

# A Leg Proprioception Based 6 DOF Odometry for Statically Stable Walking Robots

<http://link.springer.com/article/10.1007/s10514-013-9326-3>

Martin Görner · Annett Stelzer

Received: date / Accepted: date

**Abstract** This article presents a 3D odometry algorithm for statically stable walking robots that only uses proprioceptive data delivered by joint angle and joint torque sensors embedded within the legs. The algorithm intrinsically handles each kind of emerging statically stable gait and is independent of predefined gait patterns. Additionally, the algorithm can be equally applied to stiff robots as well as to robots with compliant joints. Based on the proprioceptive information a 6 degrees of freedom (DOF) pose estimate is calculated in three steps. First, point clouds, represented by the foot positions with respect to the body frame at two consecutive time steps, are matched and provide a 6 DOF estimate for the relative body motion. The obtained relative motion estimates are summed up over time giving a 6 DOF pose estimate with respect to the start frame. Second, joint torque measurement based pitch and roll angle estimates are determined. Finally in a third step, these estimates are used to stabilize the orientation angles calculated in the first step by data fusion employing an error state Kalman filter. The algorithm is implemented and tested on the DLR Crawler, an actively compliant six-legged walking robot. For this specific robot, experimental data is provided and the performance is evaluated in flat terrain and on gravel, at different joint stiffness settings and for various emerging gaits. Based on this data, problems associated with the odometry of statically stable walking robots are identified and discussed. Further, some results for crossing slopes and edges in a complete 3D scenario are presented.

---

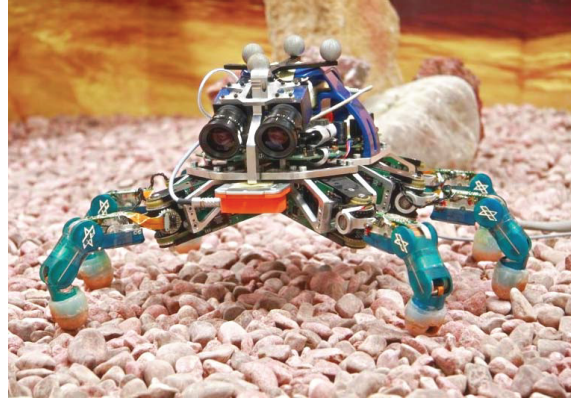
Martin Görner · Annett Stelzer  
Robotics and Mechatronics Center, German Aerospace Center (DLR e.V.), 82234 Wessling, Germany  
E-mail: martin.goerner@dlr.de

## 1 Introduction

As the technology progresses, heterogeneous groups of mobile robots operating at different levels of autonomy will be deployed in future terrestrial search and rescue missions as well as extra-terrestrial planetary exploration scenarios. Within these groups, legged robots with their high mobility and intrinsic manipulation capability are very well suited to serve as the rough terrain specialists for short range tasks. Their areas of operation involve environments such as debris cluttered urban terrain, natural and artificial caves, craters and canyons. To fulfill their tasks, the robots must be able to navigate autonomously. Usually, for the anticipated scenarios, no or incomplete a priori information, like maps or known landmarks, is available. Further, absolute reference as provided by GPS or an external magnetic field might be unavailable or extremely noisy. Thus, the robots cannot rely on these and must be able to acquire all necessary information about their state and their environment by themselves. For this purpose the robots must use all available data provided by their onboard sensors to the maximum extent. Since a robust pose estimate is crucial for autonomous navigation without reliable external reference, the robots need to optimally combine different means of ego-motion estimation. Often, inertial measurement unit (IMU) data and visual odometry are fused to provide these motion estimates. Using dead reckoning, i.e. the summation of motion increments over time, the robots keep track of their 6 degrees of freedom (DOF) poses with respect to a local frame. Usually, IMU position estimates are subject to strong drift and visual odometry is strongly dependent on the visual conditions. Thus, a 6 DOF leg odometry is a very useful additional source of information to enhance the robustness of the overall pose estimate.

In this article, we present such a leg odometry algorithm for statically stable walking robots, which in most cases are four-, six- or eight-legged. The algorithm provides a complete 6 DOF pose estimate which is solely based on proprioceptive joint angle and joint torque sensors embedded within the legs. This pose estimate is calculated by a three stage algorithm in order to exploit all the information delivered by the joint sensors. Herein, the first stage provides a 6 DOF relative motion estimate of the robot based on the kinematic configurations of its feet at two consecutive time steps. Summing up the relative motion increments results in a 6 DOF pose estimate with respect to a local start frame. Since this estimate is subject to a drift of the orientation angles, the second stage of the algorithm determines absolute pitch and roll angle estimates based on joint torque measurements. These absolute angle estimates are fused in the third stage with the pitch and roll angles of the first stage using an error state Kalman filter. Thus, only using the proprioceptive sensors embedded within the legs, a complete 6 DOF “pose sensor” is obtained that is independent of any IMU or exteroceptive sensor. The use of joint torque based pitch and roll angle estimates is usually not the first choice since most walking robots provide IMU based orientation data. Nevertheless, our intention is to present and analyze an alternative method that is based on leg sensors only and thus provides redundancy and robustness with respect to possible IMU failure. The presented algorithm is implemented and tested on a specific platform, the actively compliant, six-legged DLR Crawler shown in Fig. 1. As is true for all methods based on relative measurements, each leg odometry remains subject to some drift and thus is only useful within certain temporal and spatial bounds. Nevertheless, as robust pose estimation in real world navigation scenarios requires multiple means of pose measurement, leg odometry constitutes one possible source. As shown in [1] and [2], the leg odometry presented in this article is part of the visual navigation framework of the DLR Crawler. Herein, leg odometry as well as visual odometry are aiding sensors in an IMU based fusion scheme for robust pose estimation. The self-contained leg odometry provides redundancy and strongly enhances the robustness of the navigation in unknown terrain with respect to severe visual disturbances.

The article proceeds as follows. In section 2 we give a brief overview of the related literature. Section 3 introduces the method and presents the three stages of our algorithm. In section 4 experimental results are given for the DLR Crawler and the effects of different substrates, walking velocities and joint stiffness settings are



**Fig. 1** The DLR Crawler within the gravel testbed

evaluated for this specific platform. Finally, we conclude our work in section 5.

## 2 Related Literature

In wheeled robotics it is common practice to calculate an odometry based on wheel encoder readings and steering angles. Usually, wheel odometry returns a planar position as well as the heading angle of the vehicle. Only very few wheeled robots allow the calculation of an additional vertical motion estimate based on their kinematics. One example is the Shrimp robot developed at EPFL [3]. On this robot an advanced bogie concept provides the necessary information for the vertical motion estimate. Nevertheless, the pitch and roll angles, like on other wheeled robots, have to be estimated by use of an IMU.

In contrast to their wheeled counterparts, legged robots usually provide enough proprioceptive data from sensors embedded within their legs to calculate a complete 6 DOF pose estimate. However, due to the mechanical complexity, the high number of degrees of freedom and the high variety and variability of gaits, the problem is much harder. Only very few tested leg odometry examples returning a full 6 DOF pose are known to us. Each of those does rely on additional IMU data to either stabilize the results or to compensate for a missing degree of freedom of the pose due to kinematic constraints. The detailed work on the robot RHex [4, 5] is one of the few examples presented in literature. The robot consists of six equal, passively compliant, single degree of freedom legs and uses its hip joint encoder readings and leg deformation measurements to estimate its pose. Due to its kinematic configuration no yaw angle can be calculated by the basic odometry and the data needs to be fused with IMU readings to return a full 6 DOF pose estimate. One great advantage of this approach is that it also covers the flight phases

occurring during dynamic running. Another example of a leg odometry is an algorithm that was developed for the hexapod Ambler [6] and that was also implemented on the hexapod Lauron [7, 8]. In this approach, the supporting legs are used to determine a rigid body transformation for the robot with respect to the world frame. The algorithm assumes an ideal no slip ground contact and finds a minimum error transformation mapping the positions of the supporting feet at the current time step with respect to the body frame onto the stored positions of the supporting feet with respect to the world frame. After the minimizing transformation is found, the positions of the supporting feet in world coordinates are recalculated and updated if they changed. This is the case after a step but should not happen for legs in support according to the ideal no slip condition. In order to reduce the disturbing effect of slipping legs, individual leg weights are introduced for the transformation calculation. For Lauron these are obtained by a fuzzy reasoning based contact evaluation. The used leg odometry approach has problems with drifting pitch angle and height estimates. To improve the results the Ambler odometry discards the tilt angles and replaces them by inclinometer readings. To achieve improved results for Lauron, its odometry estimates are fused with IMU and magnetic compass based orientation data. As there is some performance data available for Ambler walking forward, unfortunately there is no detailed data published for Lauron. Two recent examples for the use of leg odometry with dynamic quadrupeds are presented by [9] and [10]. In the first article Reinstein et al. obtain a full pose estimate by fusing a velocity estimate based on leg odometry with data of an inertial navigation system using an Extended Kalman Filter. In the second article, Ma et al. published an approach to improve the navigation robustness of the robot BigDog and its successor project LS3 by multi-sensor data fusion using leg odometry [10].

Considering the computation of the relative body motion based on the location of the feet, our algorithm shows similarities to the approaches used for the robots Ambler and Lauron. The main difference is that our algorithm only employs proprioceptive sensors embedded within the legs to obtain a complete 6 DOF pose estimate that incorporates absolute pitch and roll information. This is achieved by stabilizing the joint angle based orientation estimates with joint torque measurement based absolute pitch and roll angles using an error state Kalman filter. Further, it is important that our method can be applied to stiff and compliant statically stable walking robots and is able to handle any statically stable emerging gait. Additionally, we identify error sources and introduce tuning parameters to

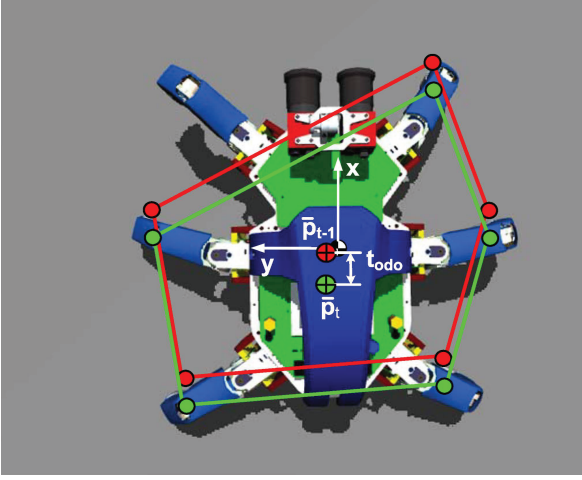
attenuate their effects. Compared to most of the previous work, the algorithm has been extensively tested and its performance is documented for the DLR Crawler walking on different ground substrates, at different joint stiffness settings as well as at different walking velocities and thus various emerging gait configurations.

### 3 Method

The basic idea of the presented 6 DOF leg odometry borrows from computer vision. It tries to estimate the change of the robot pose by finding a minimizing transformation between two point clouds which are represented by the positions of the supporting feet at two consecutive time steps. The algorithm requires at least three feet in contact and the assumption of rigid point clouds implies a no slip condition. To reduce errors caused by drifting orientation angles, absolute pitch and roll angles derived from joint torque measurements are fused with the joint angle based orientation estimates using an error state Kalman filter. Single slipping legs are detected and in case of a high sampling rate, the related measurements are discarded without strong influence on the pose estimate. Further, the method can handle any statically stable emerging gait and can be adapted to different ground substrates and different joint stiffness settings by some tuning parameters as documented for the DLR Crawler in section 4. The use of joint torque sensors for pose estimation seems to be an additional hardware effort that needs to be justified. The main reason for using such sensors in walking robots stems from a control perspective. Joint torque sensors allow the use of advanced active compliance control algorithms with underlying joint torque control loops and thus support a smooth and robust locomotion in rough terrain. Several recent examples of walking robots exist that employ joint torque sensors in addition to commonly used joint encoders [11], [12].

#### 3.1 First Stage: Joint Angle Based 6 DOF Pose Estimate

For a statically stable walking robot, as for example the DLR Crawler, a change of position and orientation is calculated using two consecutive stance feet configurations with respect to the body frame. The resulting relative motions of the robot body are summed up over time, yielding the pose of the robot with respect to the world frame. Herein, the world frame is based on the robot body frame at the start position that was aligned with the gravity vector along the z-axis. The body frame is defined as follows. The pos-



**Fig. 2** Foot contacts and their centroids for two consecutive time steps with respect to the body frame

itive x-direction points from the back to the front of the robot and the positive z-direction points upwards from its bottom to its top. The positive y-direction completes a right hand system and the origin is placed at the center of the robot body. All rotations follow the xyz convention with yaw angle  $\alpha$  defined about the z-axis, pitch angle  $\beta$  about the y-axis and roll angle  $\gamma$  about the x-axis.

In case of the DLR Crawler all calculations are performed at the high rate of 1 kHz, which allows discarding single relative motions in case of the detection of excessive slip. For statically stable walking robots, three or more legs will be in ground contact. Having more than three feet in stance results in an over-constraint problem and the odometry needs to account for this. Due to small errors in the kinematics and the rolling ground contacts of the feet, it is unlikely that the point clouds perfectly match. Thus, a rigid body transformation with the rotation matrix  $\mathbf{R}_{\text{odo}}$  and the translation vector  ${}^b\mathbf{t}_{\text{odo}}$  has to be found that minimizes the matching error. The detailed approach to solve this problem including the complete derivation was initially presented in a computer vision paper by Haralick [13]. Following, only the necessary equations are given for the purpose of completeness. The algorithm aims to minimize  $\epsilon$ , which is the sum of squared errors of the rigid body transformation augmented by constraints  $f(\boldsymbol{\lambda}, \mathbf{R}_{\text{odo}})$  that enforce an orthogonal rotation matrix,  $\mathbf{R}_{\text{odo}}\mathbf{R}_{\text{odo}}^T = \mathbf{I}$ .

$$\epsilon = \sum_{i=1}^n w_i (\| {}^b\mathbf{p}_{i,t} - \mathbf{R}_{\text{odo}} \cdot {}^b\mathbf{p}_{i,t-1} - {}^b\mathbf{t}_{\text{odo}} \|_2)^2 + f(\boldsymbol{\lambda}, \mathbf{R}) \quad (1)$$

Herein, the vectors  ${}^b\mathbf{p}_{i,t-1}$  and  ${}^b\mathbf{p}_{i,t}$  are the positions of a foot at two consecutive time steps with respect to the body frame  $b$ . These foot positions are calculated from joint angle measurements using forward kinematics. The parameter  $n$  is the number of legs of the robot, which is usually 4, 6 or 8. The parameters  $w_i$  are weights that are 1 if a foot is in a valid contact state or 0 otherwise. A valid contact state means that the foot has contact at both time steps and does not slip severely. The vector  $\boldsymbol{\lambda}$  consists of Lagrange multipliers for the six constraints.

Taking the partial derivative of  $\epsilon$  with respect to the translation vector  ${}^b\mathbf{t}_{\text{odo}}$  and setting it equal to zero results into the following equation.

$$\begin{aligned} {}^b\mathbf{t}_{\text{odo}} &= {}^b\bar{\mathbf{p}}_t - \mathbf{R}_{\text{odo}} {}^b\bar{\mathbf{p}}_{t-1} \\ &= \frac{\sum_{i=1}^n w_i {}^b\mathbf{p}_{i,t}}{\sum_{i=1}^n w_i} - \mathbf{R}_{\text{odo}} \frac{\sum_{i=1}^n w_i {}^b\mathbf{p}_{i,t-1}}{\sum_{i=1}^n w_i} \end{aligned} \quad (2)$$

The terms  ${}^b\bar{\mathbf{p}}_t$  and  ${}^b\bar{\mathbf{p}}_{t-1}$  can be considered as the centroids of the contact point clouds at two consecutive time steps as depicted in Fig. 2. Inserting  ${}^b\mathbf{t}_{\text{odo}}$  into (1) leaves  $\epsilon$  as a function of the elements of the rotation matrix and the Lagrange multipliers  $\boldsymbol{\lambda}$ . Taking now the partial derivative of  $\epsilon$  with respect to each element of the rotation matrix, setting it equal to zero,  $[\frac{\partial \epsilon}{\partial r_{\text{odo},(m,n)}}]_{3 \times 3} = 0_{3 \times 3}$ , and rearranging terms results into the following equation.

$$\mathbf{A}\mathbf{R}_{\text{odo}}^T + \mathbf{R}_{\text{odo}}\mathbf{A} = \mathbf{B} \quad (3)$$

Herein,  $\mathbf{A}$  is a symmetric matrix consisting of the six Lagrange multipliers and  $\mathbf{A}$  as well as  $\mathbf{B}$  are defined as follows.

$$\mathbf{A} = \sum_{i=1}^n w_i ({}^b\bar{\mathbf{p}}_{t-1} - {}^b\mathbf{p}_{i,t-1}) ({}^b\bar{\mathbf{p}}_{t-1} - {}^b\mathbf{p}_{i,t-1})^T \quad (4)$$

$$\mathbf{B} = [\mathbf{b}_x \ \mathbf{b}_y \ \mathbf{b}_z]_{3 \times 3} \quad (5)$$

$$\mathbf{b}_x = \sum_{i=1}^n w_i ({}^b\bar{p}_{x,t} - {}^b p_{i,x,t}) ({}^b\bar{\mathbf{p}}_{t-1} - {}^b\mathbf{p}_{i,t-1}) \quad (6)$$

$$\mathbf{b}_y = \sum_{i=1}^n w_i ({}^b\bar{p}_{y,t} - {}^b p_{i,y,t}) ({}^b\bar{\mathbf{p}}_{t-1} - {}^b\mathbf{p}_{i,t-1}) \quad (7)$$

$$\mathbf{b}_z = \sum_{i=1}^n w_i ({}^b\bar{p}_{z,t} - {}^b p_{i,z,t}) ({}^b\bar{\mathbf{p}}_{t-1} - {}^b\mathbf{p}_{i,t-1}) \quad (8)$$

Multiplying (3) with  $\mathbf{R}_{\text{odo}}$  from the left leaves an equation with symmetric left and right hand sides since  $\mathbf{A}$  and  $\mathbf{A}$  are symmetric matrices.

$$\begin{aligned} \mathbf{R}_{\text{odo}}\mathbf{B} &= \mathbf{R}_{\text{odo}}\mathbf{A}\mathbf{R}_{\text{odo}}^T + \mathbf{A} \\ &= (\mathbf{R}_{\text{odo}}\mathbf{B})^T \end{aligned} \quad (9)$$



Thus, a singular value decomposition of  $\mathbf{B}$  into the orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$  and the diagonal matrix  $\mathbf{D}$  allows the calculation of  $\mathbf{R}_{\text{odo}}$ , which can be verified by inserting (10) and (11) into (9).

$$\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (10)$$

$$\mathbf{R}_{\text{odo}} = \mathbf{V}\mathbf{U}^T \quad (11)$$

One important property of the solution for  $\mathbf{R}_{\text{odo}}$  is that the rotation matrix is only valid if its determinant is positive. If this is not true, the last column of the matrix  $\mathbf{V}$  has to be multiplied by -1 to deliver a valid result. Inserting  $\mathbf{R}_{\text{odo}}$  into (2) gives the relative translation of the robot. By propagating the relative rotation and translation the pose of the robot can be determined with respect to the world frame, i.e. relative to the gravity aligned frame at the start point of the robot.

In order to reduce errors accumulated over time leg slip should be detected and, if possible, the slipping leg should be discarded from the calculations. To assess the severity of the point cloud deformation due to slip, the quadratic error of the rigid body transformation for two consecutive time steps is calculated. If this error is higher than an acceptable threshold, the algorithm will try to identify the leg causing the strongest distortion. For this purpose, the relative distance of each leg in contact to each other stance leg is calculated and compared to the distances of the previous time step. If more than three legs are in contact, the leg with the largest change of distance to all other legs in contact is removed from the calculation by setting its weight equal to zero. In most cases this approach reduces the squared error of the transformation below the accepted threshold. If it does not suffice or only three legs are in contact, the remaining error will be compared to a second threshold and the rotation and translation during this time interval are either neglected or accepted. If many odometry calculations are neglected the odometry is invalidated. Nevertheless, slippage of the complete robot on a slope or on icy ground cannot be detected by this approach and remains a problem to be solved.

### 3.2 Second Stage: Joint Torque Measurement Based Pitch and Roll Angle Estimates

Due to the relatively slow velocity of statically stable walking robots (which is for the DLR Crawler on average below 10 cm/s), the main load measured by the joint torque sensors is assumed to originate from the gravitational force acting on the respective robot. Following the assumption of quasi-static behavior, the ground contact forces  $\mathbf{f}_i$  are calculated for each leg,

$i = 1 \dots n$ , based on the joint torques  $\boldsymbol{\tau}_i$  and the leg Jacobian  $\mathbf{J}_i$  using the well known static relation.

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \mathbf{f}_i \quad (12)$$

$$\mathbf{f}_i = (\mathbf{J}_i^T)^{-1} \boldsymbol{\tau}_i \quad (13)$$

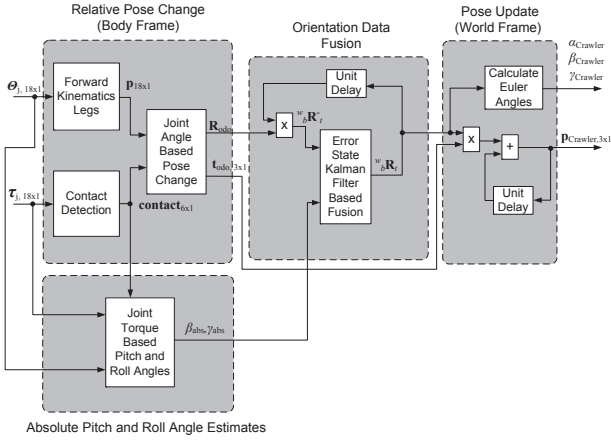
Summing up the leg contact forces  $\mathbf{f}_i$  yields the total ground contact force on the robot,  $\mathbf{f}$ , that is assumed to be mainly caused by gravity. Thus, the force vector given with respect to the body frame allows the calculation of the pitch angle  $\beta_{\text{abs}}$  and the roll angle  $\gamma_{\text{abs}}$  of the robot.

$$\gamma_{\text{abs}} = \text{atan2}(f_y, f_z) \quad (14)$$

$$\beta_{\text{abs}} = \text{atan2}(-f_x, f_y \sin(\gamma_{\text{abs}}) + f_z \cos(\gamma_{\text{abs}})) \quad (15)$$

### 3.3 Third Stage: Error State Kalman Filter Based Data Fusion

To improve the overall pose estimate, the joint angle based pitch and roll angle estimates of the first stage are fused with the joint torque measurement based pitch and roll angle estimates of the second stage using an error state Kalman filter in a feedback configuration as shown in Fig. 3. The fusion process realizes that the fast components of the joint angle based pitch and roll angle estimates are combined with the slow components of the joint torque based pitch and roll angle estimates. Thus, it removes the angle drift induced in the first stage of the algorithm and the ground impact related peaks of its second stage. It needs to be mentioned that the result is not anymore optimal in the sense of Kalman filter theory since in both stages the joint angle measurements are part of the calculations. Thus, the measurements are not anymore independent. Nevertheless, the drift error of the first stage results from the calculation of the transformation rather than from the joint angles themselves while the errors in the second stage originate in ground impacts measured by the joint torque sensors. For this reason the correlation of errors is expected to be small. Thus, the filter is a good method to combine both sources of information and to extract absolute pitch and roll angles using leg sensors only. The advantage of the error state formulation of the Kalman filter is that it does not require a complex motion model of the robot since it only estimates the error of the states but not the states themselves. Further, due to the feedback formulation the error is corrected at each step and the assumption of a perfect correction allows to simplify the prediction stage of the filter. In the following, the general equations for a Kalman filter as given in [14] are reduced to the error state formulation.



**Fig. 3** Signal flow diagram for the complete joint torque aided 6 DOF odometry implemented on the DLR Crawler

The first part of the filter is the prediction or time update step. Herein,  $\Delta \hat{\mathbf{x}}_t = [\Delta \beta_{\text{cor}}, \Delta \gamma_{\text{cor}}]^T$  represents the rotation angle error estimates and  $\Delta \mathbf{z}_t = [\Delta \beta, \Delta \gamma]^T$  the rotation angle error measurements, which are derived later in this section following the Kalman filter description.

$$\Delta \hat{\mathbf{x}}_t^- = \mathbf{A}_t \Delta \hat{\mathbf{x}}_{t-1} = 0 \quad (16)$$

$$\mathbf{P}_t^- = \mathbf{A}_t \mathbf{P}_{t-1} \mathbf{A}_t^T + \mathbf{Q}_p \quad (17)$$

In this first stage of the filter, equation (16) presents an error propagation model, where the terms  $\Delta \hat{\mathbf{x}}_t^-$  and  $\Delta \hat{\mathbf{x}}_{t-1}$  are the predicted error estimate at time  $t$  and the corrected error estimate at time  $t-1$  respectively. Due to the feedback formulation of the filter the error estimate is predicted to be zero. The process matrix  $\mathbf{A}_t$  is set to be a 2 by 2 identity matrix,  $\mathbf{A}_t = \mathbf{I}_{2 \times 2}$ . Equation (17) gives the predicted estimate of the error covariance matrix  $\mathbf{P}_t^-$  based on the previous estimate of the error covariance matrix  $\mathbf{P}_{t-1}$  and the process noise covariance matrix  $\mathbf{Q}_p$ .

The second part of the filter is the correction or measurement update step. Here the Kalman gain matrix  $\mathbf{K}_t$  is calculated based on the predicted estimate of the process error covariance matrix  $\mathbf{P}_t^-$ , the matrix  $\mathbf{H} = \mathbf{I}_{2 \times 2}$  relating the measured errors  $\Delta \mathbf{z}_t$  and the error state estimates  $\Delta \hat{\mathbf{x}}_t$  as well as the measurement noise covariance matrix  $\mathbf{Q}_m$ . Depending on the covariance matrices, the Kalman gain matrix adjusts the influence on the corrected error estimate either towards the predicted error estimate or towards the measurement. The last part of the correction step is the update of the estimated error covariance matrix  $\mathbf{P}_t$ .

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_t^- \mathbf{H}^T + \mathbf{Q}_m)^{-1} \quad (18)$$

$$\Delta \hat{\mathbf{x}}_t = \mathbf{K}_t \Delta \mathbf{z}_t \quad (19)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^- \quad (20)$$

For the odometry all calculations involving rotation matrices and Euler angles follow the xyz convention and all rotation matrices are calculated as shown below with yaw angle  $\alpha$ , pitch angle  $\beta$ , roll angle  $\gamma$ ,  $c$  representing a cosine and  $s$  representing a sine.

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_\alpha \cdot \mathbf{R}_{\beta\gamma} \\ &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\beta & s\beta s\gamma & s\beta c\gamma \\ 0 & c\gamma & -s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \\ &= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \end{aligned} \quad (21)$$

In order to fuse the orientation angles provided by the odometry and the pitch and roll angles based on the joint torque measurements an orientation angle error vector is calculated based on the predicted rotation matrix  ${}^w\mathbf{R}_t^-$  and the measured rotation matrix  $\mathbf{R}_{\beta\gamma\text{meas},t}$ . Here, the matrix  ${}^w\mathbf{R}_t^-$  is based on the corrected rotation matrix  ${}^w\mathbf{R}_{t-1}$  of the previous time step relating the body frame  $b$  and the world frame  $w$  and the matrix  $\mathbf{R}_{\text{odo},t}$  representing the incremental rotation in-between the last two consecutive time steps calculated by the first stage of the odometry algorithm.

$${}^w\mathbf{R}_t^- = {}^w\mathbf{R}_{t-1} \mathbf{R}_{\text{odo},t} \quad (22)$$

Since the joint torque measurements allow no yaw angle estimate,  $\mathbf{R}_{\beta\gamma\text{meas},t}$  only consists of pitch and roll terms. The predicted rotation matrix  ${}^w\mathbf{R}_t^-$  is separated into a rotation matrix  $\mathbf{R}_{\alpha,t}^-$  representing the yaw component and a rotation matrix  $\mathbf{R}_{\beta\gamma,t}^-$  representing the pitch and roll components. Only  $\mathbf{R}_{\beta\gamma,t}^-$  is used in the fusion process and is related to  $\mathbf{R}_{\beta\gamma\text{meas},t}$  by the following equation.

$$\mathbf{R}_\Delta = \mathbf{R}_{\beta\gamma,t}^- \cdot \mathbf{R}_{\beta\gamma\text{meas},t}^T \quad (23)$$

Herein,  $\mathbf{R}_\Delta = [r_{\Delta,(i,j)}]_{3 \times 3}$  is a matrix that represents the pitch and roll rotation error measurement and  $\Delta \mathbf{z}_t = [\Delta \beta, \Delta \gamma]^T$  can be calculated as follows.

$$\Delta \beta = \arcsin(-r_{\Delta,(3,1)}) \quad (24)$$

$$\Delta \gamma = \text{atan2}(r_{\Delta,(2,1)}, r_{\Delta,(1,1)}) \quad (25)$$

Applying both steps of the Kalman filter and using  $\Delta \hat{\mathbf{x}}_t = [\Delta \beta_{\text{cor}}, \Delta \gamma_{\text{cor}}]^T$  to calculate a corrected rotation error matrix  $\mathbf{R}_{\Delta,\text{cor}}$ , the corrected rotation matrix of the current time step  ${}^w\mathbf{R}_t$  can be calculated.

$${}^w\mathbf{R}_t = \mathbf{R}_{\alpha,t}^- \cdot \mathbf{R}_{\Delta,\text{cor}}^T \cdot \mathbf{R}_{\beta\gamma,t}^- \quad (26)$$

Using the above matrix, the position of the walking robot (the DLR Crawler in our case) with respect to the world frame  ${}^w\mathbf{p}_{\text{Crawler},t}$  can be updated based on

the relative position change  ${}^b\mathbf{t}_{\text{odo}}$  computed by the first stage of our algorithm.

$${}^w\mathbf{p}_{\text{Crawler},t} = {}^w\mathbf{p}_{\text{Crawler},t-1} + {}^w\mathbf{R}_t \cdot {}^b\mathbf{t}_{\text{odo}} \quad (27)$$

The measurement noise covariance matrix  $\mathbf{Q}_m$  is significantly larger than the process noise covariance matrix  $\mathbf{Q}_p$ . The final settings have to be found by manual filter tuning and are further discussed in the following section.

## 4 Experiments

In this section we present test results for the complete 6 DOF leg odometry algorithm using the six-legged DLR Crawler as experimental platform. The test runs were performed on flat lab floor as well as in our gravel testbed, a  $2\text{ m} \times 2\text{ m}$  box filled with gravel 10 to 15 cm high. First for completeness, we briefly describe the DLR Crawler. Next, we give an example of the performance of each single stage of the odometry algorithm to illustrate the necessity of data fusion. Then we discuss further sources of errors and present three parameters that are used to attenuate those effects and to adjust the odometry algorithm for different conditions. Following, we evaluate the performance for forward walking and turning on two different substrates, at two different joint stiffness settings and two different desired walking velocities. Further, we present some results for walking along rectangular paths and for walking uphill. During all of the test runs the robot was steered manually and ground truth measurements were recorded using an A.R.T. tracking system. Here, a target body was attached to the DLR Crawler that was tracked with four infrared cameras to obtain ground truth data of the 6 DOF pose with an average accuracy of 0.5 mm for the translational degrees of freedom and  $0.12^\circ$  for the orientation angles.

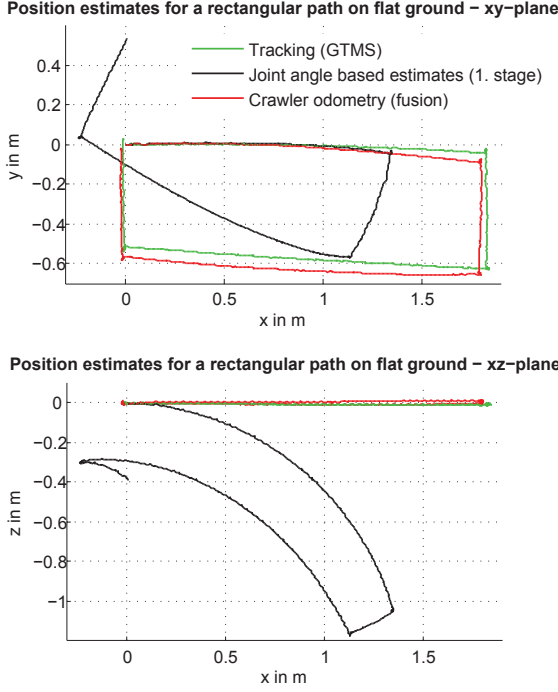
### 4.1 The DLR Crawler

As mentioned above and shown in Fig. 1, our experimental platform is the DLR Crawler [15], a six-legged, actively compliant walking robot that is based on the fingers of DLR Hand II [16]. It was built to serve as a laboratory testbed for the development of control, gait and navigation algorithms. In a common configuration, the Crawler stands 90 mm high and its feet span an area of  $350\text{ mm} \times 380\text{ mm}$ . Each of the legs has four joints, of which the distal two are coupled resulting in three active degrees of freedom. Further, each leg hosts a large set of proprioceptive sensors that enable active compliance control as well as the odometry calculations. These

sensors are motor side Hall sensors for commutation and relative joint position measurement, link side potentiometers for absolute joint position reference, link side joint torque sensors and a 6 DOF force-torque sensor within the foot. The implemented active joint compliance control emulates a virtual spring-damper system within each joint, which is enforced by an underlying torque control loop. This allows active stiffening or softening of the legs by control and thus to increase the robustness with respect to the terrain. In addition to the proprioceptive sensors within the legs, the robot is equipped with a stereo camera head for visual odometry, obstacle avoidance and terrain assessment, and an inertial measurement unit. Since the robot is designed as a testbed it uses an external power supply and external computing to avoid resource restrictions. The main gait algorithm [17] of the DLR Crawler is biologically inspired and employs results from research on stick insects [18]. Due to leg coordination mechanisms acting in-between neighboring legs, the gait emerges according to the robot state, the desired walking speed and the desired direction instead of following a fixed pattern. This gait coordination is combined with reflexes that each leg is able to activate. One of the reflexes is an elevator reflex that is triggered by collisions during the swing phase of a leg. It results in retracting and raising the leg in order to overcome the encountered obstacle. Another reflex is the stretch reflex that enforces ground contact by extending the leg if it does not find ground contact at the end of the swing phase or loses ground contact during the stance phase. The flexible gait coordination in combination with those reflexes allows the robot to overcome obstacles within the walking height reactively without planning. Nevertheless, due to its flexibility and its temporal variations the gait algorithm requires the leg odometry to account for permanently changing contact situations with at least 3 and up to 6 legs in ground contact.

### 4.2 Individual Behavior of the First and Second Stage of the Odometry Algorithm; Drift and Error Sources

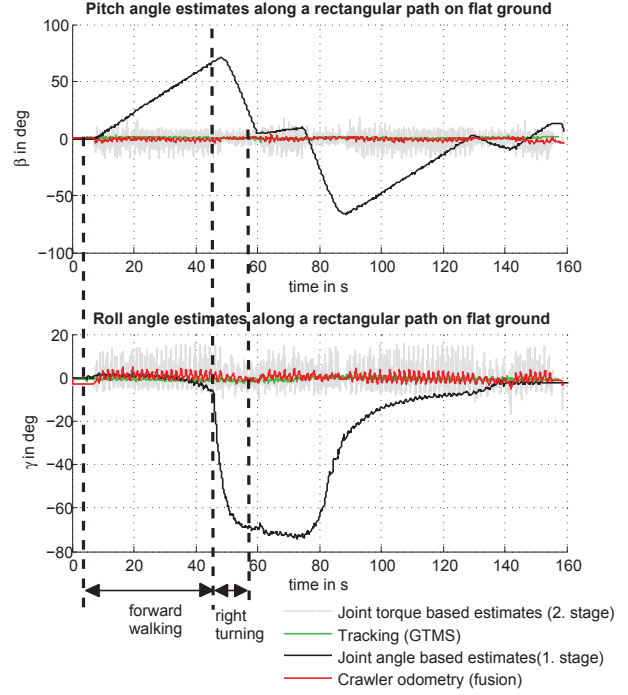
To give an example of the performance of the first stage of the leg odometry algorithm, Fig. 4 shows the path of the robot estimated by this stage in comparison to the data obtained by the ground truth measurement system (GTMS). As can be seen the path of the robot estimated by this first, purely kinematics based stage is strongly bent. This behavior is mainly attributed to the interaction of the odometry calculation and the active compliance resulting in an angular drift that is shown in the “forward walking” marked region of Fig. 5. This drift of the pitch or roll angle depends on the direction



**Fig. 4** Walking a rectangular path on flat ground - comparison of position estimates provided by the first stage of the odometry algorithm, the data fusion and the ground truth measurement system; top: projection on the xy-plane; bottom: projection on the xz-plane

of motion and strongly affects the absolute pose estimate. For forward or backward walking the pitch angle is affected while sideways walking leads to a drift of the roll angle. For pure turning this disturbing behavior was not observed. To explain the behavior, the case of forward walking is considered. Here, after touch down a front leg moves towards the center of gravity (COG) of the robot. During this motion the loading of the leg increases and causes its height to decrease due to the active compliance. Opposite to this behavior the loading of a hind leg decreases over the course of its stance motion since it moves away from the robot COG. Thus, the leg extends depending on the stiffness setting. To each calculation of the incremental pose change this behavior appears to be a tilting motion that increases the pitch angle, summing up to the strong angular drift apparent in Fig. 5. Having built up a large pitch angle each turning motion, i.e. increase or decrease of the yaw angle, transfers the pitch to a roll angle which can also be seen in the plots.

Another error is caused by the initial contact phase of the legs, especially during the execution of the stretch reflex while walking on uneven ground. The source of this error is that the algorithm considers larger parts

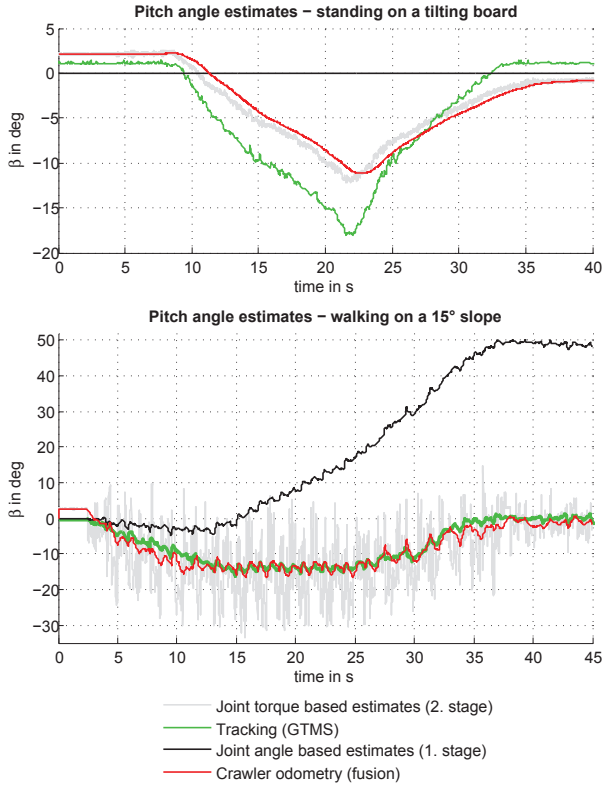


**Fig. 5** Walking a rectangular path on flat ground - comparison of pitch and roll angle estimates provided by the first and second stage of the odometry algorithm, the data fusion and the ground truth measurement system; top: pitch angle estimates; bottom: roll angle estimates

of the downward motion at the beginning of the stance phase than of the upward motion at the end of the stance phase, where the leg quickly loses contact. The downward motion of a leg caused by the stretch reflex has an effect on the translation estimation and appears to the algorithm as upward motion of the robot body. In order to attenuate this behavior, the ground contact has to be detected properly. For this purpose contact thresholds are introduced that can be adjusted and influence the pose estimate depending on walking speed and terrain, which is discussed in further detail in the following subsection on parameter tuning.

The error caused by the stretch reflexes appears randomly depending on the distribution of height differences of the ground. The error due to the compliance is somehow systematic, but depends on many parameters like joint stiffness, walking velocity and ground properties. For this reason an error prediction and correction without further sources of information is infeasible. Thus, an absolute source for the pitch and roll angles is needed to correct the odometry data. As introduced in section 3.2, the joint torque measurements provide enough data to estimate the gravity vector with respect to the body frame.





**Fig. 6** Comparison of pitch angle estimates; top: standing on a tilting platform; bottom: walking along a 15° slope

To give an example for the performance of the second stage of the odometry algorithm, Fig. 6 shows joint torque measurement based pitch angle estimates for the DLR Crawler. The upper figure refers to the Crawler standing on a slowly tilting board. It shows that the pitch angle estimate has an offset and does not closely track the pitch angle. The lower figure shows the pitch angle of the Crawler estimated for walking uphill on a slope. It clearly shows that the ground impacts during walking cause force peaks that are translated to large false peaks within the pitch angle estimates. It also shows that for walking uphill the pitch angle estimate of the first stage of the algorithm as presented in section 3.1 is dominated by the compliance induced drift. Nevertheless, standing on the tilting board or walking uphill, the data contains information about the inclination of the robot. Thus, the joint torque sensors emulate an inclination sensor even though it is not a very accurate one. Since the joint torque based pitch and roll angle estimates are very noisy but free of drift and the joint angle based estimates include little noise but strong drift components, they are very well suited to be fused by a Kalman filter as presented in the previous section. The improvements gained by this fusion pro-

cess with respect to each single stage become apparent in Fig. 4, Fig. 5 and Fig. 6. As can be seen, the large position errors caused by the compliance induced pitch angle drift are completely removed by fusion with the joint torque based absolute pitch and roll angles. The remaining position error mainly originates in a small yaw angle drift during forward walking and slight underestimation of the yaw angle during turning. Due to a missing absolute reference for the yaw angle this error cannot be removed.

#### 4.3 Tuning Parameters

The algorithm provides three parameters that are adjusted in order to increase the accuracy and to account for different conditions. These parameters are two torque thresholds used for contact detection in the first stage of the algorithm and the process noise covariance matrix of the error state Kalman filter in the third stage.

The first torque threshold is active during the swing phase of the leg and detects the initial contact that marks the onset of the stance phase. The second threshold becomes active once the stance is established. It is used to monitor if the leg loses contact during stance and to detect the onset of the next swing phase once the leg lifts. In all cases the first threshold is higher than the second and helps to discard the error-prone initial contact and loading phase of a leg that has strong influence on errors of the z-coordinate and especially the yaw angle. Thus, the first threshold is used to reduce the error of the yaw estimate while the second is mainly used to reduce the error of the estimate of the z-coordinate. During the experiments we found two distinct values for the first threshold that are independent of terrain or stiffness setting and are only influenced by the walking speed. Here, the first constant value was used for slow walking dominated by emerging pentapod or tetrapod gaits. The threshold had to be increased to the second constant value for fast walking with dominating tripod gaits. This is actually obvious since the leg load is higher during the tripod. We found that the second torque threshold is independent of the stiffness but needs to be adjusted for each terrain and once the gait changes from tetrapod to tripod.

The last tuning parameter, the process noise covariance matrix, is used to remove the remaining z-coordinate drift. The measurement noise covariance matrix is kept equal for all settings and the process noise covariance matrix is assumed to change depending on the gait, the stiffness setting and the terrain. We found that for walking on gravel the values were smaller than

for walking on the lab floor. Considering the fixed measurement noise covariance matrix, this means that for walking on gravel the joint torque based pitch and roll estimates are less trusted. Further, the process noise covariance values are smaller for slow walking on gravel than for fast walking on gravel and did not change when the stiffness changed. For walking on lab floor the values had to be increased once the gait changed from tetrapod to tripod and also had to be increased once the stiffness was increased. Altogether, there is no single set of parameters that is valid for all combinations of gait, terrain and joint stiffness. Nevertheless, different parameter sets can be identified and stored depending on the combination of the dominating gait, the stiffness setting and the terrain. This identification is currently done manually but in future will be done automatically by calibrating the leg odometry on a new terrain type with respect to the IMU and visual odometry derived pose estimates. Each time a change of dominating gait or stiffness is initiated or a change of terrain is detected by visual cues or a change of reflex activation behavior, the filter can be automatically adapted by loading the appropriate parameter set.

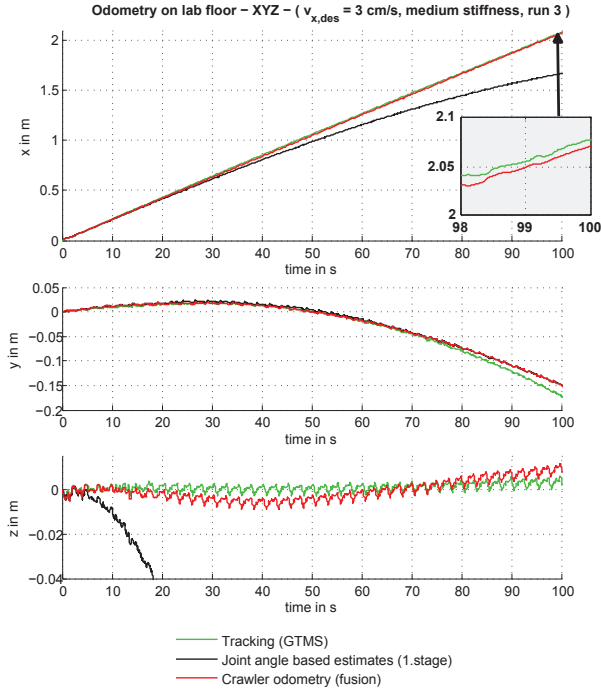
#### 4.4 Forward Walking

This set of experiments was conducted to evaluate the performance of the complete odometry and its associated errors under varying conditions. For this purpose we commanded the robot to walk forward at a certain velocity and measured the absolute translation and orientation errors when the ground truth measurement system (GTMS) indicated 0.5 m, 1 m and 1.5 m travel in the x-direction of the local start frame that has been defined in section 3.1. We commanded the robot to walk forward at two different desired velocities, at which different gaits emerge, 3 cm/s to enforce mainly a tetrapod gait and 6 cm/s to generate mainly a tripod gait. For each velocity setting runs at two different stiffness values were recorded - a medium joint stiffness setting and a stiff configuration with doubled joint stiffness values. Further, each velocity-stiffness configuration was tested on laboratory floor as well as in the gravel testbed resulting in 8 different test conditions. For each of these 8 conditions 10 separate runs were recorded and analyzed.

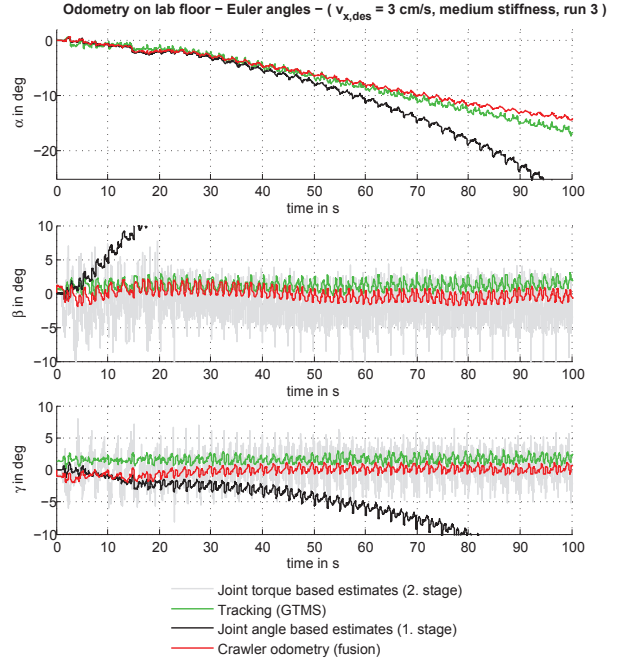
Fig. 7 displays the pose estimates for good exemplary runs at 3 cm/s on lab floor and at 6 cm/s on gravel. As can be seen in the graphs the actual walking velocity is in both cases lower than the commanded which is attributed to the medium joint stiffness setting and the resulting reduced joint trajectory tracking accuracy. After walking a comparable distance of 1.5 m

in the x-direction of the local start frame measured by the GTMS, the absolute Cartesian position errors are 1.74 cm for the run at 3 cm/s on lab floor and 3.21 cm for the run at 6 cm/s on gravel. The absolute Cartesian position errors at this distance with respect to the Cartesian path length are 0.82 % and 1.39 % respectively. In both cases the dominant source of the Cartesian position errors is a drift of the yaw angle estimate. Additionally, the baselines of the z-coordinate estimates show deviations from the baselines of the GTMS data. Nevertheless, the oscillations around the baselines closely represent the observed z-coordinate variations that result from the change of stance configurations. The baseline deviations are caused by a combination of small remaining influences from the stretch reflex and the compliance induced z-coordinate drift as well as inaccuracies in the pitch angle estimate. It can be seen that the pitch and roll angle estimates show offsets, which appear due to horizontal propulsion forces, that have been assumed to be negligible at low speeds.

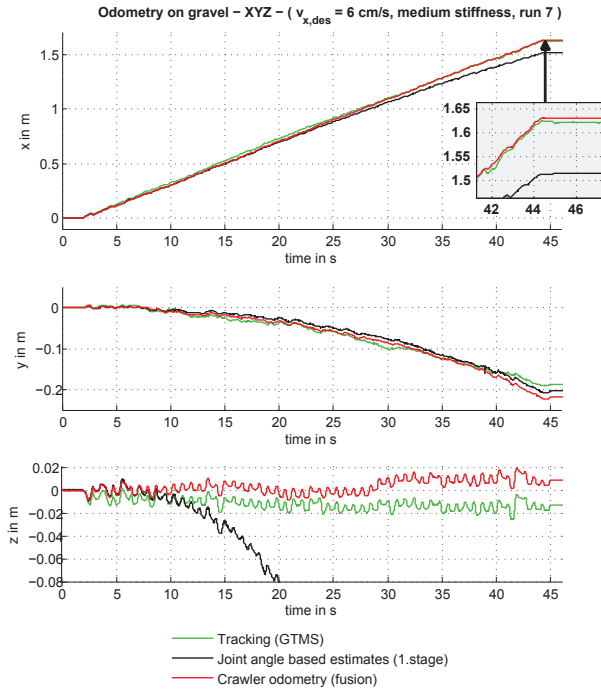
Table 1 displays the means and standard deviations of the errors observed during the forward walking experiments. The computed errors for each single run are the Cartesian position errors in the x, y and z-direction with respect to the start frame, the absolute Cartesian position error with respect to the traveled path length and the root mean square (rms) errors of the yaw, pitch and roll angles all measured after 0.5 m, 1 m and 1.5 m travel in the x-direction of the start frame. As can be seen, the odometry algorithm underestimates the distance traveled in the x-direction for all trials on lab floor. On the contrary, it overestimates the traveled distance on gravel due to increased leg slip that is partially mistaken for forward body motion. In almost all cases the lateral motion in the y-direction is overestimated which is strongest on gravel and mainly caused by larger yaw angle errors. In all cases the yaw angle rms error increases with distance indicating a yaw angle drift. Pitch and roll angle rms errors show in most cases constant values independent of the traveled distance. Only on lab floor the pitch angle shows a slight drift but with errors being smaller than the ones for gravel. The z-coordinate estimates show the largest errors for slow walks on gravel. Considering the absolute Cartesian position error with respect to the path length,  $\Delta p$ , the smallest error on gravel is obtained with a tripod gait and medium joint stiffness settings, while on lab floor a tetrapod gait and medium joint stiffness is best. Nevertheless, with the right torque thresholds and process covariance matrices all configurations show comparable results. As expected the estimation errors are smaller on lab floor where  $\Delta p$  is within 1 % to 3 % while on gravel it mainly lies in a range of 2 % to 6 %.



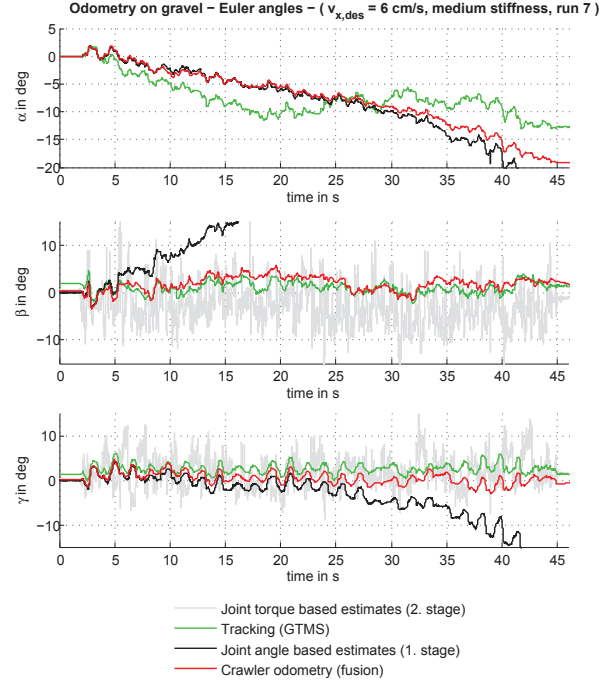
(a)



(b)



(c)



(d)

**Fig. 7** Exemplary Crawler odometry pose estimates for forward walking on different substrates and at different velocities: a) x,y,z and b) Euler angles for  $v_{des} = 3$  cm/s, medium joint stiffness, lab floor; c) x,y,z and d) Euler angles for  $v_{des} = 6$  cm/s, medium joint stiffness, gravel

**Table 1** Error behavior of the complete Crawler odometry (with fusion of first and second stage) for **forward walking**: Average Cartesian position errors ( $\Delta x, \Delta y, \Delta z$ ) and average orientation rms errors ( $\Delta \alpha_{rms}, \Delta \beta_{rms}, \Delta \gamma_{rms}$ ) at different combinations of ground substrate, desired forward velocity and joint stiffness at 50 cm, 100 cm and 150 cm distance to the start point with respect to the x-direction of the start frame (10 experimental runs for each combination);  **$\Delta p$  is the absolute Cartesian position error with respect to the traveled Cartesian path length**

$x_{tracking} = 50 \text{ cm}$								
$v_{des}$	Stiffness	$\Delta x$ in cm mean (std)	$\Delta y$ in cm mean (std)	$\Delta z$ in cm mean (std)	<b><math>\Delta p</math> in % mean (std)</b>	$\Delta \alpha_{rms}$ in $^\circ$ mean (std)	$\Delta \beta_{rms}$ in $^\circ$ mean (std)	$\Delta \gamma_{rms}$ in $^\circ$ mean (std)
Lab floor								
3 cm/s	medium	-1.15 (0.38)	0.06 (0.30)	-0.18 (0.49)	<b>1.91 (0.44)</b>	0.55 (0.21)	0.33 (0.83)	2.28 (0.24)
3 cm/s	stiff	-1.49 (0.20)	0.61 (0.33)	-0.13 (0.34)	<b>2.48 (0.29)</b>	0.46 (0.26)	1.40 (0.30)	0.99 (0.40)
6 cm/s	medium	-0.78 (0.45)	0.03 (0.52)	-0.14 (0.15)	<b>1.6 (0.33)</b>	0.34 (0.18)	1.27 (0.1)	2.37 (0.17)
6 cm/s	stiff	-0.48 (0.21)	0.14 (0.15)	-0.29 (0.26)	<b>1.03 (0.31)</b>	0.31 (0.04)	1.30 (0.19)	2.46 (0.15)
Gravel								
3 cm/s	medium	0.80 (2.28)	0.15 (2.48)	2.47 (0.89)	<b>4.82 (2.28)</b>	1.79 (1.39)	1.79 (0.67)	1.26 (0.39)
3 cm/s	stiff	0.83 (1.70)	0.99 (1.65)	3.09 (1.82)	<b>4.70 (2.18)</b>	2.44 (1.02)	2.54 (0.80)	1.76 (0.81)
6 cm/s	medium	0.87 (1.37)	1.84 (1.96)	0.04 (1.04)	<b>3.86 (1.87)</b>	2.12 (1.23)	2.45 (0.58)	2.02 (0.85)
6 cm/s	stiff	0.07 (1.62)	1.00 (1.83)	0.29 (1.50)	<b>3.59 (1.53)</b>	2.01 (1.14)	2.16 (0.51)	2.60 (1.63)
$x_{tracking} = 100 \text{ cm}$								
$v_{des}$	Stiffness	$\Delta x$ in cm mean (std)	$\Delta y$ in cm mean (std)	$\Delta z$ in cm mean (std)	<b><math>\Delta p</math> in % mean (std)</b>	$\Delta \alpha_{rms}$ in $^\circ$ mean (std)	$\Delta \beta_{rms}$ in $^\circ$ mean (std)	$\Delta \gamma_{rms}$ in $^\circ$ mean (std)
Lab floor								
3 cm/s	medium	-1.64 (0.59)	0.01 (0.55)	-0.36 (1.08)	<b>1.47 (0.48)</b>	0.59 (0.22)	1.16 (0.31)	2.05 (0.18)
3 cm/s	stiff	-2.13 (0.49)	1.12 (1.15)	0.07 (0.74)	<b>2.00 (0.41)</b>	0.76 (0.23)	1.23 (0.29)	0.83 (0.31)
6 cm/s	medium	-1.77 (0.59)	-0.08 (0.92)	-0.08 (0.46)	<b>1.63 (0.23)</b>	0.38 (0.24)	1.38 (0.2)	2.32 (0.08)
6 cm/s	stiff	-1.34 (0.40)	0.69 (0.40)	-0.46 (0.39)	<b>1.31 (0.33)</b>	0.39 (0.11)	1.50 (0.15)	2.16 (0.10)
Gravel								
3 cm/s	medium	2.33 (2.90)	1.97 (5.17)	2.15 (1.79)	<b>3.79 (2.29)</b>	2.47 (1.27)	2.31 (0.44)	1.68 (0.32)
3 cm/s	stiff	0.44 (2.22)	0.47 (2.89)	4.16 (3.10)	<b>3.37 (1.71)</b>	2.55 (1.42)	2.38 (0.28)	1.82 (0.54)
6 cm/s	medium	0.71 (1.98)	2.26 (2.40)	-0.25 (1.77)	<b>2.47 (1.25)</b>	2.29 (1.14)	2.63 (0.84)	2.27 (0.92)
6 cm/s	stiff	-0.58 (2.24)	3.39 (4.04)	-0.17 (1.52)	<b>3.37 (2.00)</b>	2.82 (1.24)	2.33 (0.50)	2.80 (1.04)
$x_{tracking} = 150 \text{ cm}$								
$v_{des}$	Stiffness	$\Delta x$ in cm mean (std)	$\Delta y$ in cm mean (std)	$\Delta z$ in cm mean (std)	<b><math>\Delta p</math> in % mean (std)</b>	$\Delta \alpha_{rms}$ in $^\circ$ mean (std)	$\Delta \beta_{rms}$ in $^\circ$ mean (std)	$\Delta \gamma_{rms}$ in $^\circ$ mean (std)
Lab floor								
3 cm/s	medium	-1.74 (0.85)	0.86 (1.22)	-0.31 (1.73)	<b>1.30 (0.52)</b>	0.83 (0.36)	1.95 (0.58)	1.93 (0.20)
3 cm/s	stiff	-2.77 (0.89)	2.97 (2.18)	0.35 (1.24)	<b>2.24 (0.46)</b>	1.34 (0.54)	1.27 (0.32)	0.76 (0.23)
6 cm/s	medium	-2.69 (0.64)	-0.3 (1.47)	0.32 (0.68)	<b>1.63 (0.24)</b>	0.5 (0.2)	1.63 (0.17)	2.31 (0.06)
6 cm/s	stiff	-2.17 (0.45)	1.13 (0.86)	-0.55 (0.41)	<b>1.37 (0.27)</b>	0.43 (0.17)	1.71 (0.09)	1.97 (0.07)
Gravel								
3 cm/s	medium	3.88 (3.55)	1.08 (7.98)	2.57 (2.17)	<b>3.71 (1.79)</b>	3.13 (1.32)	2.42 (0.51)	1.88 (0.39)
3 cm/s	stiff	1.32 (3.43)	1.83 (5.57)	4.44 (4.46)	<b>3.26 (1.60)</b>	2.95 (1.52)	2.36 (0.43)	1.98 (0.41)
6 cm/s	medium	2.81 (2.4)	2.53 (5.05)	-0.57 (2.51)	<b>2.54 (1.69)</b>	2.51 (1.48)	2.59 (0.70)	2.30 (0.79)
6 cm/s	stiff	0.56 (2.79)	5.93 (6.20)	-0.57 (2.15)	<b>3.58 (1.90)</b>	3.22 (1.23)	2.21 (0.43)	2.94 (0.73)

#### 4.5 Turning

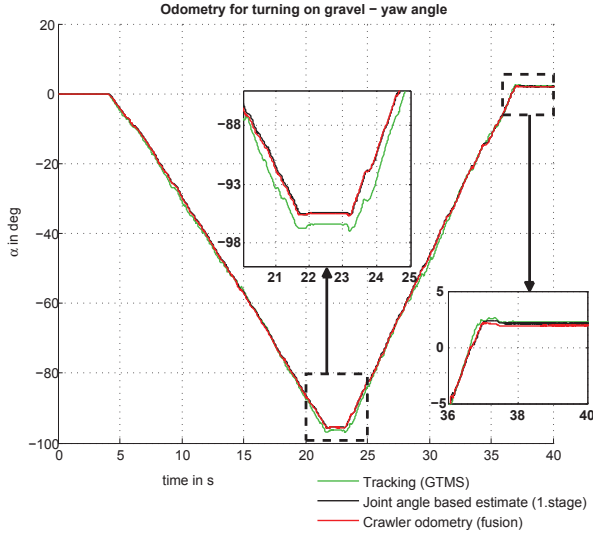
As for forward walking, a set of experiments was conducted to evaluate the performance of pure turning. To visualize an average result, Fig. 8 shows an exemplary yaw angle plot for a  $90^\circ$  right-left turn on gravel. The tuning parameters were identified for gravel as well as for lab floor and showed only very small dependence on joint stiffness setting and turning velocity. Thus, they

are only adjusted for a change of substrate. Since the algorithm appears to be independent of velocity and stiffness setting for turning, Table 2 only displays the results for five trials of turning to the right on each substrate at  $10^\circ/\text{s}$  and medium joint stiffness. The data shows that the yaw angle estimate on lab floor experiences a drift which amounts to  $2^\circ$  to  $5^\circ$  per  $90^\circ$  turn. For gravel the error is smaller and shows no drift. The Cartesian position errors are very small for turning and



**Table 2** Error behavior of the complete Crawler odometry (with fusion of first and second stage) for **turning to the right**: Average translation rms errors ( $\Delta x_{rms}, \Delta y_{rms}, \Delta z_{rms}$ ), **average absolute yaw angle errors** ( $\Delta\alpha$ ) and average orientation rms errors ( $\Delta\alpha_{rms}, \Delta\beta_{rms}, \Delta\gamma_{rms}$ ) on lab floor and gravel at  $-30^\circ$ ,  $-60^\circ$  and  $-90^\circ$  (5 experimental runs for each ground substrate);

$\alpha_{tracking}$ in $^\circ$	$\Delta x_{rms}$ in cm mean (std)	$\Delta y_{rms}$ in cm mean (std)	$\Delta z_{rms}$ in cm mean (std)	$\Delta\alpha$ in $^\circ$ mean (std)	$\Delta\alpha_{rms}$ in $^\circ$ mean (std)	$\Delta\beta_{rms}$ in $^\circ$ mean (std)	$\Delta\gamma_{rms}$ in $^\circ$ mean (std)
Lab floor							
-30	0.22 (0.03)	0.10 (0.03)	0.14 (0.07)	<b>1.29 (0.37)</b>	0.92 (0.35)	1.05 (0.21)	1.47 (0.53)
-60	0.34 (0.04)	0.16 (0.04)	0.23 (0.10)	<b>2.48 (0.46)</b>	1.53 (0.31)	0.93 (0.11)	1.27 (0.37)
-90	0.37 (0.03)	0.31 (0.08)	0.31 (0.10)	<b>3.75 (0.51)</b>	2.22 (0.33)	0.97 (0.08)	1.22 (0.32)
Gravel							
-30	0.37 (0.25)	0.44 (0.27)	0.34 (0.26)	<b>0.73 (1.91)</b>	1.15 (0.86)	1.50 (0.36)	1.92 (0.43)
-60	0.48 (0.25)	0.65 (0.26)	0.52 (0.27)	<b>1.00 (2.15)</b>	1.33 (0.80)	1.31 (0.21)	2.07 (1.02)
-90	0.65 (0.36)	0.90 (0.35)	0.72 (0.39)	<b>0.90 (1.88)</b>	1.47 (0.92)	1.29 (0.18)	1.94 (0.90)



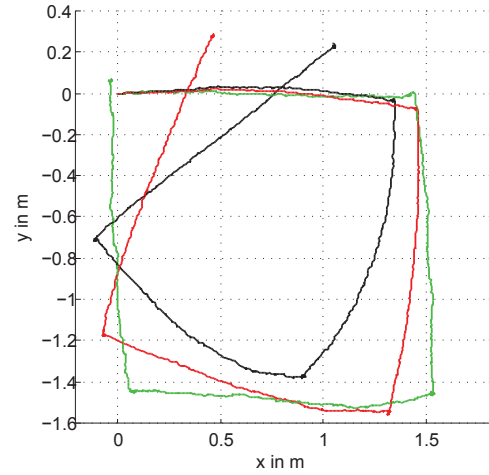
**Fig. 8** Yaw angle estimates for turning on gravel: yaw rate =  $10^\circ/\text{s}$ , medium joint stiffness

the pitch and roll angle estimates show constant but smaller errors than for forward walking. Turning to the left shows similar error behavior and the performance evaluation is omitted at this place.

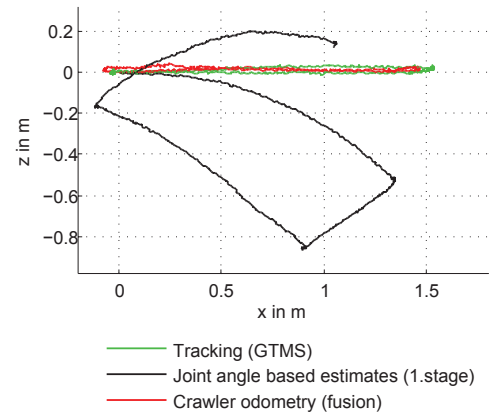
#### 4.6 Combined Motions

Next, we will give a few examples of combined motions. Fig. 9 shows an average result for walking a rectangular path on gravel. As can be seen, the absolute Cartesian position error on gravel is larger than on lab floor as shown in Fig. 4 above. In both cases the absolute Cartesian position error is mainly caused by errors of the yaw angle estimate. On lab floor the estimated path opens slightly outwards, which is a result of underestimating the yaw angle during the turning motions while it is estimated quite accurately for the forward walking parts.

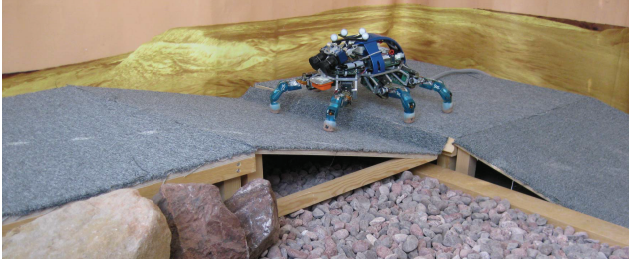
**Odometry on gravel - xy-plane ( $v_{x,des} = 6 \text{ cm/s}$ , medium stiffness)**



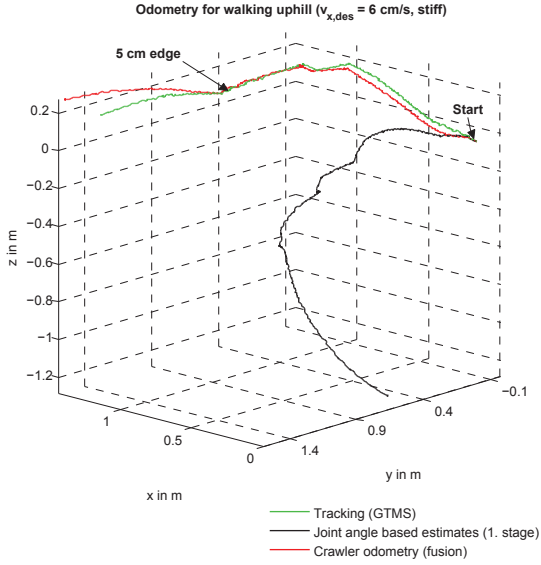
**Odometry on gravel - xz-plane ( $v_{x,des} = 6 \text{ cm/s}$ , medium stiffness)**



**Fig. 9** Walking a rectangular path on gravel - comparison of position estimates provided by the first stage of the odometry algorithm, the data fusion and the ground truth measurement system;  $v_{des} = 6 \text{ cm/s}$ , medium joint stiffness; top: projection on the xy-plane; bottom: projection on the xz-plane



(a)

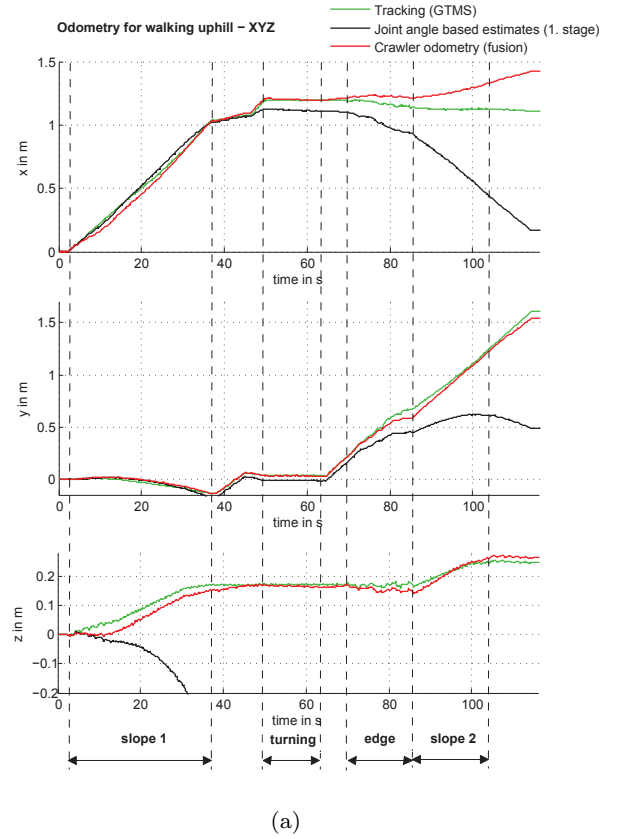


(b)

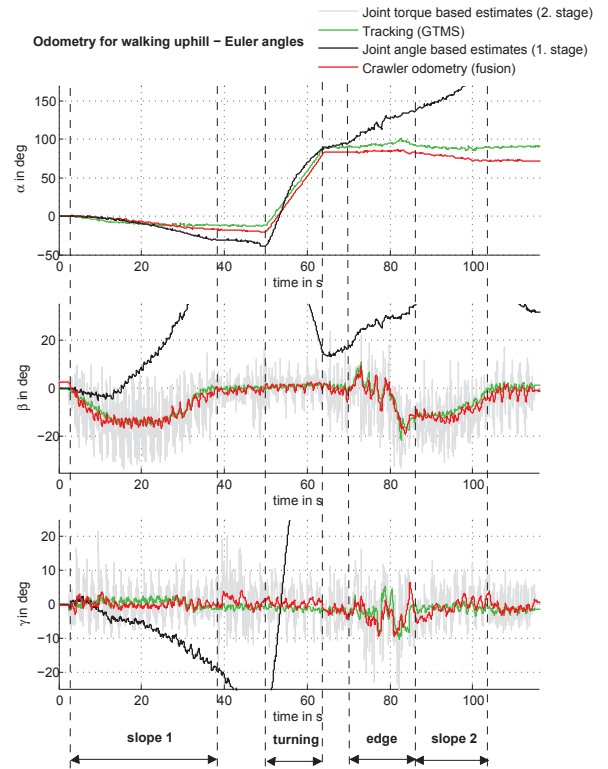
**Fig. 10** Crawler odometry position estimates for walking uphill ( $v_{des} = 6$  cm/s, high joint stiffness) - a) DLR Crawler in the test area; b) estimated and measured trajectories in xyz

The opposite happens on gravel where the rectangular path is bent inwards. Here the yaw estimates for the turning motions are quite accurate while the yaw angle estimate experiences a drift for the forward walking segments. In both cases the behavior is consistent with the performance results obtained for forward walking and turning. Even with the high number of steps the robot takes walking a distance of several meters, the z-coordinate estimate remains close to the GTMS value.

Finally, we show an average result for a path that combines forward, upward and turning motions and includes a step as an additional challenge. In this case the robot walks up a short  $15^\circ$  slope, moves sideways and turns  $90^\circ$  on a plateau, crosses a 5 cm edge and walks up a second  $15^\circ$  slope onto a second plateau as depicted in Fig 10. Along its path the odometry algorithm uses the torque thresholds and the process noise covariance matrix identified for walking with a tripod gait on lab floor and switches to the parameter set for



(a)



(b)

**Fig. 11** Crawler odometry pose estimates for walking uphill ( $v_{des} = 6$  cm/s, high joint stiffness): a) x,y,z and b) Euler angles

walking with a tripod gait on gravel while crossing the edge. This switch is initiated automatically by stronger reflex activations and height differences within the legs and improves the estimate during crossing the edge. Again, the main source of the Cartesian position error is the yaw angle estimate that experiences a drift, especially while walking along the slopes. The overall z-coordinate estimate, Fig. 11, is quite good even though it misses the onset of the upward motion and slightly overestimates it in the following due to slip along the slope. The pitch angle estimate detects the slopes and closely represents the shaky motion during crossing the edge. Apart from the edge, the pitch and roll angle estimates overestimate the shakiness of the motion along the slopes and the plateaus. Nevertheless, in our opinion the odometry results are very encouraging considering that only leg proprioception is employed.

## 5 Conclusions and Future Work

In this paper we have presented a 6 DOF leg odometry algorithm for statically stable walking robots. The strength of the algorithm is that it solely uses proprioceptive sensors embedded within the legs and provides absolute pitch and roll angle estimates based on joint torque measurements. The algorithm is applicable to stiff and compliant robots and does not rely on specific gait patterns. Being generally applicable to four-, six- or eight-legged robots, we implemented and successfully tested the leg odometry on the six-legged DLR Crawler as experimental platform. We performed numerous test runs and presented performance measures for walking at different joint stiffness settings, on different ground substrates as well as with different emerging gaits invoked by the walking velocity command. Further, we identified error sources and where possible removed or attenuated their influence on the pose estimate using heuristics. Herein, especially the torque thresholds for ground contact detection show a strong influence on the z-coordinate drift and yaw angle accuracy. Nevertheless, a remaining yaw angle drift is the main source of endpoint errors. The algorithm further allows the identification of single slipping legs and the reduction of their influence on the calculation. On the contrary, situations where the complete robot slips cannot be identified using leg sensors only. Additionally, we presented an exemplary result for the successful use of the leg odometry for motion estimation in full 3D scenarios. Being subject to spatial and temporal bounds, our algorithm is a very useful source of 6 DOF pose estimates within a local environment. As part of a visual navigation framework it is of great value since it provides

redundancy, complements the visual odometry and enhances the overall robustness of pose estimation.

In future work, we want to test our complete local stereo vision based navigation framework including the leg odometry in an outdoor environment with strongly varying substrates, lighting situations and a crater with various slopes ranging from  $10^\circ$  to  $35^\circ$ . Further, the performance of the leg odometry will be evaluated for the case of replacing the joint torque sensor based absolute pitch and roll angle estimates in the second stage of the algorithm by foot force sensor based ones.

## 6 ACKNOWLEDGMENTS

We are very thankful to Heiko Hirschmüller for providing us with valuable computer vision literature and fruitful discussions. Further, we want to thank the reviewers for their detailed and constructive comments.

## References

1. A. Chilian, H. Hirschmüller and M. Görner, Multisensor Data Fusion for Robust Pose Estimation of a Six-Legged Walking Robot, *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA*, 2011, pp 2497-2504
2. A. Stelzer, H. Hirschmüller and M. Görner, Stereo Vision Based Navigation of a Six-Legged Walking Robot in Unknown Rough Terrain, *The International Journal of Robotics Research - Special Issue on Robot Vision*, 2012, Vol. 31, No. 4, pp 381-402
3. P. Lamon and R. Siegwart, Inertial and 3D-odometry fusion in rough terrain - Towards real 3D navigation, *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan*, 2004, pp 1716-1721
4. P.-C. Lin, H. Komsuoglu and D.E. Koditschek, A Leg Configuration Measurement System for Full-Body Pose Estimates in a Hexapod Robot, *IEEE Transactions on Robotics*, 2005, Vol.21, No.3, pp 411-422
5. P.-C. Lin, H. Komsuoglu and D.E. Koditschek, Sensor Data Fusion for Body State Estimation in a Hexapod Robot With Dynamical Gaits, *IEEE Transactions on Robotics*, 2006, Vol.22, No.5, pp 932-943
6. G.P. Roston and E.P. Krotkov, Dead Reckoning Navigation for Walking Robots, *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, USA*, 1992, pp 607-612
7. B. Gassmann, J.M. Zöllner and R. Dillmann, Navigation of Walking Robots: Localisation by Odometry, *Proceedings of the 7th International Conference on Climbing and Walking Robots, Madrid, Spain*, 2004, pp 953-960
8. B. Gassmann, F. Zacharias, J.M. Zöllner and R. Dillmann, Localization of Walking Robots, *Proceedings of the 2005 International Conference on Robotics and Automation, Barcelona, Spain*, 2005, pp 1483-1488
9. M. Reinstein and M. Hoffmann, Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer, *Proceedings of the 2011 IEEE Conference on Robotics and Automation, Shanghai, China*, 2011, pp 617-624

10. J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano and D. Wooden, Robust Multi-Sensor, Day/Night 6-DOF Pose Estimation for a Dynamic Legged Vehicle in GPS-denied Environments, *Proceedings of the 2012 IEEE Conference on Robotics and Automation, Saint Paul, USA*, 2012, pp 619-626
11. Ch. Ott, M.A. Roa and G.Hirzinger, Posture and Balance Control for Biped Robots based on Contact Force Optimization, *Proceedings of the 2011 IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia*, 2011, pp 26-33
12. T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi and D.G. Caldwell, Dynamic Torque Control of a Hydraulic Quadruped Robot, *Proceedings of the 2012 IEEE Conference on Robotics and Automation, Saint Paul, USA*, 2012
13. R.M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G.Vaidya and M.B. Kim, Pose Estimation from Corresponding Point Data, *IEEE Transactions on Systems, Man and Cybernetics*, 1989, Vol.19, No.6, pp 1426-1446
14. G. Welch and G.Bishop, An Introduction to the Kalman Filter, *Technical Report - TR 95-041 - Department of Computer Science, University of North Carolina, Chapel Hill, USA*, 2006
15. M.Görner, Th. Wimböck, A. Baumann, M. Fuchs, T. Bahls, M. Grebenstein, Ch. Borst, J. Butterfass and G. Hirzinger, The DLR-Crawler, A Testbed for Actively Compliant Walking Based on the Fingers of DLR-Hand II, *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*, 2008, pp 1525-1531
16. J. Butterfass, M. Grebenstein, H. Liu and G. Hirzinger, DLR-Hand II: Next Generation of a Dextrous Robot Hand, *Proceedings of the 2001 IEEE Conference on Robotics and Automation, Seoul, Korea*, 2001, pp 109-114
17. M. Görner, Th. 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*, 2009, Vol.36, No. 4, pp 344-351
18. H. Cruse, What Mechanisms Coordinate Leg Movement in Walking Arthropods, *Trends in Neuroscience*, 1990, Vol. 13, pp 15-21