Humanoid odometric localization integrating kinematic, inertial and visual information

Giuseppe Oriolo[1] · Antonio Paolillo[1] · Lorenzo Rosa[1] · Marilena Vendittelli[1]

Received: 1 October 2014 / Accepted: 9 September 2015

Abstract We present a method for odometric localization of humanoid robots using standard sensing equipment, i.e., a monocular camera, an inertial measurement unit (IMU), joint encoders and foot pressure sensors. Data from all these sources are integrated using the prediction-correction paradigm of the Extended Kalman Filter. Position and orientation of the torso, defined as the representative body of the robot, are predicted through kinematic computations based on joint encoder readings; an asynchronous mechanism triggered by the pressure sensors is used to update the placement of the support foot. The correction step of the filter uses as measurements the torso orientation, provided by the IMU, and the head pose, reconstructed by a VSLAM algorithm. The proposed method is validated on the humanoid NAO through two sets of experiments: open-loop motions aimed at assessing the accuracy of localization with respect to a ground truth, and closed-loop motions where the humanoid pose estimates are used in real-time as feedback signals for trajectory control.

Keywords Humanoid robots · Localization · Odometry · Visual SLAM · EKF

1 Introduction

In mobile robotics, maintaining an accurate estimate of the robot placement in the world is a basic prerequisite for autonomy. For a humanoid robot, this amounts to reconstructing the *pose* (position and orientation) of one of its bodies (e.g., the torso) with respect to a fixed reference frame. Once this information is available, it is possible to localize any other point on the robot using kinematic computations based on joint encoder readings.

Methods for humanoid localization can be roughly classified in three main categories: (i) odometric localization, (ii) localization over an a priori known map and (iii) Simultaneous Localization and Mapping (SLAM).

The basic principle of odometric localization is to use some form of velocity measurement to keep track of the robot displacement (relative to the starting location) by numerical integration of the motion model. This is often acceptable for short-range operation of wheeled mobile robots, in which velocity may be determined from proprioceptive sensors such as wheel encoders (*dead reckoning* or *pure odometry*). However, pure odometry is very imprecise for humanoid robots, due to the presence of many sources of uncertainty and inaccuracy in motion execution, such as foot slippage, impacts with the ground, and so on.

Pure odometry integrating encoder and inertial data is used by Chestnutt et al. (2009) to merge successive 3D laser scans and reconstruct local maps of the environment around a humanoid robot; older scans are progressively deleted to reduce the discrepancy effect between local maps due to build-up over time of the pose estimation error.

Electronic supplementary material The online version of this article (doi:10.1007/s10514-015-9498-0) contains supplementary material, which is available to authorized users.

✉ Lorenzo Rosa
rosa@diag.uniroma1.it

Giuseppe Oriolo
oriolo@diag.uniroma1.it

Antonio Paolillo
paolillo@diag.uniroma1.it

Marilena Vendittelli
vendittelli@diag.uniroma1.it

[1] Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Rome, Italy

Published online: 22 September 2015

Effective odometric localization methods usually rely on visual information. In particular, Visual Odometry (VO) is a technique for reconstructing camera displacements by tracking the apparent motion of visual features (Scaramuzza and Fraundorfer 2011). VO has been used to reconstruct the pose of cameras mounted on humanoid robots by Takaoka et al. (2004) and Ozawa et al. (2005). A VO algorithm with improved robustness to motion blur due to walking has been proposed by Pretto et al. (2009).

Another odometric localization approach based on vision relies on a comparison between the current camera image and a set of images stored in advance, e.g., see the work by Ido et al. (2009). This technique is different from VO, in that it requires previous information about the environment.

Methods for humanoid localization that need an a priori known map of the environment have been proposed by Thompson et al. (2006) and Hornung et al. (2010). In these works, measurements from a laser range finder are integrated with odometric data reconstructed from proprioceptive sensors. Visual information from a monocular camera is instead used by Alcantarilla et al. (2013) for localization in a 3D map of the environment.

Among SLAM-based techniques for localization of humanoid robots, we mention the work by Tellez et al. (2008), that makes use of odometric data as well as measurements from laser range finders mounted on the robot feet. Another popular approach is Visual SLAM (VSLAM): for example, Davison et al. (2007) combine a monocular VSLAM algorithm (Davison 2003) with inertial motion reconstruction using an Extended Kalman Filter (EKF). Stasse et al. (2006) use the same VSLAM module considering the reference motion provided by the walking pattern generator as measurements to be fed into the EKF. Other localization methods based on VSLAM use a particle filter based on stereo visual data (Kwak et al. 2009) or integrate a priori knowledge and inertial measurements (Hernandez et al. 2011). In a related method, Ahn et al. (2012) use encoder and inertial data to improve the mobility model within a VSLAM module running on a humanoid.

The method we present in this paper may be classified as odometric localization, because it maintains an estimate of the humanoid pose without requiring or building a map of the environment. This is achieved using measurements from sensors that are found in the standard equipment of most humanoid robots, i.e., joint encoders, foot pressure sensors, a monocular camera in the head, and an inertial measurement unit (IMU) on the torso. In particular, visual information coming from the camera is fed to a vision-based pose estimation algorithm acting as an enhanced sensor and supplying a measurement of the head pose. For this, we use a monocular VSLAM rather than a VO algorithm in view of its higher accuracy, obtained at the cost of an increased computational load which however does not preclude a real-time implementation.

The structure of our algorithm is that of an EKF, in which a pose prediction is computed using the differential kinematics from the support foot to the torso and the relevant joint encoder readings. For the correction, we use as measurements the head pose coming from the VSLAM algorithm and the torso orientation provided by the IMU. The filter is made aware of the current placement of the support foot by an asynchronous update mechanism triggered by the foot pressure sensors.

The main features of the proposed method are:

- no a priori map of the environment is needed;
- rather than considering a generic motion model, we make explicit use of the humanoid kinematics;
- the hybrid (partly continuous-time, partly discrete-time) nature of the walking gait is directly accounted for in the prediction step;
- integration of kinematic data (joint encoders and pressure sensors measurements) with visual information provides robustness with respect to unmodeled effects, such as temporary loss of image features or blur due to sway motion and impacts;
- use of the EKF framework results in a localization system that is amenable to on-board implementation thanks to its light computational load;
- humanoid pose estimates are generated at a high rate, making their use possible as feedback information in a control loop.

With respect to the paper (Oriolo et al. 2012) where this approach was first introduced, the present work adds many technical details on the method and its implementation, as well as an extensive experimental study. In particular, we shall present two sets of experiments. In the first set, humanoid pose estimates computed by our localization system will be compared with the ground truth in a series of open-loop motion trials. In the second set, pose estimates are used in real-time to close the loop in a trajectory control scheme, in order to prove that the proposed localization module can be effectively used for higher-level tasks.

The paper is organized as follows. Section 2 is used to define the relevant quantities and to formulate our odometric localization problem. Section 3 provides a description of the proposed general method, while Sect. 4 details its implementation on the small humanoid robot NAO. The performance of the localization system is then analyzed through several experiments in both open-loop (Sect. 5) and closed-loop (Sect. 6) motion trials. Section 7 concludes the paper.

2 Problem formulation

For a mobile robot, *odometric localization* consists in maintaining a real-time estimate of the robot placement in the world by keeping track of its relative displacements, which are reconstructed from proprioceptive and/or exteroceptive sensor data. To formalize this problem for a humanoid robot, we shall first discuss the basic geometry and then define the sensory equipment used by our localization system.

Figure 1 shows a schematic representation of the robot kinematic structure together with the reference frames of interest, i.e., the fixed world frame $\mathcal{F}_w$ and the moving frames $\mathcal{F}_t$ and $\mathcal{F}_h$, respectively attached to the robot torso and head (Fig. 1a). Let $\boldsymbol{p}_t$, $\boldsymbol{o}_t$ ($\boldsymbol{p}_h$, $\boldsymbol{o}_h$) be the position and orientation of $\mathcal{F}_t$ ($\mathcal{F}_h$) with respect to $\mathcal{F}_w$. The torso and head frames $\mathcal{F}_t$ and $\mathcal{F}_h$ are kinematically related via the neck joints, whose configuration vector is $\boldsymbol{q}_n$ (Fig. 1b).

Denote by $\boldsymbol{R}_t$ the rotation matrix from $\mathcal{F}_w$ to $\mathcal{F}_t$ and by $\boldsymbol{p}_h^t$, $\boldsymbol{R}_h^t$ the position and orientation of $\mathcal{F}_h$ with respect to $\mathcal{F}_t$, and note that both $\boldsymbol{p}_h^t$ and $\boldsymbol{R}_h^t$ are functions of the neck joint angles $\boldsymbol{q}_n$. It is

$$\boldsymbol{p}_h = \boldsymbol{p}_t + \boldsymbol{R}_t\, \boldsymbol{p}_h^t \tag{1}$$

$$\boldsymbol{o}_h = \boldsymbol{\Omega}(\boldsymbol{R}_h) = \boldsymbol{\Omega}(\boldsymbol{R}_t \boldsymbol{R}_h^t), \tag{2}$$

where $\boldsymbol{\Omega}(\cdot)$ is a function that extracts from a rotation matrix the corresponding orientation value in a minimal representation, such as roll-pitch-yaw angles.

Kinematic computations play an important role in our localization algorithm. During locomotion, these computations hinge on the support foot, which represents the base of an open kinematic chain whose endpoint is the origin of $\mathcal{F}_t$. Hence, a *support frame* $\mathcal{F}_s$ is attached to the support foot. Accordingly, define the *support joints* as those located between $\mathcal{F}_s$ and $\mathcal{F}_t$ (i.e, the joints of the support leg and those of the pelvis), and denote their configuration by $\boldsymbol{q}_s$. Upon completion of each step, $\mathcal{F}_s$ jumps to a new placement, which is the landing posture of the former swinging foot; the support joints are also appropriately redefined (see Fig. 2).

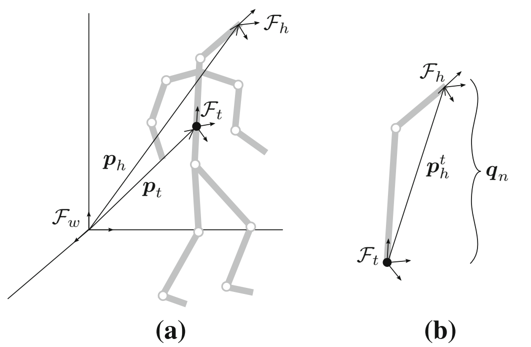


Fig. 1 Relevant frames for humanoid localization: **a** $\mathcal{F}_w$ (world), $\mathcal{F}_t$ (torso) and $\mathcal{F}_h$ (head); **b** enlarged view of the kinematic chain from the torso to the head, with the associated neck joint variables $\boldsymbol{q}_n$

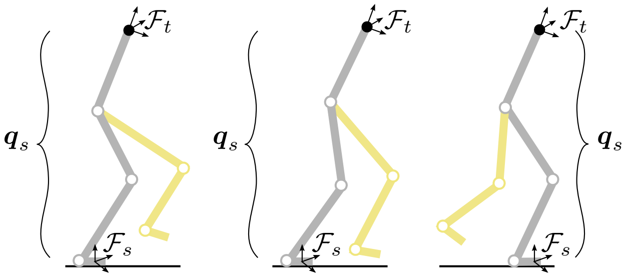


Fig. 2 During locomotion, the base of the kinematic chain is the support foot. At each step completion, its representative frame is discontinuously moved to a new placement. The identity of the support joints in $\boldsymbol{q}_s$ is changed in accordance

Coming to sensory requirements, we shall assume the availability of a monocular camera in the robot head, an IMU on the torso, encoders at the joints and pressure sensors under the feet. This is a rather standard equipment for a humanoid platform. In our localization filter, encoders and pressure sensors provide data for kinematic state prediction, while camera and IMU are involved in the measurement model. In particular:

- We assume that camera images are fed to an off-the-shelf VSLAM algorithm which reconstructs the position and orientation $\boldsymbol{p}_h$, $\boldsymbol{o}_h$ of the head frame $\mathcal{F}_h$. The ensemble of camera and VSLAM algorithm is considered as an intelligent visual sensor, which is used as a black box.
- As for the IMU, only the measurement of the torso orientation $\boldsymbol{o}_t$ will be used, because velocity data are typically very inaccurate or downright unavailable, as in the NAO platform used for our experiments.

The localization problem can now be precisely formulated as follows. Given initial estimates $\hat{\boldsymbol{p}}_{t,0}$ and $\hat{\boldsymbol{o}}_{t,0}$ for the humanoid torso position and orientation, provide continuously updated estimates $\hat{\boldsymbol{p}}_t$, $\hat{\boldsymbol{o}}_t$ as the robot moves, using measurements of $\boldsymbol{p}_h$, $\boldsymbol{o}_h$ coming from the camera+VSLAM sensor, measurements of $\boldsymbol{o}_t$ yielded by the IMU, joint readings from the encoders and signals from the pressure sensors.

Once an estimate of the torso position and orientation is available, one may clearly reconstruct the corresponding information for any other part of the robot body through direct kinematics.

3 The proposed method

The proposed method for odometric localization of humanoid robots follows the typical prediction-correction structure of an EKF. At each sampling instant, a prediction of the torso position and orientation is obtained using the differential kinematic map from $\mathcal{F}_s$ to $\mathcal{F}_t$ and the encoder readings for the support joints. A correction is then computed on the basis of the difference between the expected values of the outputs (head pose and torso orientation) and their measurements coming from the camera + VSLAM sensor and the IMU. Information about the current placement of the support foot is provided to the filter by an asynchronous updating mechanism triggered by the signals from foot pressure sensors. Below, we describe in detail the proposed algorithm.

Let $\boldsymbol{x} = (\boldsymbol{p}_t, \boldsymbol{o}_t)$ be the *pose* of the torso frame $\mathcal{F}_t$, i.e., its position and orientation with respect to the world frame $\mathcal{F}_w$. Our filter will take $\boldsymbol{x}$ as state to be estimated. Denote by $\boldsymbol{o}_s$ the orientation of $\mathcal{F}_s$ with respect to $\mathcal{F}_w$, and by $\boldsymbol{J}(\boldsymbol{q}_s, \boldsymbol{o}_s)$ the Jacobian matrix of the kinematic map from the support frame $\mathcal{F}_s$ to $\mathcal{F}_t$ (note that $\boldsymbol{J}$ does not depend on the position of $\mathcal{F}_s$). We adopt the following state-transition model for $\boldsymbol{x}$:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q}_s, \boldsymbol{o}_s)\dot{\boldsymbol{q}}_s. \tag{3}$$

Equation (3) is a *kinematic* model, with the velocities $\dot{\boldsymbol{q}}_s$ of the support joints acting as control inputs. There are essentially three reasons for not using a state-transition model based on robot *dynamics*. First, the full dynamic equations of humanoid platforms are often not available. Second, they are in any case very complex, and their numerical integration would be too time-consuming for real-time localization. Finally, the appropriate control inputs for a dynamic model are the joint torques, typically not accessible for measurements.

Note that the evolution of $\boldsymbol{q}_s$ and $\boldsymbol{o}_s$ is not described by the state-transition model (3). Indeed, in the proposed filter the value of $\boldsymbol{q}_s$ is simply read from joint encoders, while $\boldsymbol{o}_s$ is asynchronously updated at the completion of a step by appropriately processing the output signals coming from the pressure sensors. See the next section for a detailed description of such updating mechanism in the case of a NAO humanoid.

The output model needed to derive the localization filter expresses the measured variables $\boldsymbol{y}$ as a function of the system state $\boldsymbol{x}$. In particular, $\boldsymbol{y}$ includes the head pose $(\boldsymbol{p}_h, \boldsymbol{o}_h)$ and the torso orientation $\boldsymbol{o}_t$, respectively provided by the camera+VLSAM sensor and the IMU. Using Eqs. (1–2) we obtain

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{q}_n) = \begin{pmatrix} \boldsymbol{p}_t + \boldsymbol{R}_t \boldsymbol{p}_h^t \\ \boldsymbol{\Omega}(\boldsymbol{R}_t \boldsymbol{R}_h^t) \\ \boldsymbol{o}_t \end{pmatrix}. \tag{4}$$

Note on the dependence of the output $\boldsymbol{y}$ on $\boldsymbol{q}_n$ (the configuration of the neck joints) through $\boldsymbol{p}_h^t$ and $\boldsymbol{R}_h^t$. As done for $\boldsymbol{q}_s$ in the state-transition model (3), the value of $\boldsymbol{q}_n$ to be used in (4) is read from joint encoders.

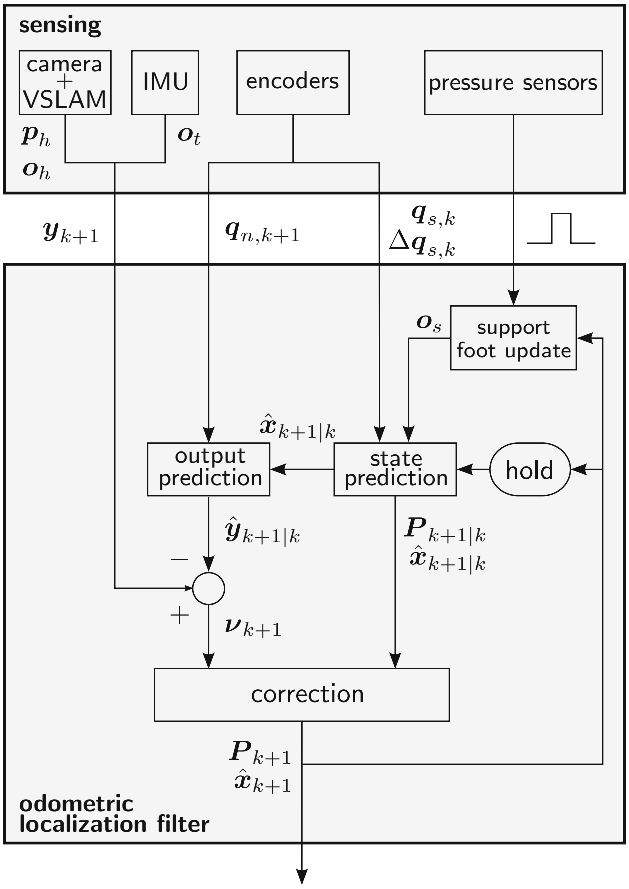


Fig. 3 Block diagram representing the proposed filter for odometric localization. Note that signals coming from pressure sensors are used to trigger an asynchronous updating mechanism which makes the filter aware of the current orientation of the support foot

Let T be the sampling interval of the filter, and use the subscript k to indicate the value that a variable assumes at time kT. The deterministic state-transition model (3–4) translates to the discrete-time stochastic system

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + T\,\boldsymbol{J}(\boldsymbol{q}_{s,k}, \boldsymbol{o}_s)\dot{\boldsymbol{q}}_{s,k} + \boldsymbol{v}_k \tag{5}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{q}_{n,k}) + \boldsymbol{w}_k, \tag{6}$$

where $\boldsymbol{v}_k$, $\boldsymbol{w}_k$ are zero-mean white gaussian noises with covariance matrices $\boldsymbol{V}_k \in \mathbb{R}^{6\times 6}$, $\boldsymbol{W}_k \in \mathbb{R}^{9\times 9}$, respectively. Note that $\boldsymbol{o}_s$ does not get the k subscript because it is updated asynchronously.

We shall now provide explicit equations for our odometric localization filter, whose functional structure is illustrated in Fig. 3.

3.1 State prediction

At the time sample t_{k+1}, a prediction $\hat{\boldsymbol{x}}_{k+1|k}$ is generated from the current estimate $\hat{\boldsymbol{x}}_k$ using Eq. (5):

$$\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_k + \boldsymbol{J}(\boldsymbol{q}_{s,k}, \boldsymbol{o}_s)\Delta\boldsymbol{q}_{s,k},$$

where the vector increment $\Delta \boldsymbol{q}_{s,k} = \boldsymbol{q}_{s,k+1} - \boldsymbol{q}_{s,k}$ in encoder readings is used to approximate the velocity input term $T\,\dot{\boldsymbol{q}}_{s,k}$.

From the structure of the discrete-time system (5), the accompanying covariance prediction follows:

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{P}_k + \boldsymbol{V}_k. \tag{7}$$

3.2 Output prediction

The predicted output associated to the predicted state $\hat{\boldsymbol{x}}_{k+1|k}$ is computed using (6):

$$\hat{\boldsymbol{y}}_{k+1|k} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{q}_{n,k+1}\right).$$

The value $\boldsymbol{q}_{n,k+1}$ of the neck joint variables at time t_{k+1} to be used in this computation is provided by the corresponding joint encoders.

3.3 Correction

To correct the predicted state, we first compute the innovation, i.e., the difference between the measured and the predicted output:

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}.$$

The corrected state estimate is then defined as

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{G}_{k+1}\boldsymbol{\nu}_{k+1},$$

where $\boldsymbol{G} \in \mathbb{R}^{6\times 9}$ is the Kalman gain matrix

$$\boldsymbol{G}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^T\left(\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^T + \boldsymbol{W}_{k+1}\right)^{-1}$$

with

$$\boldsymbol{H}_{k+1} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}}.$$

The actual expression of $\boldsymbol{H}_{k+1}$ will depend on the specific choice of the coordinates for representing orientation, that is embedded in the function $\boldsymbol{\Omega}(\cdot)$.

The corrected covariance estimate is

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{G}_{k+1}\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k}.$$

3.4 Support foot update

Whenever the completion of a step is detected on the basis of the foot pressure sensors, the support frame $\mathcal{F}_s$ is instantaneously displaced to the new pose. As already noticed, the differential kinematic map (3) does not depend on the position of $\mathcal{F}_s$. Therefore, only its orientation $\boldsymbol{o}_s$ is updated, using a forward kinematic computation from $\mathcal{F}_t$ (now acting as a base frame placed at its estimated pose) to $\mathcal{F}_s$, and reading the value of the new support joints from the corresponding encoders (see Fig. 2).

4 Experimental setup

The proposed localization method has been validated on the humanoid robot NAO developed by Aldebaran robotics. This section describes the relevant robot hardware, the ground truth system used to assess the localization accuracy and the distinctive features of the filter instance developed for the localization of NAO.

4.1 Robot hardware

Figure 4 reports a schematic of the robot structure with the kinematic chains and the reference frames of interest. NAO has 5 degrees of freedom in each leg, 1 in the pelvis, and 2 in the neck; therefore, for the support and neck joints we have respectively $\boldsymbol{q}_s \in \mathbb{R}^6$ and $\boldsymbol{q}_n \in \mathbb{R}^2$. The frame $\mathcal{F}_s$ is placed on the support foot, aligned with the ankle articulation. The torso frame $\mathcal{F}_t$ has been chosen as one of the three spatial references used by the API methods of NAO, while the head frame $\mathcal{F}_h$ has been placed on the top of the head, aligned with the neck articulation.

The frame $\mathcal{F}_c$ has origin in the focus of the camera that provides the information used by the VSLAM algorithm embedded in the localization filter. This is a CMOS digital camera with a diagonal FOV of 72.6°. By taking images with a resolution of 320 × 240 pixels it is possible to obtain a frame rate of 30 Hz, which is the maximum frequency allowed for image acquisition.

The sensor suite necessary to implement the localization method is completed by an IMU located in the chest, the magnetic rotary encoders mounted on each joint, and the Force Sensitive Resistors (FSRs) placed under each foot. The IMU yields roll and pitch angles measurements relative to the torso frame $\mathcal{F}_t$; hence, for the output vector of Eq. (4) we have $\boldsymbol{y} \in \mathbb{R}^8$, being the yaw angle measurement not provided. The encoders provide measures of the joint angles for all the kinematic computations required by the filter with a resolution of 0.1°. The pressure measurements coming from the FSRs are used to generate the switching signal for the support foot update. Measures from IMU, encoders and FSRs are updated at a nominal rate of 100 Hz.

4.2 Ground truth system

An external fixed camera has been used within a ground truth system to asses the filter performance. In particular, using the algorithm in (Garrido-Jurado et al. 2014) it is possible to

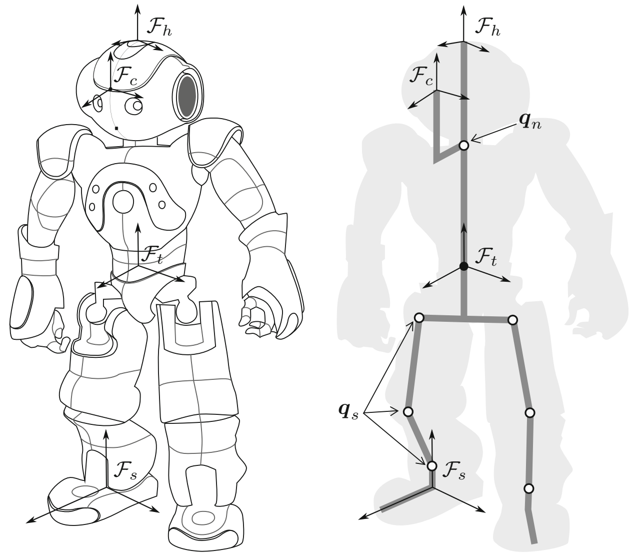


Fig. 4 The humanoid robot NAO with the kinematic chains and the reference frames involved in the localization process

obtain accurate estimates of the robot head pose by tracking a known textured marker positioned on the robot head. The estimation error between the ground truth and the odometric localization system proposed in this work is derived by propagating the estimates of the torso pose provided by the odometric filter up to the head frame.

4.3 Filter implementation

The VSLAM method for estimating the head pose chosen in our implementation is the ready-to-use Parallel Tracking and Mapping (PTAM) algorithm developed by Klein and Murray (2007) for augmented reality applications. In particular, we have seamlessly integrated in our framework the reference implementation of PTAM available for free download.[1]

PTAM robustly tracks the pose of a hand-held monocular camera in unknown environments through the construction of a dense map of 3D point features collected from video frames. Real-time operation of the algorithm is obtained through the parallel running of the tracking and mapping processes. In the present work, PTAM is used to estimate the pose of the camera frame $\mathcal{F}_c$ (see Fig. 4). The head pose used by the filter is readily obtained through a rigid transformation from $\mathcal{F}_c$ to $\mathcal{F}_h$.

The maximum output frequency of PTAM is limited by the camera frame rate (30 Hz). Thus, our kinematic EKF runs at the same rate, although multiple prediction steps are taken to exploit the higher rate of the IMU and encoders readings (100 Hz).

In our current implementation, PTAM runs on an external desktop computer to avoid overburdening the robot CPU. Optimized, computationally lighter versions of PTAM suitable for on-board implementation are available in the literature (Weiss et al. 2011) and can be easily integrated in our filter. The main reasons for using the full version of PTAM in this work are: (i) it is available as open-source software; (ii) we want to compare the performance of purely visual estimation with that achieved by fusion of visual, inertial and kinematic data.

Finally, it is important to mention that, being based on monocular vision, PTAM needs an initial guess of the world metric scale since it cannot be recovered from pure image measurements (Weiss and Siegwart 2011). Therefore, all our experiments include an initialization procedure consisting in a translation of the robot head by a known quantity so as to build a stereo pair from two views of the same scene. Matching of the image features detected on them allows PTAM to recover the depth information and to lay the basis for the construction of its 3D visual map. The needed sideways translation of the camera is obtained in our experimental setup by using a built-in NAO function allowing to command a 3D motion of the head.

A peculiar aspect of the filter instance proposed in this work is the asynchronous update of the support foot identity. This is triggered by a switching signal obtained by processing the FSRs output. For illustration, in Fig. 5 are plotted the relevant signals of this elaboration process relative to a six-steps walking motion segment. The uppermost plot reports the FSRs signals filtered from measurement noise.

The dashed horizontal lines indicate two threshold values corresponding respectively to the inception of a contact between the foot and the ground and the beginning of a single support phase during which the weight of the robot is completely supported by one of the two feet, the other one being swinging.

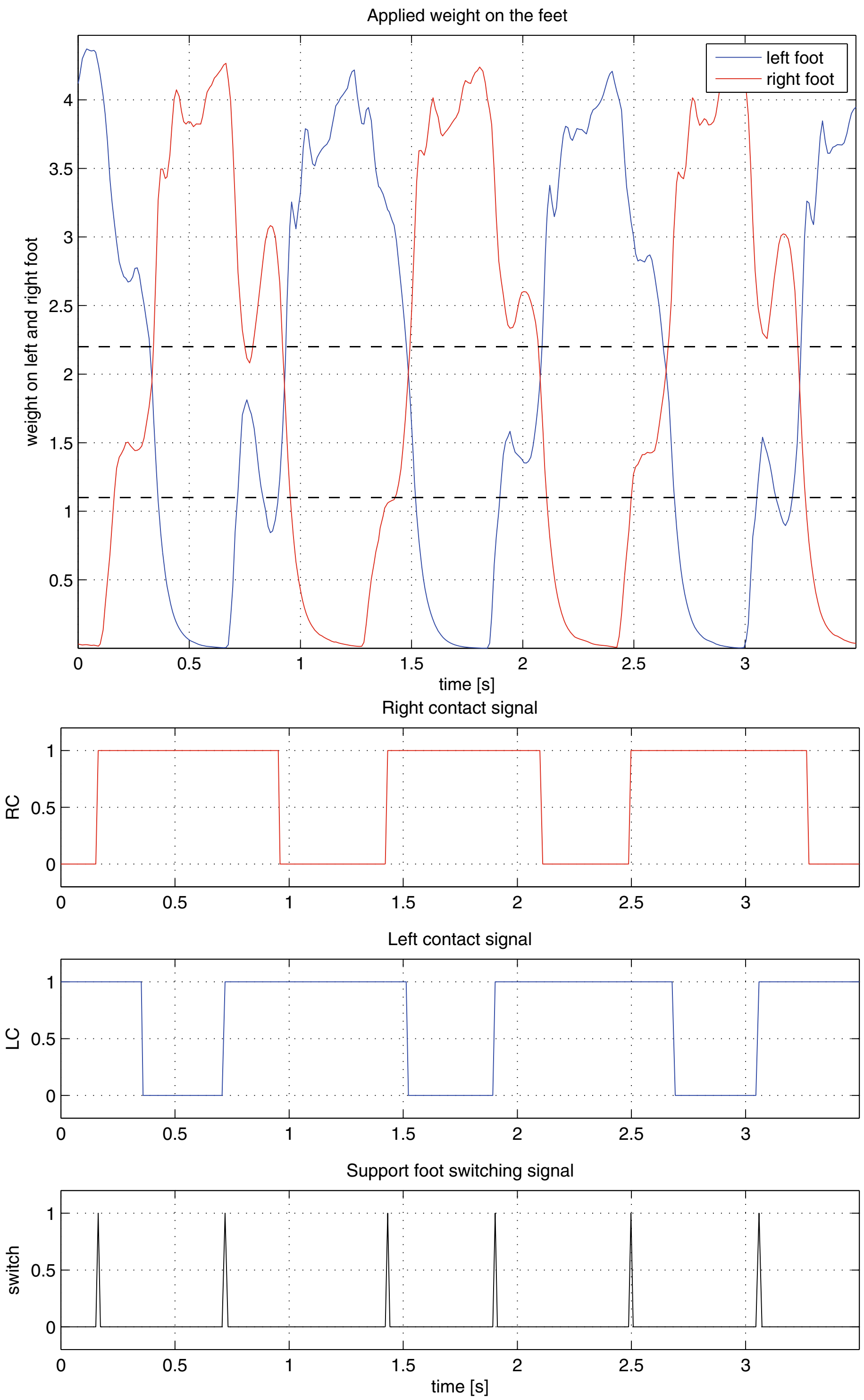
The two thresholds are used to generate the contact signals RC, for the right foot, and LC, for the left foot. Specifically, when a filtered FSR signal raises above the lower threshold the corresponding contact signal switches to the high logical level if it is currently in the low level. Complementarily, a FSR signal crossing first the high and then the low level threshold in sequence makes the corresponding contact signal switch to the low logical level if it is currently in the high level state. This switching logic introduces a kind of hysteresis in the generation of the contact signals that filters the signal chattering around one of the two thresholds typical of the initial phase of a change in the contact foot.

The impulsive signal that triggers the support foot update is generated by the raising edge of either RC or LC as illustrated by the last plot of Fig. 5.

It is worth noticing at this point that NAO APIs provide functions for the detection of the support foot change and

[1] https://github.com/Oxford-PTAM/PTAM-GPL.

Fig. 5 Generation of the impulsive signal for the support foot update from the FSRs measures



for the kinematics-based odometric localization of the robot. These functions could be used to predict the robot state using the proposed framework. However, the objective of this work was to develop and implement a general method easily extendible to other humanoids. In addition, since the technical details behind these functions are not known, it would be not possible to have full control on signal synchronization.

To complete the description of our implementation of the proposed localization method, we mention that all the results reported in the following sections have been obtained by assigning to the covariance matrices the following values:

$$\boldsymbol{V} = \mathrm{diag}\{5, 5, 5, 100, 100, 100\} \times 10^{-6}$$
$$\boldsymbol{W} = \mathrm{diag}\{5, \ldots, 5, 5 \cdot 10^{-4}, 5, 5\} \times 10^{-2},$$

with appropriate measurement units, i.e., meters for the entries relative to the position and radians for those related to the orientation.

These values have been found through an extensive experimental study and reflect the quite high accuracy in the prediction of the torso pose based on encoders measures, the uncertainties being introduced essentially by the contact dynamics. A low value of the covariance in $\boldsymbol{W}$ is associated to the measure of the head yaw angle provided by PTAM for two reasons: (i) experiments have demonstrated the reliability of PTAM in estimating this state variable, (ii) no exteroceptive sensor other than the camera (through PTAM) provides measures for the correction of the robot heading direction. These matrices have been kept constant through the reported experiments.

5 Direct validation via localization experiments

This section reports on a set of experiments aimed at assessing the quality and reliability of the humanoid pose estimation achievable through the proposed method. The presented results have been obtained by filtering the data collected during the execution of fundamental locomotion tasks in different environmental conditions.

In particular, open-loop walking on flat floor[2] along a straight line and along a circle are first considered as primitives of human-like locomotion on long distances (Mombaur et al. 2010; Truong et al. 2010). The *move* function, available in the NAO software suite, allows to specify the desired forward and angular speed of a mobile frame with origin between the robot feet and x-axis pointing forward along the robot sagittal plane.

To fully evaluate the accuracy of the method, no calibration procedure has been run to eliminate systematic errors, though the methods in (Kelly 2004; Hornung et al. 2014) can be used to improve the prediction.

In the first experiment NAO is commanded to walk straight. Figure 6 shows the top view of the robot head cartesian motion. Data obtained from the robot built-in odometry throughout the function *getPosition* are plotted in black, while the path followed by the robot as reconstructed by the ground truth system is shown in red. The filter estimate is plotted in green.

A comparison of odometry and ground truth plots shows the fast deviation of NAO from the commanded path. The plot in green visually assesses the accuracy of the filter in reconstructing the real robot motion. A quantitative evaluation of the filter accuracy is, instead, provided by the norm of the 3D cartesian error introduced by the proposed filter and by the NAO odometry with respect to the ground truth reported in Fig. 7. While the error of the built-in odometry module diverges, the error associated to the filter estimates remains limited. More specifically, the root mean square of the filter absolute cartesian error is $e_{\text{rms}} = 0.01873$ m. Correspondingly, the root mean square of the built-in odometry error is $e_{\text{rms}} = 0.095358$ m.

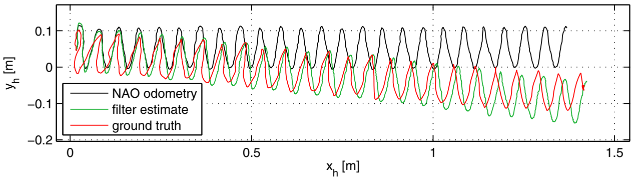


Fig. 6 First localization experiment: *line*. NAO odometry and filter estimate versus ground truth

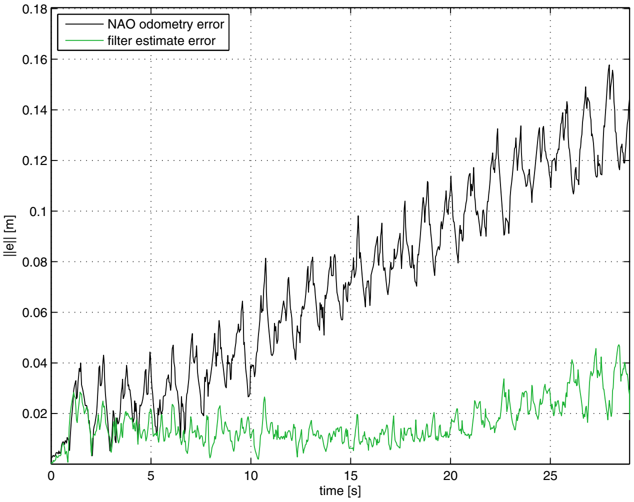


Fig. 7 First localization experiment: *line*. Norm of the head position error with respect to the ground truth

In the second experiment NAO is commanded to walk counterclockwise along a circle. The top view of the robot head motion as reconstructed respectively from the ground truth data, the robot built-in odometry and the presented filter is reported in Fig. 8. At the start, the robot right foot is positioned at (0, 0).

Note how, although still accurate, the quality of localization deteriorates with that of the head pose estimate provided by PTAM [compare with Fig. 9, around the point (0.3, 0.7)]. This loss of accuracy is essentially due to a fluctuation in the number and quality of the image features tracked by PTAM from frame to frame. In particular, most of the features acquired during the initialization phase go out of the camera FOV quite early due to the change in NAO's heading along the circle. This forces PTAM to expand the map by

[2] In principle, our localization method can be used on terrain with variable slopes. However, using the NAO built-in locomotion functions, relying on the flat floor assumption, it is only possible to allow very small variations in the slope that are not distinguishable from measurement noise. With sufficiently high slope values the robot falls down.

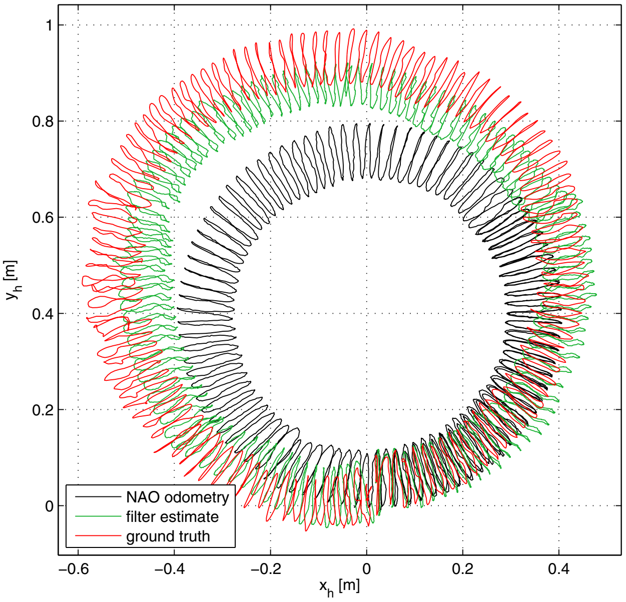


Fig. 8 Second localization experiment: *circle*. NAO odometry and filter estimate versus ground truth

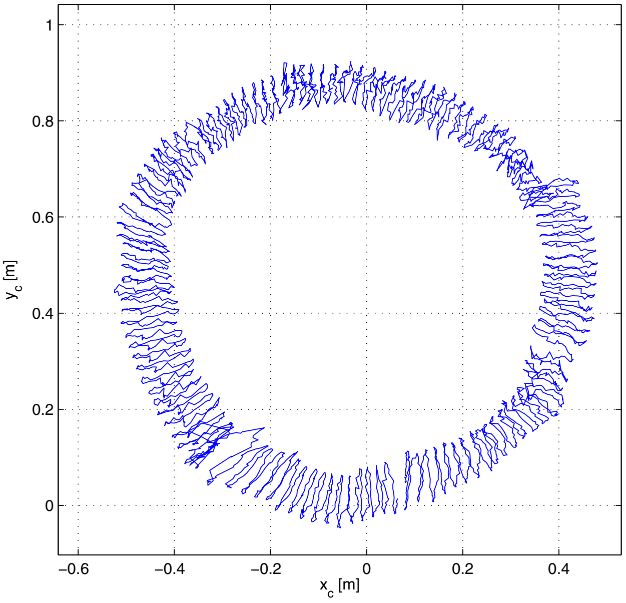


Fig. 9 Second localization experiment: *circle*. Camera position provided by the VSLAM algorithm PTAM

adding new features. The quality of the newly added features is variable and heavily dependent on the environment. The filter error is however recovered in the last part of the circle when PTAM finds the image features of the initial part of the map and can exploit the typical SLAM loop closure.

The positive effect of the loop closure can be further appreciated from the results of a third experiment in which NAO travels two rounds along the same ideal circle. The reconstructed robot path is reported in Fig. 10 that confirms the reliability of the proposed filter and in particular its ability to recover the estimation error when a loop closure is possible. Figure 11 shows the evolution of the norm of the cartesian error generated by the filter and the built-in odometry error. The effect of the loop closure is to maintain limited the filter error while the error associated to the built-in odometry diverges through the rounds.

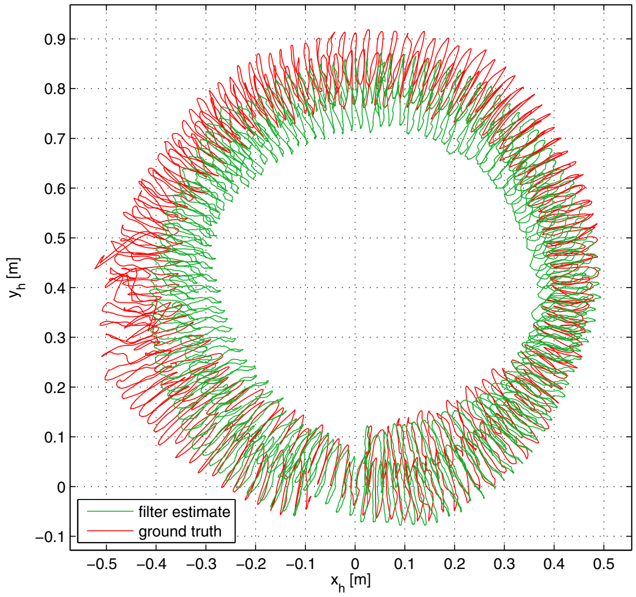


Fig. 10 Third localization experiment: *circle*, *two rounds*. Filter estimate versus ground truth

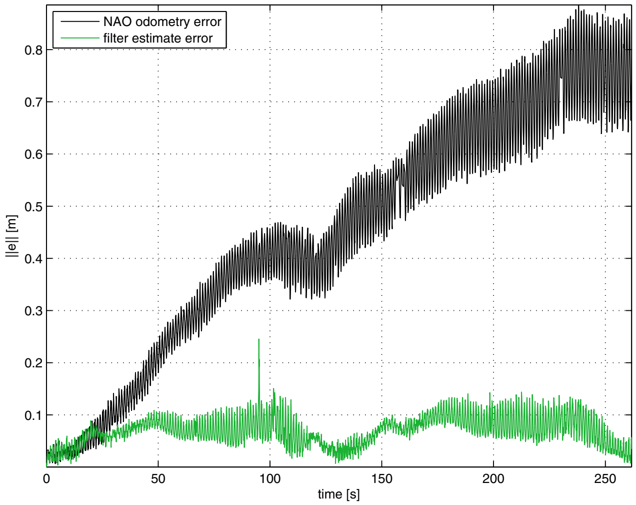


Fig. 11 Third localization experiment: *circle*, *two rounds*. Norm of the head position error with respect to the ground truth

It is worth noting that, in our experimental setup the heading angle correction is only due to PTAM since the robot IMU does not provide this information. For humanoids equipped with an IMU that provides the complete robot attitude the

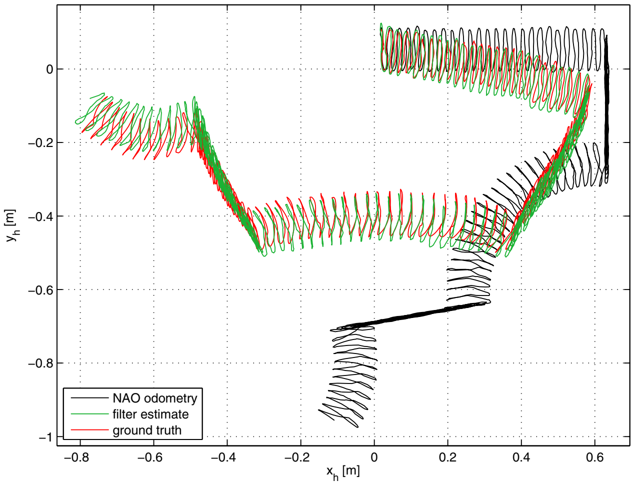


Fig. 12 Fourth localization experiment: omnidirectional walk toward randomly varied directions. NAO odometry and filter estimate versus ground truth

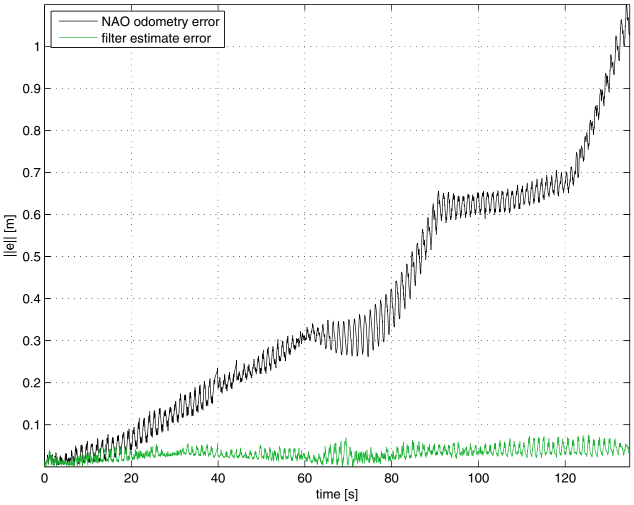


Fig. 13 Fourth localization experiment: omnidirectional walk toward randomly varied directions. Norm of the head position error with respect to the ground truth

filter estimate is expected to be more robust to PTAM inaccuracies.

To further test the performance of the proposed localization method, we have run experiments exploiting the omnidirectional motion capability of humanoids. Figure 12 shows the top view of the robot head cartesian motion. Each segment of the walking motion correspond to a fixed set of parameters in the function *setWalkTargetVelocity* accepting as input the desired displacement in the x and y direction, rotation of the swinging foot with respect to the supporting one and frequency of steps. Plot of the cartesian error in Fig. 13 shows that the accuracy remains within the bounds of the previous experiments.

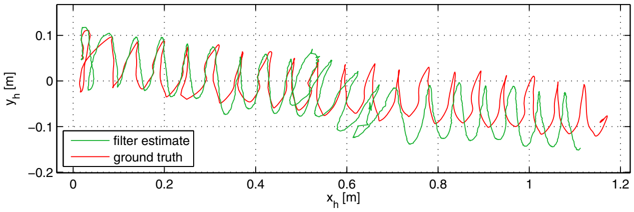


Fig. 14 Fifth localization experiment: temporary camera occlusion. Filter estimate versus ground truth

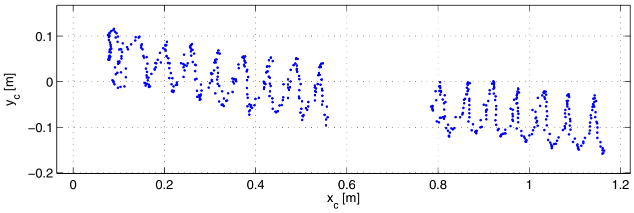


Fig. 15 Fifth localization experiment: temporary camera occlusion. Camera position provided by PTAM

The last experiment of this section illustrates the impact of a lack of measurement updates on the localization accuracy. Measures may lack when PTAM is unable to extract any feature from an image because of a temporary occlusion. In the proposed experiment this situation has been simulated by artificially occluding the camera FOV for about 10 seconds. The cartesian motion reconstructed through the filter and the ground truth is shown in Fig. 14, while Fig. 15 reports the measurements provided by PTAM. When the camera FOV is occluded PTAM stops providing any output. This obviously increases the error in the estimates provided by the filter which is now based on the prediction only. In practical applications, when pose estimates are needed to perform a feedback control action, the proposed filter still provides a useful signal. The use of a pure VSLAM might instead lead to a lack of measurements, hence preventing the computation of the control inputs.

6 Indirect validation via control experiments

This section reports on control experiments using as feedback signal the state estimate provided by the proposed filter. In particular, the presented results have been obtained by applying the control and sway motion filtering techniques derived in (Oriolo et al. 2013) for tracking workspace trajectories with the torso motion after cancellation of the sway oscillations.

The evolution of this controlled output is associated to a unicycle-like model which corresponds to a natural locomotion behavior for long distance displacements (Mombaur et al. 2010; Truong et al. 2010). With this model, any trajec-

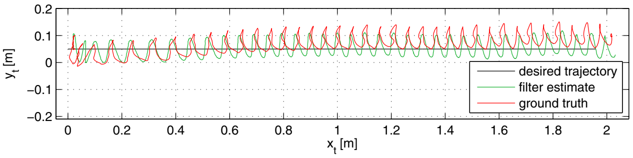


Fig. 16 Trajectory tracking: *straight line*. Filter estimate and ground truth versus desired trajectory

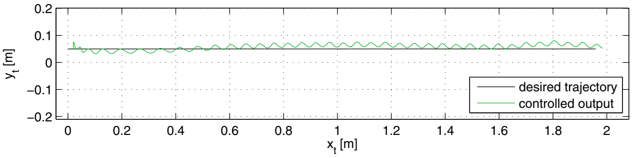


Fig. 17 Trajectory tracking: *straight line*. Estimated torso trajectory after sway motion cancellation (controlled output) versus desired trajectory

tory tracking controller designed for unicycle-like robots is suitable to drive the humanoid along a specified path.

In the reported experiments we have used the tracking controller proposed in (Samson 1993), which consists of a nonlinear time-invariant controller. The driving and steering velocity commands issued from the controller are sent to the robot using the NAO built-in function *move*. The sway motion is canceled through low-pass filtering, one of the two solution methods proposed in (Oriolo et al. 2013). The trajectory controller runs at 100 Hz, the same rate of the localization filter.

For the first experiment, the desired trajectory is a straight line to be executed at the speed of 0.5 m/s. The executed path is reported in Fig. 16 showing the top view of the torso motion as reconstructed by the odometric localization filter (in green) and by the ground truth system (in red), and the reference trajectory (the black line). The accuracy of the localization filter is, as expected, the same of the open-loop experiments and a qualitative comparison with Fig. 6 suggests that it is suitable for use in the control loop.

A quantitative evaluation of the tracking accuracy can be obtained from the evolution of the controlled output, given by the estimated torso position after sway motion cancellation, shown in Fig. 17 together with the reference linear trajectory.

The transient error associated to the starting phase of the walking motion is due both to the error in the robot initial pose and to the foot slippage usually accompanying the first step of the walking motion. As soon as the walking gait becomes regular, the control action is more effective and the executed path converges to the desired trajectory. Overall, the robot tracks the desired trajectory with a root mean square of the tracking error $e_{\mathrm{rms}} = 0.0388$ m.

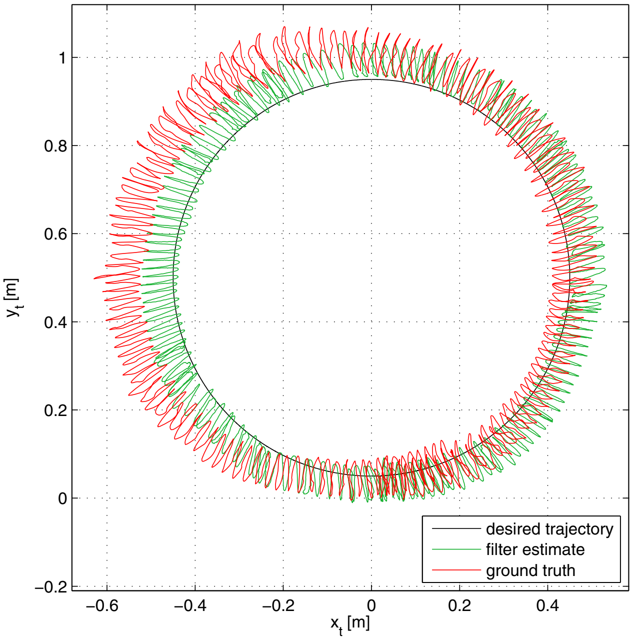


Fig. 18 Trajectory tracking: *circle*. Estimated and ground truth torso versus desired trajectory

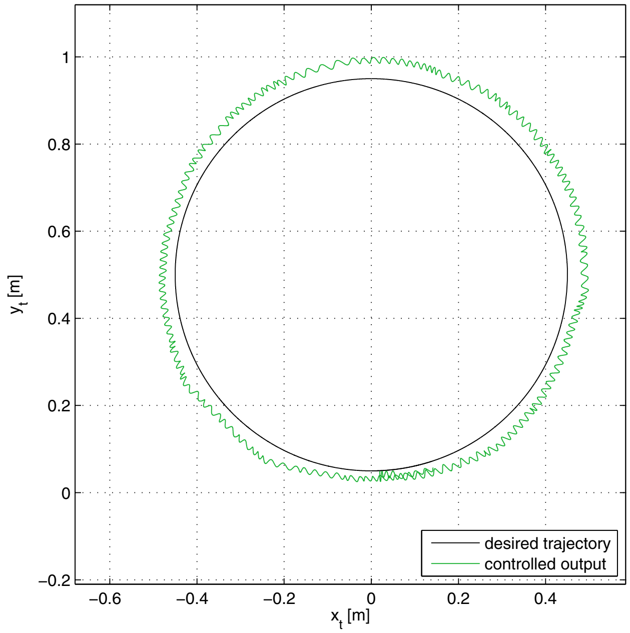


Fig. 19 Trajectory tracking: *circle*. Estimated torso trajectory after sway motion cancellation versus desired trajectory

The reference trajectory in the second experiment is a circle to be traced at the linear speed of 0.018 m/s and angular speed of 0.04 rad/s. Figure 18 allows to compare the estimated, ground truth and desired torso trajectory. The averaged (i.e., after sway motion cancellation) torso motion is shown in Fig. 19. The root mean square of the tracking error for this experiment is $e_{\mathrm{rms}} = 0.0488$ m.

It is worth observing that, the residual tracking error is to be ascribed not only to the localization inaccuracy but also to the trajectory control algorithm which is not robust with respect to the perturbation introduced in considering an approximate model for control design and to NAO's actuation errors. This error can be reduced but not eliminated. We do not analyze the controller robustness properties because trajectory tracking is not the focus of the paper and the real localization accuracy should be evaluated on the open-loop experiments. The above reported experiments aim at showing the real-time performance of the algorithm.

Movie clips illustrating both open-loop and control experiments are included in the video accompanying this paper.

7 Conclusions

We have presented a method for odometric localization of humanoid robots that integrates kinematic, inertial and visual information in an EKF framework. At each sampling instant, a pose for the torso is predicted using the differential kinematic map from the current support foot to the torso itself using the joint encoders value. This prediction is then corrected using the measurements from the camera (head pose reconstructed through a VSLAM algorithm) and the IMU (torso orientation). The support foot is updated asynchronously at the detection of a step completion through the pressure sensors under the feet.

Open-loop locomotion trials with the small size humanoid NAO have been used to directly assess the accuracy of the proposed localization method with respect to a ground truth. Control experiments using the humanoid pose estimates in real-time as feedback signals for tracking a desired workspace trajectory have shown that the localization module is suitable for use in an integrated control architecture for the execution of higher level tasks.

Salient software components of our implementation of the proposed filter is available for download.[3] We designed the software to be as much as possible independent from middleware frameworks like ROS or specific robot software libraries. Accordingly, we used NAO APIs only to access sensor data and to map the velocity commands to the robot walking engine.

References

Ahn, S., Yoon, S., Hyung, S., Kwak, N., & Roh, K. S. (2012). On-board odometry estimation for 3D vision-based SLAM of humanoid robot. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4006–4012).

Alcantarilla, P., Stasse, O., Druon, S., Bergasa, L., & Dellaert, F. (2013). How to localize humanoids with a single camera? *Autonomous Robots*, *34*(1–2), 47–71.

Chestnutt, J., Takaoka, Y., Suga, K., Nishiwaki, K., Kuffner, J., & Kagami, S. (2009). Biped navigation in rough environments using on-board sensing. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3543–3548).

Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *9th International Conference on Computer Vision* (pp. 1403–1410).

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Garrido-Jurado, S., Muñoz Salinas, R., Madrid-Cuevas, F. J., & Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, *47*(6), 2280–2292.

Hernandez, E., Ibarra, J. M., Neira, J., Cisneros, R., & Lavín, J. E. (2011). Visual SLAM with oriented landmarks and partial odometry. In *21st IEEE International Conference on Electrical Communications and Computers* (pp. 39–45).

Hornung, A., Wurm, K. M., & Bennewitz, M. (2010). Humanoid robot localization in complex indoor environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1690–1695).

Hornung, A., Osswald, S., Maier, D., & Bennewitz, M. (2014). Monte carlo localization for humanoid robot navigation in complex indoor environments. *International Journal of Humanoid Robotics*, *11*(02), 1441002.

Ido, J., Shimizu, Y., Matsumoto, Y., & Ogasawara, T. (2009). Indoor navigation for a humanoid robot using a view sequence. *International Journal of Robotics Research*, *28*(2), 315–325.

Kelly, A. (2004). Fast and easy systematic and stochastic odometry calibration. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 4, 3188–3194).

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (pp. 225–234).

Kwak, N., Stasse, O., Foissotte, T., & Yokoi, K. (2009). 3D grid and particle based SLAM for a humanoid robot. In *2009 9th IEEE-RAS International Conference on Humanoid Robots* (pp. 62–67).

Mombaur, K., Truong, A., & Laumond, J.-P. (2010). From human to humanoid locomotion—An inverse optimal control approach. *Autonomous Robots*, *28*, 369–383.

Oriolo, G., Paolillo, A., Rosa, L., & Vendittelli, M. (2012) Vision-based odometric localization for humanoids using a kinematic EKF. In *2012 12th IEEE-RAS International Conference on Humanoid Robots* (pp. 153–158).

Oriolo, G., Paolillo, A., Rosa, L., & Vendittelli, M. (2013). Vision-based trajectory control for humanoid navigation. In *2013 13th IEEE-RAS International Conference on Humanoid Robots* (pp. 118–123).

Ozawa, R., Takaoka, Y., Kida, Y., Nishiwaki, K., Chestnutt, J., Kuffner, J., Kagami, J., Mizoguch, H., & Inoue, H. (2005). Using visual odometry to create 3D maps for online footstep planning. In *2005 IEEE International Conference on Systems, Man, and Cybernetics* (Vol. 3, pp. 2643–2648).

Pretto, A., Menegatti, E., Bennewitz, M., Burgard, W., & Pagello, E. (2009). A visual odometry framework robust to motion blur. In *2009 IEEE International Conference on Robotics and Automation* (pp. 2250–2257).

Samson, C. (1993). Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research*, *12*(1), 55–64.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry part I: The first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, *18*(4), 80–92.

Stasse, O., Davison, A. C., Sellaouti, R., & Yokoi, K. (2006). Real-time 3D SLAM for a humanoid robot cosidering pattern generator infor-

[3] http://www.dis.uniroma1.it/~labrob/research/VisOdoLoc4Hum/software.

mation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 348–355).

Takaoka, Y., Kida, Y., Kagami, S., Mizoguchi, H., & Kanade, T. (2004). 3D map building for a humanoid robot by using visual odometry. In *2004 IEEE International Conference on Systems, Man, and Cybernetics* (Vol. 5, pp. 4444–4449).

Tellez, R., Ferro, F., Mora, D., Pinyol, D., & Faconti, D. (2008). Autonomous humanoid navigation using laser and odometry data. In *2008 8th IEEE-RAS International Conference on Humanoid Robots* (pp. 500–506).

Thompson, S., Kagami, S., & Nishiwaki, K. (2006). Localisation for autonomous humanoid navigation. In *2006 IEEE-RAS International Conference on Humanoid Robots* (pp. 13–19).

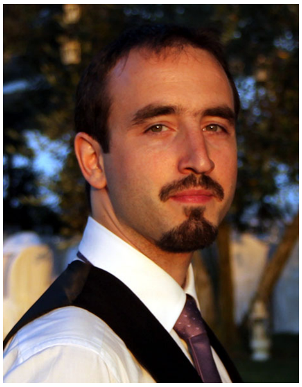
Truong, T.-V.-A., Flavigne, D., Pettre, J., Mombaur, K., & Laumond, J.-P. (2010). Reactive synthesizing of human locomotion combining nonholonomic and holonomic behaviors. In *3rd IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (pp. 632–637).

Weiss, S., & Siegwart, R. (2011). Real-time metric state estimation for modular vision-inertial systems. In *2011 IEEE International Conference on Robotics and Automation* (pp. 4531–4537).

Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, *28*(6), 854–874.

Giuseppe Oriolo is an Associate Professor of Automatic Control and Robotics at the Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG) of Sapienza University of Rome where he also coordinates the Robotics Laboratory. He has held adjoint teaching positions at various italian universities, including the Universities of Siena, Cassino, and Roma Tre. He has been Associate Editor and Editor of the IEEE Transactions on Robotics respectively in the years 2001–2005 and 2009–2013.

Antonio Paolillo is a Ph.D. student in System Engineering at DIAG since 2011. In 2010 he was a visiting student at the Örebro University. He received the Master degree from Sapienza University of Rome in 2011. From January to July 2014 he was a visiting scholar at LIRMM, University of Montpellier 2. He is a member of the Robotics Laboratory.



Lorenzo Rosa received the Ph.D. in System Engineering in 2014 from Sapienza University of Rome, where he is a researcher at the Robotics Laboratory. His research focuses on the use of vision for state estimation and control of mobile robots. In 2012–2013 he was visiting scholar at Laboratoire I3S (CNRS), in Sophia Antipolis.

Marilena Vendittelli received the Ph.D. in Systems Engineering in 1997 from Sapienza University of Rome. She held a postdoc position at LAAS-CNRS in Toulouse from 1997 to 1998, funded by Marie Curie Research Training Grants. She is currently Assistant Professor at DIAG and she is a member of the Robotics Laboratory. She has been Associate Editor for the IEEE Transactions on Robotics during the years 2010–2013.