Pose and velocity estimation for soccer robots

Citation for published version (APA):
Kon, J. J., Houtman, W., Kuijpers, W. J. P., & van de Molengraft, M. J. G. (2018). Pose and velocity estimation for soccer robots. *Student Undergraduate Research E-Journal*, *4*. https://doi.org/10.25609/sure.v4.2840

Document license:
CC BY-SA

DOI:
10.25609/sure.v4.2840

Document status and date:
Published: 09/11/2018

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Pose and Velocity Estimation for Soccer Robots

Johan Kon
Eindhoven
University of
Technology
j.j.kon@student.tue.nl

Wouter Houtman
Eindhoven
University of
Technology
w.houtman@tue.nl

Wouter Kuijpers
Eindhoven
University of
Technology
w.j.p.kuijpers@tue.nl

René van de Molengraft
Eindhoven
University of
Technology
m.j.g.v.d.molengraft@tue.nl

ABSTRACT

This paper details the design and real-time implementation of a planar state estimator for soccer robots. A camera system, encoders, gyroscope and accelerometer are combined in a two-stage Kalman filter through a constant acceleration model. Inflating Noise Variance is employed to handle slip and ensure convergence in stationary periods. The approach offers substantial improvement w.r.t. the old pose estimator.

AUTHOR KEYWORDS

localization, pose estimation, Kalman filtering, RoboCup MSL

INTRODUCTION

Navigating robots autonomously in multi-agent, dynamic environments based on certain desires and goals is an important research topic within mobile robotics [1]. To carry out the task of autonomous navigation, localization is a prerequisite. The RoboCup Middle Size League (MSL)[1], in which multiple teams compete in autonomous robot soccer, is an example of an application area in which localization plays a role. All agents are required to play using on-board sensors only. It poses a challenging environment for localization due to high velocities, aggressive acceleration patterns resulting in slip and collisions between agents.

Tech United Eindhoven is among the competing teams in the MSL with their robots, called TURTLEs, of which a system overview is given in [2] and most recent developments in [3]. To localize, the TURTLEs use an on-board camera system, dubbed Omnivision (OV), that directly finds the robots position (x,y) and orientation θ, together called pose, based on the visible field lines [4] at 50 [Hz]. As this sample rate is too low for the low-level velocity control loops and as the OV contains periods of failure, additional rotary encoders that measure the rotation of each of the three omniwheels at 1000 [Hz], are used to interpolate between the OV samples. The encoder increments are kinematically mapped, i.e. without taking slip into consideration, into pose increments, which are summed to obtain the absolute pose. This is called dead-reckoning and is sensitive to drift due to wheel slip. The dead-reckoned pose is constantly compared with the OV pose and reset to the OV pose when the difference exceeds a certain threshold.

Results of a realization of this strategy for the x position of a TURTLE are shown in Figure 1. One can observe that the camera samples, that are available at 50 [Hz], are combined with the encoder odometry to obtain a pose estimate at 1000 [Hz] using dead-reckoning. This reset-based pose estimation strategy results in steps in the pose estimate, has limited noise reduction, is not robust to OV outliers (32 [s]), and is sensitive to drift in periods of OV failure (32 to 33.4 [s]).

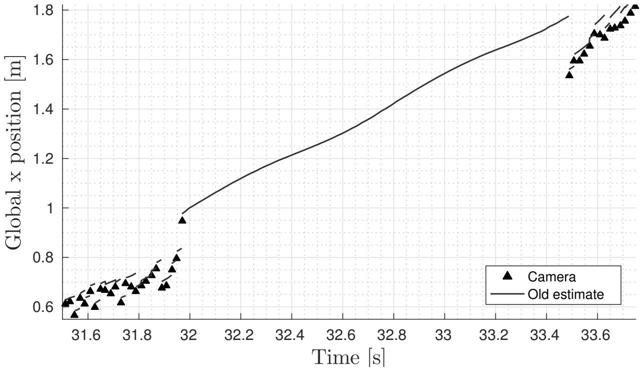

Figure 1: Steps, outlier sensitivity and drift in the current pose estimate

The herefore presented properties manifest themselves in overshooting the setpoint position and mispositioning. To increase the performance of the TURTLEs, a new strategy for pose estimation is required, which

i) does not contain resets,
ii) is robust to camera noise and insensitive to drift and slip,
iii) is suited for real-time implementation at 1000 [Hz].

This paper starts with a study into the possible solution frameworks from literature, followed by a discussion of the available sensors. The chosen framework from literature is then combined with the sensors. The proposed approach is validated by means of real-time implementation on the robots and compared with the old estimator.

LITERATURE STUDY

A literature study revealed a variety of suitable frameworks for the localization problem, including kernel adaptive filters (KAF), moving horizon estimators (MHE), Kalman filter (KF) and its various flavors [5], complementary filters (CF), and particle filters (PF). The techniques were compared with respect to performance and computational complexity. The KF's light-weight, recursive linear algebra is suitable for real-time implementation at 1000 [Hz], as opposed to the computational burden of MHE, KAF and PF. It does however give up performance w.r.t MHE, in the sense that is no longer optimal due to non-valid Gaussian and linearity assumptions, but still outperforms CF due to the in-

[1] http://www.RoboCup.org/

SRC 2018, November 9, 2018, The Netherlands.

clusion of a model. The KF thus offers the best compromise between computational complexity and performance. Combined with the linear behavior of the system at the sample time scale, which ensures stability and limits suboptimality of the KF induced by the non-linear models, the KF is the most suitable framework for the estimation problem at hand. The KF framework allows for inclusion of a model, relating the various state quantities, and integration of multiple sensors, directly or indirectly measuring state quantities. The decision for the KF architecture requires a definition of a state vector, i.e. a vector containing all relevant physical quantities of the estimation problem. For the planar problem at hand, the state consists of the robots pose (x, y, θ) and its first and second time derivative, indicated by a dot and double dot respectively. The state has been split into an **o**rientation and **p**osition state, as given by

$$\mathbf{o} = (\theta, \dot{\theta}, \ddot{\theta})^T, \quad (1)$$

$$\mathbf{p} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, c^I_{\ddot{x}}, c^I_{\ddot{y}})^T, \quad (2)$$

in which $c^I_{\ddot{x}}$ and $c^I_{\ddot{y}}$ are the measurement offsets of the accelerometer in its x and y direction in [m/s^2]. The proposed approach consists of 1) the integration of the already used sensors into the KF framework, 2) the inclusion of extra sensors providing relevant information about the state and 3) the employment of a model for high-frequent noise rejection.

SENSORS

The sensors available for pose estimation, which are already integrated in the software, are the on-board camera system and the rotary encoders on the wheels. In addition, the inertial measurement unit (IMU) Xsens MTi-3 AHRS, which contains a magnetometer, gyroscope and accelerometer, has been interfaced in the software, such that its gyroscope and accelerometer could be used. The magnetometer has been left out, as the OV and gyroscope proved to be sufficient for a high quality orientation estimate.

Omnivision

The OV directly returns the robot pose in the inertial frame by means of a field line fitting algorithm and guarantees observability of the system. However, it runs at 50 [Hz], which is too slow for the low-level control loops and occasionally has periods of 2-3 [s] in which it is unable to find a pose, temporarily removing the observability of the system. Furthermore, the OV pose is delayed by 40 [ms] due to processing of the image.

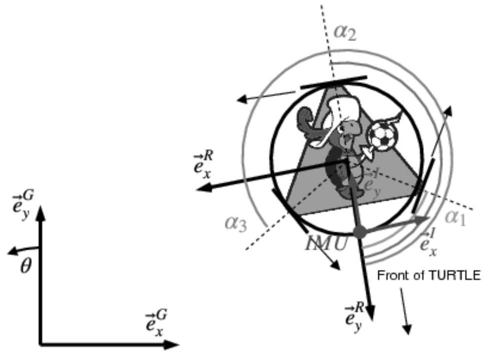

Figure 2: Frame conventions for the TURTLE

The noise characteristics of the OV pose are approximated by three independent, white, zero-centered Gaussian random variables with position varying variance, as the TURTLE's position influences the amount of line points in the image on which the fit is made: the more line points, the more accurate and precise the estimated pose.

Encoders

The encoders are used to measure wheel rotation at 1000 [Hz], but can be converted to wheel speeds by numerically differentiating them, as the encoder signals contain no noise and their resolution is sufficiently high to neglect the quantized nature. The robot velocities in the inertial frame can then be mapped onto each of the wheel speeds $\dot{\phi}_i$ by a coordinate transformation into local robot frame followed by a kinematic mapping onto the wheels, according to

$$\dot{\phi}_i = \frac{1}{r_{w,i}}\Big(- \sin(\theta + \alpha_i + \frac{\pi}{2})\dot{x} + \cos(\theta + \alpha_i + \frac{\pi}{2})\dot{y} + R_i\dot{\theta}\Big), \quad (3)$$

in which $r_{w,i}$, α_i and R_i are the wheel radius, wheel axle offset w.r.t local y-axis and perpendicular distance from wheel to platform center respectively, as shown in Figure 2. This kinematic mapping degrades in cases of extreme slip or skid, in which the wheels lose traction and the wheel speeds are not representative for the platform speeds anymore. Therefore, the robot velocities should not be based on the wheel velocities in the situation where slip is present.

Gyroscope

The gyroscope returns the rate of turn $\dot{\theta}$ of the robot at 500 [Hz] with some measurement offset, as each point on the rigid robot body has the same rotational velocity. The gyroscope drift is negligible, so the static offset can be removed by calibration. The noise is modeled by a white, Gaussian variable with a constant variance.

Accelerometer

The accelerometer measures acceleration in its 3-axis Cartesian body frame, as defined in Figure 2. It is mounted in the front, outside the center of the robot on a piece of tape. This way of mounting causes contributions of tangential and centripetal accelerations, due to angular accelerations and velocities of the robot, in respective x and y axis of the accelerometer on top of the rotated global linear accelerations. The accelerations (a^I_x, a^I_y) as measured in the IMU frame can thus be expressed as

$$\begin{bmatrix} a^I_x \\ a^I_y \end{bmatrix} = \begin{bmatrix} -R_I\ddot{\theta} \\ R_I\dot{\theta}^2 \end{bmatrix} + \begin{bmatrix} -\cos(\theta) & -\sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} c_{\ddot{x}^I} \\ c_{\ddot{y}^I} \end{bmatrix}, \quad (4)$$

in which R_I is the radius from the robot center to the IMU. The measured accelerations are further contaminated by vibrations from the omniwheels due to non-smooth transitions between the rollers. Spectral analysis shows that these vibrations are bounded between 10 and 30 [Hz] in match-like driving conditions, while the actual robot accelerations are situated between 0 and 5 [Hz]. The power of the omniwheel vibrations has the same order of magnitude as the actual accelerations and therefore heavily reduces the usefulness of the raw measurement. Lastly, the tape causes hysteresis in the positioning of the IMU, causing its rotation to change by up to 1-2 [deg] around each axis in both directions. This change in orientation causes a change in projection of the gravity vector, which is measurable as a changing offset in the accelerometer measurement which cannot be calibrated.

FILTER DESIGN

The available sensors can now be combined in the chosen KF framework through a model, which relates various state quantities to each other and allows for noise rejection due to its low gain at high frequencies. The full filter is a two-stage approach to the estimation problem, consisting of a KF to estimate **o** based on the OV and gyroscope, followed by a KF to estimate **p** given **o** based on the OV, encoders and accelerometer and their respective measurement models, combined with a preprocessor to i) time align the sensors by a simple delay, ii) convert encoder pulses to wheel speeds, iii) filter the omniwheel vibrations with a linear phase finite impulse response (FIR) low-pass filter (LPF), and iv) detect new samples to deal with the different sample rates of the sensors. This 2KF approach is illustrated in Figure 3. The motivation for a two-stage approach is threefold:

i. The accelerometer and encoders can be utilized more effectively in the second stage, due to the more accurate coordinate transformation from local to global robot speed and the ability to better compensate for the tangential and centripetal accelerations of (4).
ii. The non-linear measurement models of the accelerometer and encoders simplify to linear ones given the orientation.
iii. Splitting the state decreases the computational load due to smaller dimensional matrices.

The process model employed in both stages of the filter is a constant acceleration model, in which the state at the next time step $k + 1$ is the discretely integrated state of time step k, assuming that the acceleration is constant, as given by

$$\mathbf{o}_{k+1} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \mathbf{o}_k, \qquad (5)$$

in which T_s is the sample time. Although the system does not have memory at acceleration level, the closed control loops combined with continuous errors ensure that the force and thus the acceleration can be approximated as a constant on the sample time scale. In practice, it proved to be the best prediction for the state transition, outperforming both a dynamic wheel model based on slip curves and a simplified model in which the wheel was treated as transmission ratio.

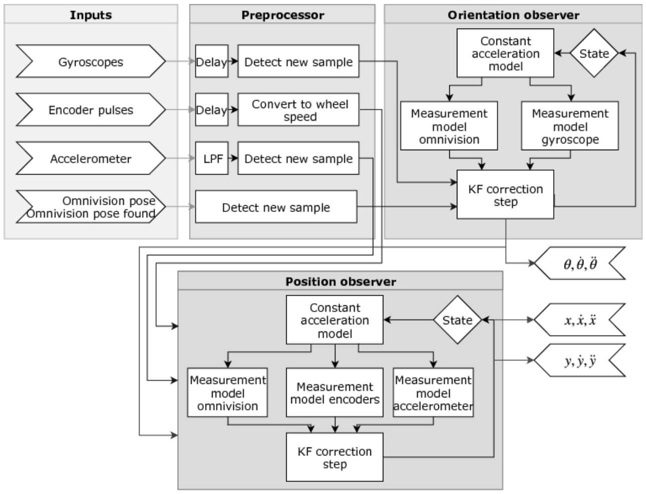

Figure 3: Two-stage Kalman filter architecture

The concept of Inflating Noise Variance (InNoVa) [6] is employed in the position estimator of Figure 3 in the process model and accelerometer and encoder measurement model:

- The covariance of the process model is scaled with an approximation to the deviation from the constant acceleration assumption and a velocity contribution to account for unmodeled friction. No actuation and no velocity therefore means low prediction covariance, allowing for convergence in stationary periods.
- The covariance of the encoders is scaled with the integrated actuation in a finite window in the past. Periods of extreme actuations strongly associated with slip and skid therefore ensure high covariances such that the encoders are not used as robot velocities.
- The covariance of the accelerometer is scaled with the with the statistical average of the first order propagated variances of **o** to prevent taking **o** as absolute truth and overcompensating the linear accelerations. The covariance of the accelerometer offsets is scaled with the rate of turn around the x and y axis to dynamically readjust the offset and deal with the varying IMU orientation.

RESULTS

After implementation in the Tech United software, the designed filter has been tested with respect to the criteria specified in the problem statement. In this analysis, the OV will be used as ground truth both due to its superior performance regarding noise, bias and sample rate w.r.t. other available systems and the lack of time to fully set up another ground truth measurement system. The analysis is conducted on a set of results gathered during match-like driving conditions including slip and camera failure.

Orientation

Figure 4 shows the old estimate, 2KF estimate and OV orientation measurements during slip. The inter OV sample drift and resulting resets due to slip present in the old estimate can clearly be seen around 49.5 and 51.5 [s]. The 2KF does not suffer from slip or drift and interpolates and smooths the OV data due to the inclusion of the gyroscope, resulting in an estimate that is devoid of resets and robust to OV noise.

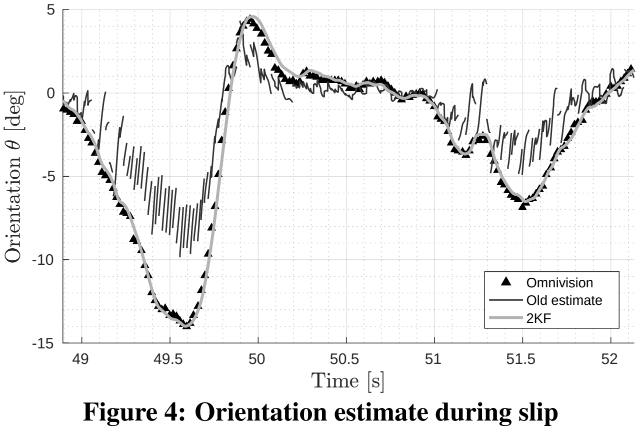

Figure 4: Orientation estimate during slip

Figure 5 shows both orientation estimates during two successive periods of OV failure (75-76 [s] and 77-79 [s]). The gyroscope based dead-reckoned orientation estimate of the

2KF is able to consistently predict the first new OV sample within 0.5 [deg] accuracy (76.2 and 78.9 [s]). The old estimate is completely lost, introducing resets of up to 12 [deg] and even bigger errors, possibly up to 18 [deg].

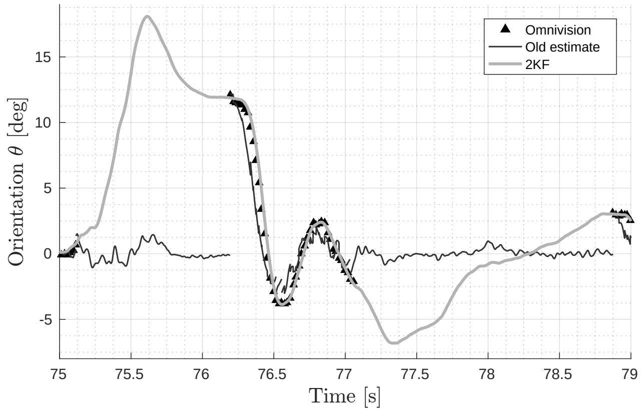

Figure 5: Orientation estimate during OV failure

Position

Figure 6 illustrates both x position estimates in a period of wheel slip. The slip can clearly be seen from 49 [s] onwards: the old estimate, which uses the kinematically mapped wheel speeds, starts to drift between OV samples into the direction the robot will move in 1 [s] later. The deviation from the OV is caused by wheel spin due to excessive actuation and slip needed to generate a tractive force to accelerate the robot, i.e. the wheels first start turning into a certain direction before the platform moves along. The 2KF estimate properly smooths the OV data and is unaffected by the wheel spin due to the InNoVa employed for the encoder measurements and the inclusion of the accelerometer: the contradiction between the position and acceleration information from the OV and accelerometer and the velocity information from the encoders combined with the inflated variation of the encoders, causes the 2KF to temporarily ignore the wheel speeds.

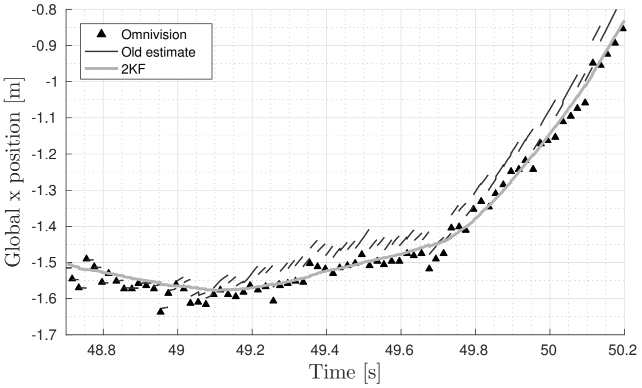

Figure 6: Position estimate during slip

Table 1 shows the absolute difference between the two estimates and the new OV sample after a period of dead-reckoning. Although the 2KF can still introduce significant errors of up to 19 [cm], its performance is better than or comparable to the old estimate due to the inclusion of the accelerometer and the two-stage architecture which allows for a more accurate mapping from wheel to global speed. The quality of dead-reckoning is largely dependent on the quality of the accelerometer offset estimation, which determines the extent to which the estimate is able to drift in combination with the slip during OV failure.

Table 1: Dead-reckoning errors in [m]

#	$\lvert e_x\rvert$ 2KF	$\lvert e_x\rvert$ old	$\Delta\lvert e_x\rvert$	$\lvert e_y\rvert$ 2KF	$\lvert e_y\rvert$ old	$\Delta\lvert e_y\rvert$
1	0.013	0.138	-0.125	0.013	0.193	-0.180
2	0.143	0.420	-0.277	0.052	0.184	-0.132
3	0.186	0.443	-0.257	0.121	0.256	-0.135
4	0.035	0.232	-0.197	0.088	0.064	+0.024

The results for the velocity exhibit similar characteristics w.r.t noise reduction and robustness to slip.

CONCLUSION

This paper presents the design and real-time implementation of a two-stage Kalman filter approach to the planar pose estimation problem to increase the quality of localization on the TURTLEs. The adaption of the KF framework in combination with the integration of extra sensors and the inclusion of a constant acceleration model ensure that the estimator is robust to noise of all employed sensors. The approach is less sensitive to slip due to InNoVa employed for the encoders and therefore eliminates inter OV sample drift and, combined with the extra sensors, limits drift and improves performance during dead-reckoning. The performance during dead-reckoning is however still influenced by the IMU mounting, introducing the need for the design of a proper mount. In addition, the time alignment strategy introduces a delayed pose estimate, possibly limiting the bandwidth of the low-level controllers, calling for more advanced techniques to deal with time delayed measurements. These limitations are only relevant due to the aggressive nature of the RoboCup environment, making the approach also applicable to use cases with less slip and lower velocities.

ROLE OF THE STUDENT

The design of the filter, integration of the sensor and implementation in the robot software were carried out by Johan Kon for his Bachelor Thesis as part of his Bachelor ME at TU/e. The work was supervised by Wouter Houtman, Wouter Kuijpers and René van de Molengraft.

REFERENCES

1. *Autonomous Navigation in Dynamic Environments*. Springer-Verlag Berlin Heidelberg, 2007.
2. C.A. López Martínez, F. Schoenmakers et al. Tech United Eindhoven Team Description Paper 2014 MSL
3. F. Schoenmakers, K. Meessen et al. Tech United Eindhoven Team Description Paper 2018 MSL
4. D. Bruijnen, W. Aangenent, J. van Helvoort, and R. van de Molengraft. From Vision to Realtime Motion Control for the RoboCup Domain. In *2007 IEEE International Conference on Control Applications*, pages 545–550. IEEE, oct 2007.
5. G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, 1995.
6. J. Zhang, G. Welch, G. Bishop, and Z. Huang. A Two-Stage Kalman Filter Approach for Robust and Real-Time Power System State Estimation. *IEEE Transactions on Sustainable Energy*, 5(2):629–636, apr 2014.