties make it well suited for this application. Estimates of



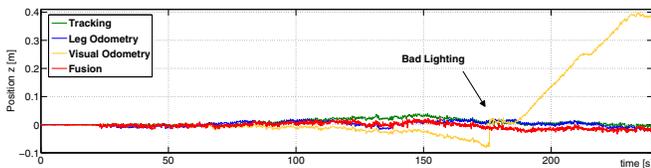
(a) Test setup



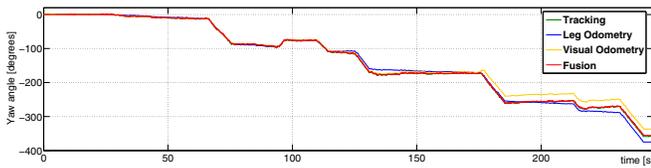
(b) Sample images



(c) Recorded trajectories



(d) z-coordinates



(e) Yaw angles

Fig. 5. Test run with poor lighting conditions

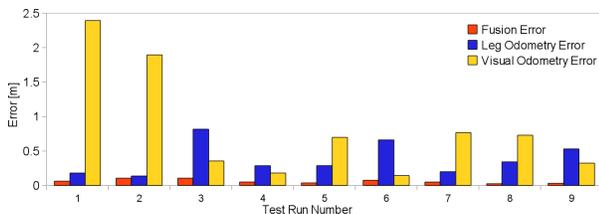


Fig. 6. End point errors of fusion, visual and leg odometry trajectories. Good visual conditions only in runs 4, 6 and 9.

visual odometry errors are used to weight the measurements differently according to the current visual conditions.

Experimental results show that the data fusion algorithm improves robustness as well as accuracy of the pose estimate compared to using only a single sensor. The gravity vector allows an absolute roll and pitch angle measurement which limits the drift in the position estimate. The visual odometry error estimate reliably detects ill-conditioned measurements as caused by bad visual conditions and the data fusion filter puts a lower weight on those measurements. Using the leg odometry measurements, the robot is able to overcome visually poor areas with a good accuracy.

However, it can also be seen, that the position estimate still suffers from a drift caused by summing up translational and yaw angle errors in the relative motion estimates. This error can only be eliminated by using absolute measurements such as compass or GPS data (where available) or landmark positions. In future, we would also like to develop an error model for the leg odometry so that incorrect odometry measurements are assigned lower weights in the fusion process.

REFERENCES

- [1] M. Görner, T. Wimböck, and G. Hirzinger, "The DLR Crawler: Evaluation of gaits and control of an actively compliant six-legged walking robot," *Industrial Robot: An International Journal*, vol. 36, no. 4, pp. 344–351, 2009.
- [2] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.
- [3] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, 2002.
- [4] K. Konolige, M. Agrawal, and J. Sola, "Large scale visual odometry for rough terrain," in *Proc. International Symposium on Robotics Research*, 2007.
- [5] D. Helmick, Y. Cheng, D. Clouse, L. Matthies, and S. Roumeliotis, "Path following using visual odometry for a Mars rover in high-slip environments," in *IEEE Aerospace Conference*, 2004, pp. 772–789.
- [6] P. Lamon and R. Siegwart, "Inertial and 3d-odometry fusion in rough terrain-towards real 3d navigation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2004, pp. 1716–1721.
- [7] H. Hirschmüller, P. Innocent, and J. Garibaldi, "Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics," in *Proc. 7th International Conference on Control, Automation, Robotics and Vision*, December 2002, pp. 1099–1104.
- [8] H. Hirschmüller, "Stereo Processing by Semi-Global Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, February 2008.
- [9] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–248, 1987.
- [10] S. Roumeliotis, G. Sukhatme, and G. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 2, 1999, pp. 1656–1663.
- [11] J. Vasconcelos, P. Oliveira, and C. Silvestre, "Inertial Navigation System Aided by GPS and Selective Frequency Contents of Vector Measurements," in *Proc. AIAA Guidance, Navigation and Control Conference*, San Francisco, USA, August 2005.
- [12] J. Bortz, "A New Mathematical Formulation for Strapdown Inertial Navigation," *IEEE Transactions on Aerospace Electronic Systems*, vol. 7, pp. 61–66, 1971.
- [13] S. Roumeliotis and J. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *Proc. IEEE International Conference on Robotics and Automation*, 2002, vol. 2, 2002, pp. 1788–1795.