Proprioceptive Navigation, Slip Estimation and Slip Control for Autonomous Wheeled Mobile Robots

Martin Seyr
Institute of Mechanics and Mechatronics
Vienna University of Technology
martin.seyr@tuwien.ac.at

Stefan Jakubek
Institute of Mechanics and Mechatronics
Vienna University of Technology
stefan.jakubek@tuwien.ac.at

***Abstract*— For a two-wheeled differentially driven mobile robot a navigation and slip control algorithm is developed. The presented concept for purely proprioceptive navigation combines state estimation via extended Kalman filter from inertial sensor data (i.e. gyro and acceleration sensors) and odometric measurements (i.e. wheel angular encoders). The advantages of both types of sensors are exploited by selective mixing. Tangential slip detection and side-slip angle measurement enable slip control by transiently overriding a pre-planned trajectory. Experimental results demonstrating the performance of the proposed system are presented.**

***Keywords*—mobile robot navigation, sensor fusion, slip detection, side-slip angle estimation**

I. Introduction

Autonomous mobile robot control generally raises three problems: control, navigation and motion planning. The present work concentrates on planar navigation, i.e. estimation of the current position and orientation, and slip control, without any absolute position measurement such as GPS, radio beacons or landmark detection. Only data from acceleration sensors, a gyro-sensor (inertial sensors) and wheel encoders (odometric sensors) are fused. This is known as proprioceptive navigation or dead reckoning as opposed to exteroceptive navigation.
Naturally, the position error is subject to unbounded accumulation. Nevertheless, improved proprioceptive navigation systems help increase the allowable travel distance between absolute position updates, [1].
The standard method for sensor fusion is extended Kalman filtering (EKF), e.g. [2], [3]. An EKF yields an optimal estimate in the sense that the statistical variance of the estimated states is minimised. However, biased measurement produces erroneous estimates. Typical sources of measurement bias are inevitable sensor drift for inertial sensors and inaccuracies of geometric parameters and wheel slip for odometric measurement.

- To reduce the effect of sensor bias, error models predicting the bias have been developed, [4]; or else it can be estimated and re-calibrated, [5].
- To calibrate geometric parameters such as wheel diameter and wheelbase, which are vital to odometric navigation, a statistical method has been developed in [6], [7].
- To overcome the effects of wheel slip or bumps (causing discontinuous ground contact), which introduces a non-systematic error into odometry, inertial data is used only transiently during periods where odometric measurement is likely to be unreliable. Aside from these periods, odometric measurement is used exclusively, because it is not affected by drift and has little bias when properly calibrated, [1], [8].

In the presented approach both gyro and acceleration sensors are transiently substituted for wheel encoder data when necessary, thus enabling reliable side-slip angle estimation and tangential slip detection and ultimately allowing for slip control.
The proposed navigation system is used in conjunction with the predictive trajectory tracking algorithm from [9]. Experimental results show that the combined navigation and control system enables appropriate tracking of highly dynamic trajectories.

II. Hardware

The autonomous robot Tinyphoon (www.tinyphoon.com), [10], Fig. 1, has two wheels with rubber tires and two felt shoes, one at the front and one at the rear to stabilise it around the pitch axis. It fits into a cuboid with a 75mm square footprint. The two wheels are supported by ball bearings and powered by two individual DC-motors.
Exhibiting a power-mass-ratio of 22.5W/kg, the robot is capable of accelerating far beyond the slip boundary. Its maximum velocity is approximately 4m/s.

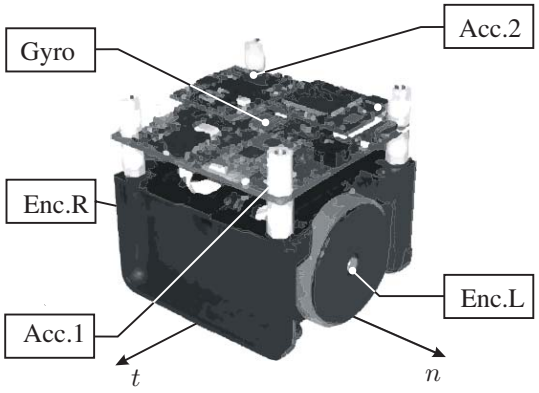

Fig. 1. Autonomous mini-robot Tinyphoon, http://www.tinyphoon.com

The robot is equipped with two single-chip two-axis acceleration sensors measuring tangential and lateral acceleration,

a single-chip gyro sensor and wheel encoders for each side. Therefore, seven measurement channels sampled at the same sampling rate of 2ms are available.
An Analog Devices Blackfin DSP running at 600MHz provides sufficient calculation performance to execute both the predictive control algorithm and the navigation system.

III. Kinematics

The kinematics of the unicycle-type mobile robot under consideration of side-slip, Fig. 2, are given by

$$\begin{aligned} \dot{x} &= v\cos(\varphi+\alpha) = v_\mathrm{t}\cos\varphi - v_\mathrm{n}\sin\varphi \\ \dot{y} &= v\sin(\varphi+\alpha) = v_\mathrm{t}\sin\varphi + v_\mathrm{n}\cos\varphi \\ \dot{\varphi} &= \omega, \end{aligned} \tag{1}$$

where x and y denote the inertial coordinates of the robot's center of gravity, φ denotes its the inertial attitude angle, α is the side-slip angle, v the effective velocity with components v_t and v_n and ω denotes the yaw rate.

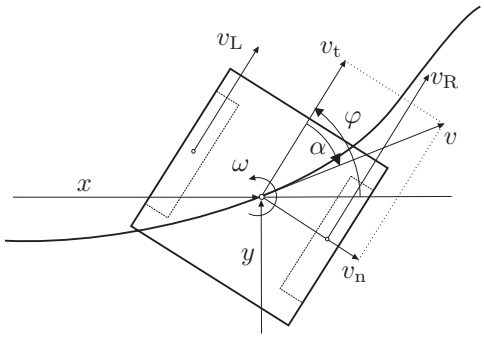

Fig. 2. Kinematics of a two-wheeled mobile robot under consideration of side-slip

To calculate the robot's position and orientation x, y and φ, estimation of v_t, v_n and ω and subsequent integration using (1) is necessary.

IV. Combined odometric and inertial navigation

As mentioned in the introduction, the general idea is to use data from inertial sensors rather than data from odometric sensors only during periods where odometry is likely to be unreliable, i.e. when tangential or side-slip occur.
The task is therefore to detect these situations. As functions of appropriate criteria, mixing ratios are calculated, by which a combination ranging from purely inertial estimation to purely odometric estimation is defined.
In Fig. 3 a complete block diagram of the navigation algorithm is depicted.
The system operates at two different sampling times, $T_\mathrm{N} = 2$ms primarily for navigation (integer sampling instants in the following indicated by k) and $T_\mathrm{C} = 20$ms for control (sampling instants indicated by K), $T_\mathrm{C} > T_\mathrm{N}$, $T_\mathrm{C}/T_\mathrm{N} \in \mathbb{N}^+$. Some parts of the navigation algorithm are also executed at T_C, Fig. 3.

A. Inertial Navigation by Extended Kalman Filter (EKF)

The two-stage (prediction-update) EKF processes the following sensor data at a sampling time of T_N:

- *Accelerometers:* Two channels in tangential direction, two channels in lateral direction. The sensors are subject to unknown drift which prohibits purely inertial navigation. Another problem is strong mechanical excitation caused by commotions.
- *Gyro sensor:* One channel measuring the yaw rate. Compared to the accelerometers, the drift is marginal, thus providing a very reliable measurement.

The state prediction using a nonlinear discrete model is given by

$$\hat{\boldsymbol{x}}_{k|k-1} = \begin{bmatrix} \hat{a}_{\mathrm{t},k-1} + \hat{v}_{\mathrm{n},k-1}\hat{\omega}_{k-1} \\ \hat{a}_{\mathrm{n},k-1} - \hat{v}_{\mathrm{t},k-1}\hat{\omega}_{k-1} \\ 0 \\ 0 \\ 0 \end{bmatrix} T_\mathrm{N} + \underbrace{\begin{bmatrix} \hat{v}_{\mathrm{t},k-1} \\ \hat{v}_{\mathrm{n},k-1} \\ \hat{\omega}_{k-1} \\ \hat{a}_{\mathrm{t},k-1} \\ \hat{a}_{\mathrm{n},k-1} \end{bmatrix}}_{\hat{\boldsymbol{x}}_{k-1}}, \tag{2}$$

where v_t, v_n, ω and the absolute accelerations a_t and a_n are chosen as the states. The index $k|k-1$ can be spelled out as 'prediction of states at instant k, based on data of instant $k-1$'. The measurement vector is written as

$$\boldsymbol{y}_k = \begin{bmatrix} \omega_k \\ (a_{\mathrm{t}1,k} + a_{\mathrm{t}2,k})/2 \\ (a_{\mathrm{n}1,k} + a_{\mathrm{n}2,k})/2 \end{bmatrix}, \tag{3}$$

where the arithmetic mean of the two respective acceleration measurements is used to reduce noise variance, thus effectively yielding three measurements instead of five.
The prediction of the measurements is then given as

$$\hat{\boldsymbol{y}}_{k|k-1} = \begin{bmatrix} \hat{\omega}_{k|k-1} \\ \hat{a}_{\mathrm{t},k|k-1} \\ \hat{a}_{\mathrm{n},k|k-1} \end{bmatrix}. \tag{4}$$

The Jacobian of the nonlinear discrete state space representation (2) is calculated as

$$\boldsymbol{G}_{k-1|k-1} = \tag{5}$$
$$= \begin{bmatrix} 1 & \hat{\omega}_{k-1}T_\mathrm{N} & \hat{v}_{\mathrm{n},k-1}T_\mathrm{N} & T_\mathrm{N} & 0 \\ -\hat{\omega}_{k-1}T_\mathrm{N} & 1 & -\hat{v}_{\mathrm{t},k-1}T_\mathrm{N} & 0 & T_\mathrm{N} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

whereas the measurement prediction (4) is linear in the states and therefore a constant measurement matrix $\boldsymbol{C} \in \mathbb{R}^{3\times 5}$ is obtained. The covariance of the prediction $\hat{\boldsymbol{x}}_{k|k-1}$ is given by

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{G}_{k-1|k-1}\boldsymbol{P}_{k-1}\boldsymbol{G}_{k-1|k-1}^T + \boldsymbol{Q}, \tag{6}$$

where $\boldsymbol{Q} \in \mathbb{R}^{5\times 5}$ denotes the (constant) covariance matrix of the states, whose entries are tuned to be as small as possible

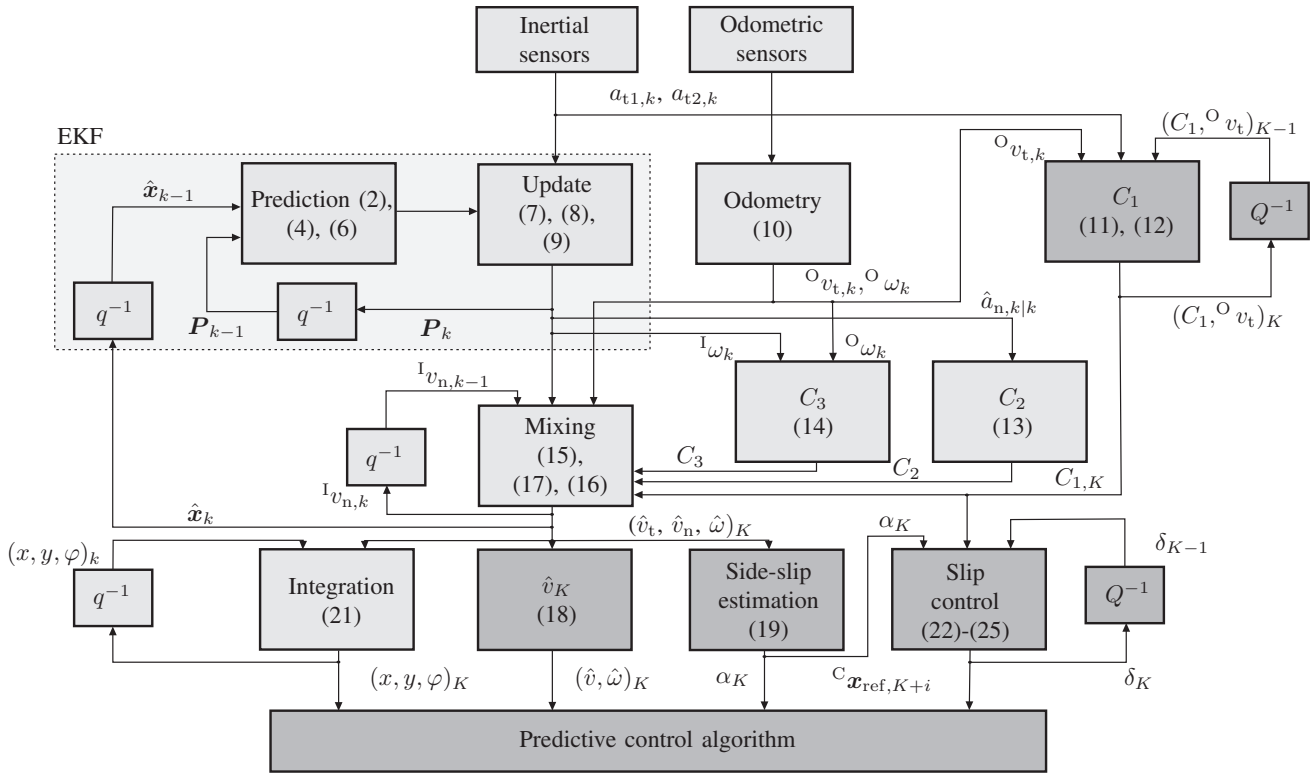

Fig. 3. Block diagram, light grey: T_N, dark grey: T_C

but large enough to ensure convergence.
The Kalman gain matrix $\boldsymbol{L}_K \in \mathbb{R}^{5\times3}$ is obtained by minimization of the trace of the state covariance matrix $\boldsymbol{P}_k \in \mathbb{R}^{5\times5}$. This calculation is performed at T_C to save calculation time, without recognisable deterioration of accuracy.

$$\boldsymbol{L}_K = \boldsymbol{P}_{k|k-1}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}), \tag{7}$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the (constant) covariance of the measurements, obtained from sensor manufacturer's data sheets or from analysis of the sensor signals in steady state.
Using the Kalman gain matrix $\boldsymbol{L}_K$ and the difference between predicted and actual measurement, the states are updated to obtain the final state estimate,

$$\boldsymbol{x}_k = \boldsymbol{x}_{k|k-1} + \boldsymbol{L}_K(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}). \tag{8}$$

As a last step, the covariance of the updated state estimate

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{L}_K\boldsymbol{C})\boldsymbol{P}_{k|k-1}(\boldsymbol{I} - \boldsymbol{L}_K\boldsymbol{C})^T + \boldsymbol{L}_K\boldsymbol{R}\boldsymbol{L}_K^T \tag{9}$$

is calculated to be used in the next sampling instant.

B. Odometry

From the encoders, which are free of bias due to their principle of operation, the wheel velocities v_R, v_L are obtained at relatively low noise. The only source of noise is quantisation due to the finite number of encoder impulses per wheel revolution.
Aside from non-systematic errors due to wheel slip, odometric data is generally of very high quality.
To obtain tangential velocity v_t and yaw rate ω, a purely deterministic geometric transformation is applied,

$$\begin{aligned} v_{\mathrm{t},k} &= (v_{\mathrm{R},k} + v_{\mathrm{L},k})/2 \\ \omega_k &= (v_{\mathrm{R},k} - v_{\mathrm{L},k})/b, \end{aligned} \tag{10}$$

where the geometric parameters wheelbase b and the wheel radius used to transform the wheel angular velocity into the wheel velocities have been calibrated offline using results from [6], [7].
Naturally, from odometric measurement no information about the lateral (side-slip-) velocity v_n can be obtained.

C. Tangential slip detection

A batch of $T_\mathrm{C}/T_\mathrm{N} = 10$ samples of sensor data is processed. The average tangential velocity difference of the current batch after an initial reset is calculated by discrete integration of the arithmetic mean of the tangential accelerations (upper left index O stands for odometric estimate)

$$\overline{\Delta v}_K = \frac{T_\mathrm{N}}{T_\mathrm{C}} \sum_{k=1}^{T_\mathrm{C}/T_\mathrm{N}} \left({}^{O}v_{\mathrm{t},k} - {}^{O}v_{\mathrm{t},K-1} - (a_{\mathrm{t}1,k} + a_{\mathrm{t}2,k})\frac{T_\mathrm{N}}{2} \right), \tag{11}$$

which effectively means that a trend is detected, but not a stationary slip. For the latter, the reliability of the acceleration sensors does not suffice.

D. Mixing

The final estimate is generated by linear mixing between odometric (upper left index O) and inertial (upper left index I) estimation.
As the criterium for mixing of the tangential velocity v_t the tangential slip is used. The mixing parameter is calculated every T_C using

$$C_{1,K} = p\, C_{1,K-1} + (1-p) \underbrace{\frac{1}{1+\exp(a_1(|\overline{\Delta v}_K| - b_1))}}_{\tilde{C}_{1,K}}, \quad (12)$$

a double sided logsig-function as depicted in Fig. 4. Additionally, C_1 is dynamically relaxed following discrete PT1-behavior.

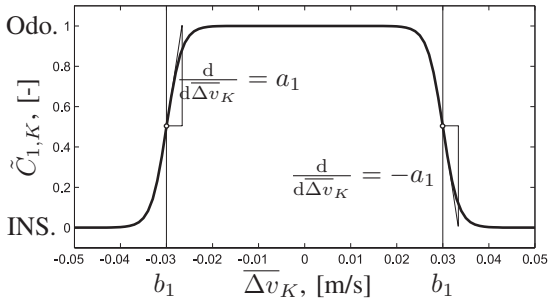

Fig. 4. Double-sided logsig-function as switch between odometric (Odo.) and inertial (INS.) navigation

The tuning parameters are obtained as follows:

- a_1: steepness, chosen as steep as possible, but as flat as necessary to avoid limit cycle behavior.
- b_1: threshold, chosen as small as possible, but large enough to exclude faulty slip detection due to acceleration sensor bias.
- p: discrete PT1-pole, chosen as close to zero (fast) as possible, but large enough to avoid limit cycle behavior.

For the lateral velocity v_n, the absolute side-acceleration (calculated by the EKF) is used as a criterium, its mixing parameter computed every T_N by

$$C_{2,k} = \frac{1}{1+\exp(a_2(|\hat{a}_{\mathrm{n},k|k}| - b_2))}, \quad (13)$$

where the parameters a_2 and b_2 are obtained much in the same way as for the tangential velocity.
The criterium for yaw-rate mixing is simply the absolute difference between odometric and inertial measurement, [7], [1], when the difference becomes too large, odometric measurement is assumed to be unreliable, therefore the mixing parameter is obtained every T_N by

$$C_{3,k} = \frac{1}{1+\exp(a_3(|{}^\mathrm{O}\omega_k - {}^\mathrm{I}\omega_k| - b_3))} \quad (14)$$

with parameters a_3 and b_3 as above.
The linear mixing laws for tangential velocity v_t,

$$\hat{v}_{\mathrm{t},k} = C_{1,K}\,{}^\mathrm{O}v_{\mathrm{t},k} + (1-C_{1,K})^\mathrm{I}v_{\mathrm{t},k} \quad (15)$$

and yaw rate ω,

$$\hat{\omega}_k = C_{3,k}\,{}^\mathrm{O}\omega_k + (1-C_{3,k})^\mathrm{I}\omega_k, \quad (16)$$

are straightforward, whereas the lateral velocity v_n, calculated by

$$\hat{v}_{\mathrm{n},k} = C_{2,k}\hat{v}_{\mathrm{n},k-1}q_2 + (1-C_{2,k})^\mathrm{I}v_{\mathrm{n},k}, \quad (17)$$

is taken fully from EKF when the absolute lateral acceleration is beyond the threshold b_2 but geometrically relaxed towards zero when the lateral acceleration is within the threshold. This is necessary, because from odometry no information about lateral velocity can be obtained, i.e. ${}^\mathrm{O}v_{\mathrm{n},k}$ does not exist. Without relaxation, however, frequent crossing of the threshold due to inevitably noisy lateral acceleration would result in frequent reset of the lateral velocity to zero.
The factor q_2, $0 < q_2 < 1$ of geometric progression is chosen as small as possible but large enough to ensure consistent velocity measurement.
The final estimate is fed back to the EKF, which means that the EKF does not work consistently in a strict sense, since its output is overwritten after every sampling instant, Fig. 3.

E. Calculation of the side-slip angle

The resulting track speed, also needed in the control algorithm [9], is calculated by

$$\hat{v}_K = \sqrt{\hat{v}_{\mathrm{t},K}^2 + \hat{v}_{\mathrm{n},K}^2}\,\mathrm{sign}(\hat{v}_{\mathrm{t},K}), \quad (18)$$

implicitly assuming that the side-slip angle does not exceed $[-\pi/2; \pi/2]$ (a reasonable assumption) by using the sign-function. The side-slip angle is calculated from the velocity components,

$$\alpha_K = \mathrm{atan}\left(\frac{\hat{v}_{\mathrm{n},K}}{\hat{v}_{\mathrm{t},K}}\right) f(\hat{v}_K, \hat{\omega}_K), \quad (19)$$

and multiplied by a linear validity function

$$f(\hat{v}_K, \hat{\omega}_K) = \begin{cases} 0 & : & & |\hat{v}_K\hat{\omega}_K| & < & c_\alpha \\ \frac{|\hat{v}_K\hat{\omega}_K| - c_\alpha}{d_\alpha - c_\alpha} & : & c_\alpha & < |\hat{v}_K\hat{\omega}_K| & < & d_\alpha \\ 1 & : & d_\alpha & < |\hat{v}_K\hat{\omega}_K| & & \end{cases} \quad (20)$$

enforcing the side-slip angle to vanish for low track speed or low curvature, where otherwise no meaningful results can be obtained due to numerical reasons, and the side-slip can reasonably be assumed to be very close to zero.

F. Deterministic integration of positions

Finally, the positions and orientations are calculated by deterministic integration using a simple-forward-Euler-discretised representation of (1),

$$\begin{aligned} x_{k+1} &= x_k + (v_{\mathrm{t},k}\cos\varphi_k - v_{\mathrm{n},k}\sin\varphi_k)T_\mathrm{N} \\ y_{k+1} &= y_k + (v_{\mathrm{t},k}\sin\varphi_k + v_{\mathrm{n},k}\cos\varphi_k)T_\mathrm{N} \\ \varphi_{k+1} &= \varphi_k + \omega_k T_\mathrm{N}. \end{aligned} \quad (21)$$

V. SLIP CONTROL

The reference input to the predictive control algorithm (executed at T_C) [9] consists of future reference positions and orientation up to a prediction horizon. These data are generated by a trajectory planning algorithm (upper left index T). A *trajectory* implicitly contains velocity information as opposed to a *path*, which is purely geometric.

In the case of excessive slip, the slip control can now transiently override the trajectory given by the planning algorithm by interpolating the original trajectory points with a reduced step size (nominal 1), thus remaining on the same geometric path, but with a certain delay. When no tangential or lateral slip occurs, the step size is increased beyond 1 to re-match the original trajectory.

The step size δ_K, also based on the previous value δ_{K-1} to ensure continuity, is computed every T_C by

$$\delta_K = \min\Bigg(\max\Bigg(\delta_{K-1}\Bigg(\delta_\mathrm{min} + \min\Bigg(C_{1,K}, \frac{1}{1+\exp(a_4\,(|\alpha_K| - b_4))}\Bigg)(q_\delta - \delta_\mathrm{min})\Bigg), \delta_\mathrm{min}\Bigg), \delta_{\mathrm{max},K}\Bigg), \tag{22}$$

where the maximum step size $\delta_{\mathrm{max},K}$ is given by

$$\delta_{\mathrm{max},K} = \begin{cases} \delta_\mathrm{max}(>1) & : & & K-\Pi_K & > & c_\delta \\ 1 & : & c_\delta & > & K-\Pi_K & > & d_\delta \\ \delta_\mathrm{rem}(<1) & : & d_\delta & > & K-\Pi_K & & \end{cases} \tag{23}$$

The position on the preplanned trajectory is then given as

$$\Pi_{K+1} = \Pi_K + \delta_K, \tag{24}$$

whereas the nominal integer position on the preplanned trajectory naturally equals K.

A number of parameters determine the behavior:

- $\delta_\mathrm{min} < 1$: Minimum step size, determines the reduction of velocity in case of slip,
- $\delta_\mathrm{max} > 1$: Maximum step size, determines how quickly the original trajectory is recovered,
- q_δ: factor of geometric growth of the step size after reduction, if chosen too large, limit cycle behavior results,
- a_4, b_4: steepness and threshold for side-slip angle,
- $\delta_\mathrm{rem} < 1$: Step size to match reference trajectory after overshoot,
- c_δ, d_δ: Upper/lower threshold for re-matching.

The reference positions and orientation finally fed to the predictive control algorithm are obtained by interpolation using

$$\begin{aligned} {}^\mathrm{C}\boldsymbol{x}_{\mathrm{ref},K+i} = {}&{}^\mathrm{T}\boldsymbol{x}_\mathrm{ref}\big(\mathrm{floor}(\Pi_K + i\delta_K)\big) + \\ &\big(\Pi_K + i\delta_K - \mathrm{floor}(\Pi_K + i\delta_K)\big) \\ &\Big({}^\mathrm{T}\boldsymbol{x}_\mathrm{ref}\big(\mathrm{ceil}(\Pi_K + i\delta_K)\big) - {}^\mathrm{T}\boldsymbol{x}_\mathrm{ref}\big(\mathrm{floor}(\Pi_K + i\delta_K)\big)\Big). \end{aligned} \tag{25}$$

Thus the progression of velocities associated with the reference trajectory is expanded or compressed according to δ with equal area, which means that both absolute velocity and velocity gradients (acceleration or deceleration) are reduced.

Fig. 5 gives an example of how the slip control algorithm reduces the step size in a curve, i.e. follows the geometric path with a certain delay, and subsequently catches up with the original trajectory as soon as the curvature decreases, i.e. the side-slip decays. In the picture, associated trajectory points are connected with an oval.

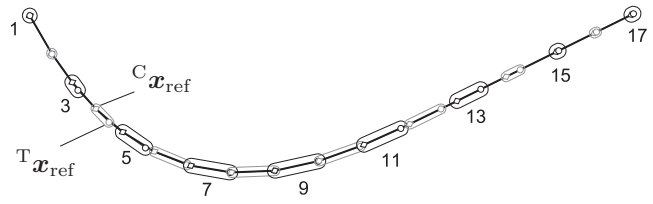

Fig. 5. Example for the variation of the step size during detected side-slip or tangential slip and corresponding interpolation of the pre-planned trajectory

VI. EXPERIMENTAL RESULTS

To demonstrate the performance of the navigation and slip control algorithm, two different trajectories to be tracked have been generated:

- A corner with linear acceleration and deceleration and a cosine-shaped curvature profile, thus producing large side-accelerations.
- A square with sinusoidal velocity and yaw rate profiles, demanding high translative accelerations and decelerations.

The evaluation of the navigation algorithm was performed using an independent offline video system, enabling accurate position and orientation measurement at a framerate of about 25fps and comparing its data offline (i.e. after the experiment) with logged navigation data obtained via an RF-module from the autonomous robot.

In Fig. 6 the effect of the slip control algorithm is obvious. Due to high side-acceleration, a side-slip angle builds up, which is estimated with high accuracy. The slip control algorithm reduces the step size. Later, when circumstances are 'safe' in regard to slip, the step size is increased to catch up with the original trajectory.

In Fig. 7 the action of the slip control algorithm for tangential wheel slip is demonstrated. In situations, where tangential slip is detected (indicated by the black circles), the step size is temporarily reduced. In these situations, also a temporary divergence of the velocity estimate can be observed (indicated by the light gray areas), which is a consequence of the finite reaction time of switching between odometric and inertial estimation.

Finally, in Fig. 8 the position estimation for corner and square is visualised.

VII. CONCLUSION

In this paper a navigation, slip estimation and slip control algorithm for an autonomous mobile robot, entirely based on

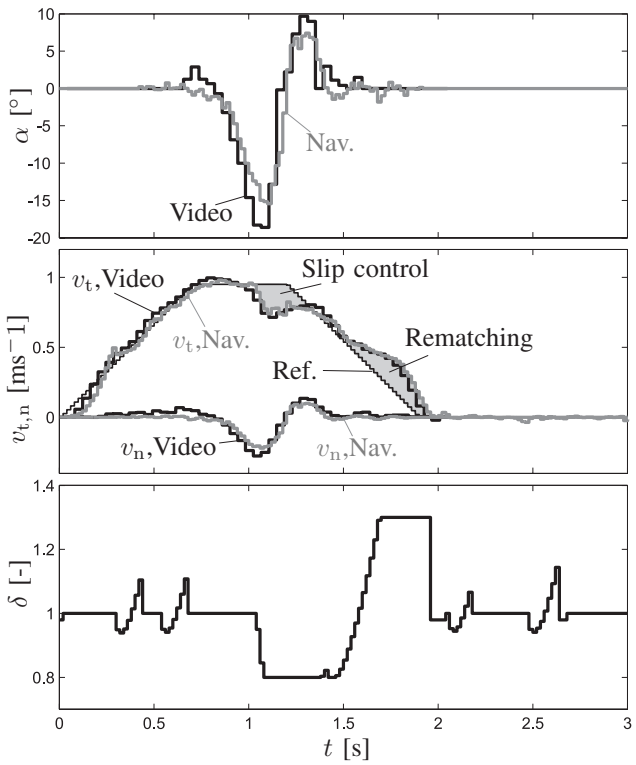

Fig. 6. Tangential and lateral velocity and side slip angle estimate and slip control during tracking of a corner, Ref. indicates the velocities associated with the reference trajectory

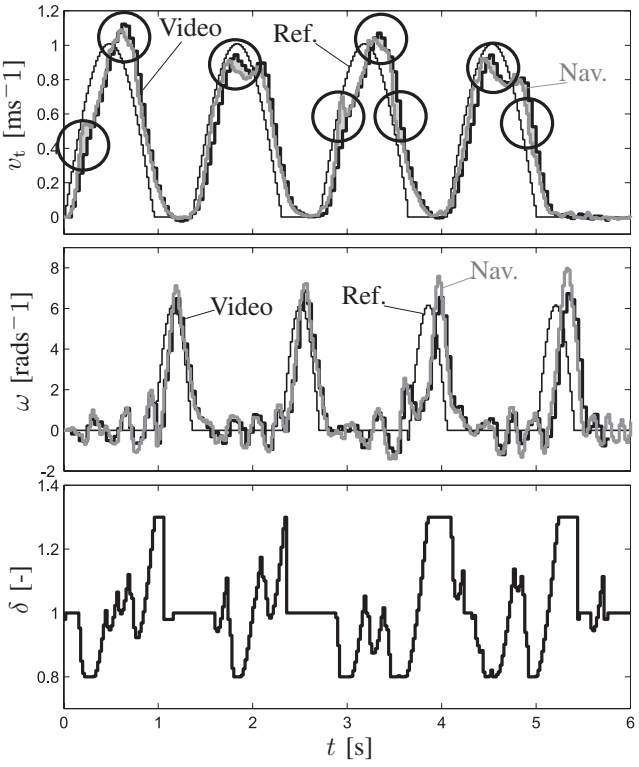

Fig. 7. Tangential velocity and yaw rate estimation and slip control during tracking of a square, Ref. indicates the velocities associated with the reference trajectory

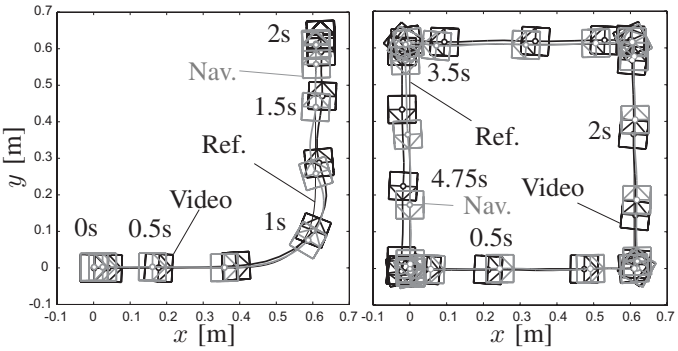

Fig. 8. Posture estimates during tracking of a corner (left) and a square (right)

proprioceptive sensor data has been presented.

The navigation algorithm's governing parameters depend on the employed sensors' characteristics but are independent of the robot's physical characteristics (mass, dimensions).

Experimental results show that navigation with a degree of accuracy of approximately 2.5% position error with reference to absolute trajectory length is possible.

Furthermore, slip control is made possible by reliable slip estimation, both lateral and tangential. In turn, slip-control supports accurate navigation, because it limits the periods with excessive slip to very short intervals.

ACKNOWLEDGMENT

This work was financially supported by the Austrian Science Fund FWF, www.fwf.ac.at.

REFERENCES

[1] J. Borenstein, L. Feng, "Gyrodometry: A new method for combining data from gyros and odometry in mobile robots," in *IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota*, 2005.

[2] S.I. Roumeliotis, G.A. Bekey, "An Extended Kalman Filter for frequent local and infrequent global sensor data fusion," in *SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, 1997.

[3] Y. Fuke, E. Krotkov, "Dead reckoning for a lunar rover on uneven terrain," in *IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota*, 1996.

[4] B. Barshan, H. F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots," *IEEE Transactions on Robotics and Automation*, no. 11, pp. 328–342, 1995.

[5] E. Solda, R. Worst, J. Hertzberg, "Poor man's gyro-based localization," in *IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisboa, Portugal*, 2004.

[6] J. Borenstein, L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," *IEEE Transactions on Robotics and Automation*, no. 12, pp. 869–880, 1996.

[7] J. Borenstein, H. R. Everett, L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*. Technical Report, University of Michigan, 1996.

[8] R. Mazl, L. Preucil, "Sensor data fusion for inertial navigation of trains in GPS-dark areas," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 345 – 350, June 2003.

[9] M. Seyr, S. Jakubek, "Mobile robot predictive trajectory tracking," in *ICINCO, Barcelona, Spain*, 2005.

[10] Novak, G. and Mahlknecht, S., "TINYPHOON - A Tiny Autonomous Mobile Robot," in *IEEE ISIE 2005 International Symposium on Industrial Electronics 2005*, Dubrovnik, Croatia, June 20-23 2005, pp. 1533–1538.