Position tracking of a model race car with Inertial Measurement Unit, laser mouse sensor and Extended Kalman Filter

Dimitar Naydenov and Markus Bader

Institute of Computer Aided Automation Vienna University of Technology,
Treitlstrasse 1/4. Floor/E183-1, 1040 Vienna, Austria
e0926254@student.tuwien.ac.at,markus.bader@tuwien.ac.at
https://www.auto.tuwien.ac.at/

Abstract. Autonomous navigation (path-planning, self-localisation and mapping) has become important for the scientific community due to the increasing interest in self-driving vehicles. This work presents a position tracking system without wheel or motor encoders. The system relies on combining data from the input throttle and steering angle (motion commands) with measurements from an Inertial Measurement Unit (IMU) and a laser mouse sensor, using an Extended Kalman Filter (EKF) as a correction mechanism. The system is able to track the vehicle pose at a rate of 100Hz, allowing an accurate position estimate between the self-localisation cycles, which are typically working at a rate of 5-20Hz. The approach has been implemented and evaluated, achieving position accuracy of 97% for distances up to 10m on a flat surface.

Keywords: autonomous navigation, position tracking, dead reckoning, Ackermann steering, IMU, optical mouse sensor, Extended Kalman Filter

1 Introduction

In the last couple of years companies such as Google and Tesla show their interest in the research for self-driving cars. Robotic competitions such as Freescale Cup[1], Carolo-Cup[2] and others explore the possibilities of deploying autonomous vehicles for street navigation, parallel parking and other computer operated transportation tasks. Autonomous navigation requires self-localisation to be implemented to estimate the robot position on a given map. This is commonly done by using laser range sensors or cameras which provide measurements at a rate of 5-20Hz. Often the vehicles are capable of speeds up to 50km/h, which means that a robot would travel more than 1m within an update cycle. Therefore a system is needed to estimate the vehicle pose in between self-localisation cycles at a higher frequency. In this paper a prototype position tracking system for indoor robot navigation is developed, which relies on an Extended Kalman Filter

[1] Freescale Cup: http://www.nxp.com/freescalecup
[2] Carolo Cup: https://wiki.ifr.ing.tu-bs.de/carolocup

to combine the data from the input control commands and the sensor data from an IMU and a laser mouse sensor. The system is mounted on a model race car based on the Tamiya TT-02 four-wheel drive chassis for performance evaluation.

2 Related Work

The development of a precise localisation system is critical for many robot implementations. Some of the techniques for tracking a mobile base include:

Absolute measurement: An example is the Global Positioning System (GPS), which has been the preferred method for localisation of transportation vehicles for the last decade. A receiver mounted on the vehicle obtains signals from geostationary satellites, which provide the geographical coordinates of the object with accuracy up to 3 meters [4]. This method, however, has limited functionality indoors and lacks the precision, when it comes to short distance movements, which is why it is not suitable for this project.

Dead reckoning: This method relies on calculating the postition of a mobile base from its previous position, based on information about the current velocity and heading [1]. For example, measuring the rotation of the wheels could be used to calculate the travelled distance [2]. Another approach in this category is the implementation of an Inertial Measurement Unit (IMU), which measures the acceleration and angular velocity [3]. This paper focuses on combining the data obtained by several dead reckoning techniques to estimate the race car postition.

The Extended Kalman Filter is selected to perform the data fusion and is preffered for its low complexity over particle filters, associated with self-localisation techniques [9] and FastSLAM algorithms [5], which require much higher computational power.

3 Approach

The developed system uses the following set of data sources, software tools, and algorithms to estimate the robot position:

1. **Mechanical model of the robot:** the input throttle and steering angle together with mathematical representation of the vehicle geometry provide basic motion information.
2. **Inertial Measurement Unit:** the IMU is provides orientation information, which is independent from the robot platform and the control commands.
3. **Laser sensor from an optical mouse:** the mouse sensor measures the dispacement of the vehicle relative to the ground beneath it.
4. **Robotic Operating System (ROS):** a software framework, designed to provide a universal platform for the development of robot applications [6].
5. **Extended Kalman Filter:** the filter uses the data acquired from the mechanical model and the sensors to produce a position estimate, based on the confidence interval, attributed to each data source [8].

3.1 Mechanical model

The purpose of this project is to deploy a position tracking system based on the principle of dead reckoning on a model race car with Ackermann geometry. In order to accomplish that a series of tests are performed to establish the relationship between the input control and the corresponding linear velocity $\boldsymbol{v}$ and steering angle ϕ. Figure 1 illustrates the details of the Ackermann mechanics.

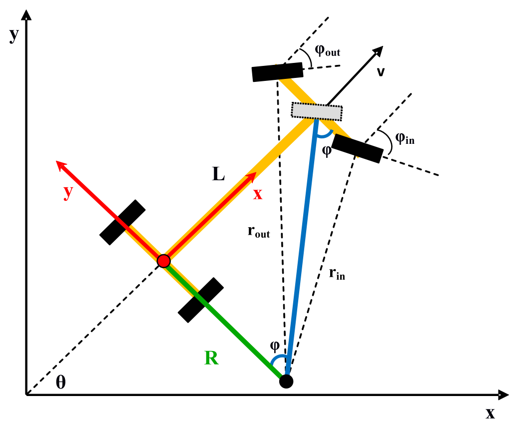


Fig. 1. Tricycle approximation of the Ackermann steering geometry with a steering angle ϕ and a momentary radius of rotation R for a vehicle with a wheelbase L.

The resulting model is used to provide an estimate of the position and orientation of the mobile base. According to the geometry, angular velocity could be defined as the product between the current velocity and the momentary radius, which is a function of the steering angle ϕ:

$$tan(\phi) = \frac{L}{R}; \qquad R = \frac{L}{tan(\phi)}; \qquad \omega = \frac{v}{R}; \qquad \omega = \frac{v \cdot tan(\phi)}{L} \tag{1}$$

The system state could be described in this case by the following equation:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} + v_t \cdot dt \cdot cos(\theta_{t-1}) \\ y_{t-1} + v_t \cdot dt \cdot sin(\theta_{t-1}) \\ \theta_{t-1} + \omega_t \cdot dt \end{pmatrix} = \begin{pmatrix} x_{t-1} + v_t \cdot dt \cdot cos(\theta_{t-1}) \\ y_{t-1} + v_t \cdot dt \cdot sin(\theta_{t-1}) \\ \theta_{t-1} + \frac{v_t \cdot tan(\phi_t)}{L} \cdot dt \end{pmatrix} \tag{2}$$

This approach on its own does not provide enough accuracy for the postition calculation, because of the possible errors in the model construction. For example, the conversion from control values to speed and angle is prone to measurement errors. Moreover, the inertia of the vehicle has to be taken into consideration because the robot needs time to accelerate from $\boldsymbol{v_1}$ to a certain velocity $\boldsymbol{v_2}$. To get better estimates a set of sensors and a correction algorithm are used to improve the accuracy in the position tracking of the vehicle.

3.2 Position tracking system

Fig. 2 shows the structure of the position tracking system. The main controller is a 16MHz ATMega328P, mounted on an Arduino Nano development board which reads the data from the MPU-6050 IMU and the ADNS-9500 laser mouse sensor. The sensor data is sent together with a timestamp to the ESP-8266 WiFi module, which transfers it to a remote computer running the Robot Operating System (ROS) for further analysis. Velocity commands are issued either manually with a joystick or by the ROS nodes used for autonomous navigation. Control data is sent to the Arduino over the wireless link to operate the Electronic Speed Controller (ESC) and the steering servo, using pulse-width-modulation (PWM). The correction in the position of the vehicle takes place onboard the remote computer, which uses the control commands and the sensor data as inputs to an EKF algorithm to produce a corrected state. The separation of data acquisition and algorithm execution on different platforms allows a sensor sampling frequency of 100Hz, which minimises the discretisation error.

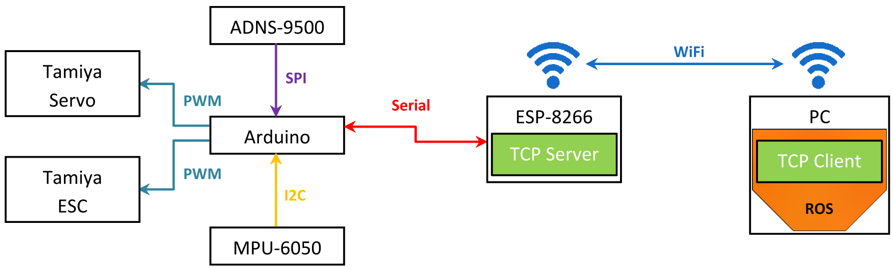


Fig. 2. Communication flow between the components of the position tracking system

The sensor combination is used to compensate for the imperfections in the mechanical model. It is independent from the used platform: the IMU delivers data about its orientation, regardless of how rotation was introduced to the system. The mouse sensor measures the displacement relative to the surface beneath it, so as long as the vehicle is situated on the ground, every movement will be detected. The independence from the underlying mechanics - Ackermann steering, differential drive or some other platform - allows the combination of sensors to detect changes in the robot state, which are not a direct effect of the input control. Therefore the car momentum and wheel slipping/skating could also be accounted for. On the other hand, the mechanical model could compensate the gyro drift of the IMU, since in the absence of movement no change in orientation is expected. Moreover the laser sensor precision depends on the surface roughness and its distance from the ground, which could change in the course of operation, while the velocity commands remain impartial to the terrain. By combining the data from each model in an adequate manner, both approaches could be used in a way to compensate the disadvantages of each other. This is achieved by using the Extended Kalman Filter as a data fusion algorithm.

3.3 Sensor fusion

After data acquisition an Extended Kalman Filter is used to combine the sensor information and the input control commands to produce a weighted average of the vehicle state. The EKF uses a nonlinear state transition function $g(\mathbf{x{,}u})$ to describe how a system evolves from the state $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ under the influence of the control input $\mathbf{u}_t$ and the process noise $\mathbf{w}_t$:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t \tag{3}$$

The system is also supposed to be observable and the performed measurements are described by the following equation:

$$\mathbf{z}_t = H_t \mathbf{x}_t + \mathbf{v}_t \tag{4}$$

where $\mathbf{z}_t$ represents the measurement vector, the matrix H_t transforms the state vector parameters into measurement domain and $\mathbf{v}_t$ is the vector containing the measurement noise, assumed to be zero mean Gaussian white noise with covariance R_t.

The algorithm comprises of two steps - state prediction and measurement update. In the prediction step, an estimate of the state parameters together with an estimate of the new covariance at time t is performed. The prediction step uses the mechanical model for the Ackermann geometry:

$$\hat{\mathbf{x}_t} = \begin{pmatrix} \mu_{x,t-1} + v_t \cdot dt \cdot cos(\mu_{\theta,t-1}) \\ \mu_{y,t-1} + v_t \cdot dt \cdot sin(\mu_{\theta,t-1}) \\ \mu_{\theta,t-1} + \frac{v_t \cdot tan(\phi_t)}{L} \cdot dt \end{pmatrix}; \qquad \hat{P}_t = G_t P_{t-1} G_t^T + V_t M V_t^T \tag{5}$$

For the covariance prediction the state transition function will be linearised at the current mean μ_{t-1} by calculating the jacobian G_t of $g(\mathbf{x}, \mathbf{u})$ with respect to the current state $\mathbf{x}_{t-1}$ and the jacobian V_t of $g(\mathbf{x}, \mathbf{u})$ with respect to the conrol values $\mathbf{u}_t$ to transform the uncertainty in the control, represented by the matrix M, into uncertainty in the state [8]:

$$G_t = \frac{\partial g(\mu_{t-1}, \mathbf{u}_t)}{\partial \mathbf{x}_{t-1}}; \quad V_t = \frac{\partial g(\mu_{t-1}, \mathbf{u}_t)}{\partial \mathbf{u}_t} \tag{6}$$

The correction stage comprises of state correction and covariance update:

$$\mathbf{x}_c = \hat{\mathbf{x}_t} + K(\mathbf{z}_t - H\hat{\mathbf{x}_t}) \tag{7}$$

$$P_c = (I - KH)\hat{P}_t \tag{8}$$

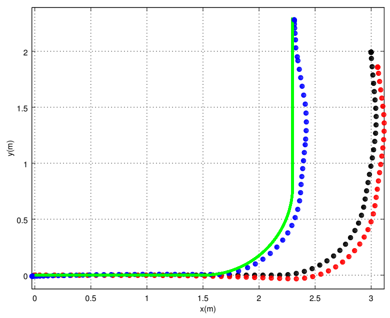
The state correction $\mathbf{x}_c$ is calculated from the state prediction $\hat{\mathbf{x}_t}$ and from the difference between the measurement $\mathbf{z}_t$ and the measurement prediction $H\hat{\mathbf{x}_t}$ from equation 4. The difference is multiplied by a coefficient K, known as the Kalman gain.

$$K = \hat{P}_t H^T (H\hat{P}_t H^T + R_t)^{-1} \tag{9}$$

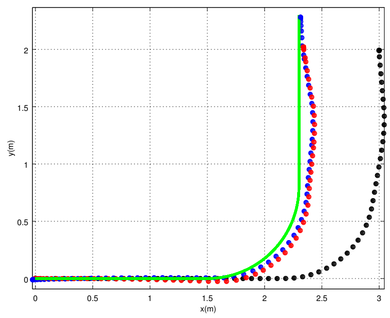
The Kalman gain is a ratio between the the predicted covariance and the measurement noise. The capability of the algorithm to modify the influence of each data source based on the noise associated with it makes the EKF a powerful tool for data fusion [7].

4 Results

The race car was propagated along a certain path to evaluate the accuracy of the position tracking system. The test track consists of a 1.5m long straight line, followed by a 90°arc with a radius of 0.8m and another 1.5m straight section. Figure 3 illustrates the trajectories, computed separately by the mechanical model and the sensor combinations, as well as the data fusion with the EKF.



(a) EKF output (in red) with angle correction



(b) EKF output (in red) with angle and position correction

Fig. 3. Position tracking system trajectories: the test track is coloured in green; the black dots represent the path from the mechanical model; the trajectory from the sensors is marked in blue; the filter output is shown in red

The mechanical model predicts that the vehicle will turn left at a further position along the x-axis, because the model does not account for the vehicle acceleration and therefore the resulting distance estimate will be greater than the actual one. The output of the sensor combination resembles a curve, much similar to the actual test track. On a flat, non-reflecting surface with medium vehicle speed, the sensor measurements indicate high precision. Running the EKF algorithm with correction in the angle estimate produces a trajectory, similar to the one from the mechanical model, but with the curvature from the sensor combination, as shown in Fig. 3a. If the uncertainty in the mechanical model is decreased or the IMU noise is increased, the EKF output will shift towards the state, predicted by the mechanics (the black trajectory). This ability of the filter to change the preffered data source could be utilised to compensate the drift of the IMU gyroscope, when no input velocity is present. Fig. 3b shows the filter output with position correction, where the less noisy laser sensor determines practically the position output. This behaviour can be altered by manipulating the associated uncertainty in case the lens gets contaminated by dirt and the mechanical model remains the only source of motion information. Additional tests on 2m, 5m, and 10m straight courses and three curved tracks with similar lengths indicate high precision measurements with small covariance, as shown in Fig. 4 and Fig. 5.

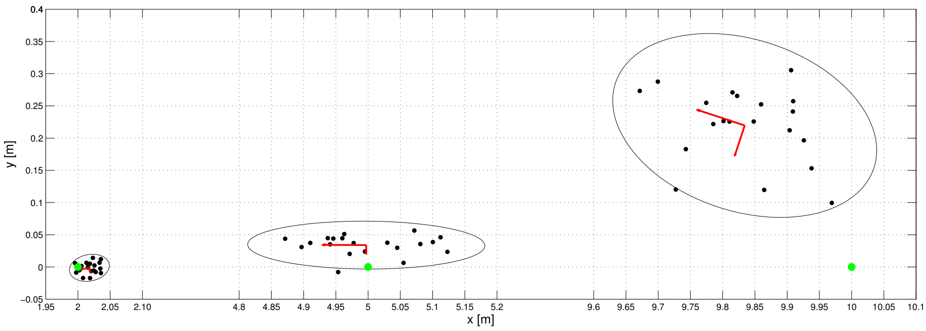


Fig. 4. Sensor results for 2m, 5m and 10m straight lines: the measurements are shown in black; the green circle marks the desired end points; the red lines show the axes of the covariance ellipses

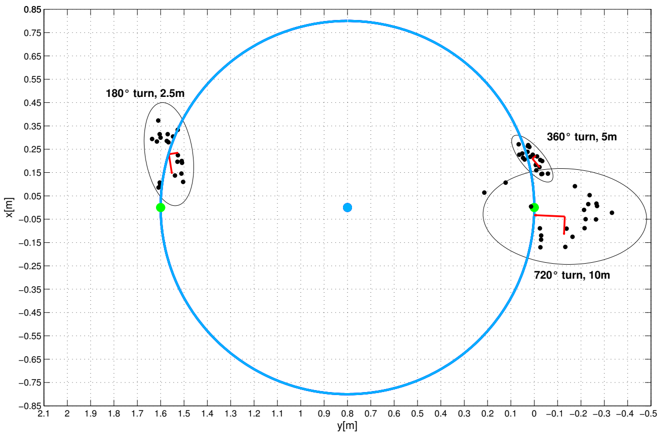


Fig. 5. Sensor results for 180°, 360° and 720° arcs: the measurements are shown in black; the green circles mark the desired end points at 0°(right) and 180°(left); the red lines show the axes of the covariance ellipses; the vehicle moves counter clockwise, starting from (0m, 0m)

5 Conclusion

The delevoped system succeeded in combining the control input from the mechanical model with the sensor data using the EKF to provide a weighted average of the position estimate of the model race car, based on the noise associated with each data source. The experiment results indicate precise sensor measurements for distances up to 10m with mean error under 5% of the total travelled distance. The obtained data exhibits also small variance, with a ratio of the standard deviation to the total traversed path ranging from 0.3% for the straight line tests to 3.5% for the curved track. Moreover, as the system performs as a regular ROS node, it can be easily extended to participate in more complex robot scenarios, such as autonomous navigation in known and unknown environments and support the implementation of self-localisation in the future. The state model used in the EKF could be extended to include the velocity of the vehicle by introducing additional sensors. Detailed examination of the filter in different situations would provide feedback to customise the algorithm for optimal use with the selected mechanical model and sensors. Overall, the system performs as a reliable mechanism for indoor position tracking at a rate of 100Hz.

6 Acknowledgement

The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) according to grant agreement No.6746927 (MPCv1).

References

1. J. Borenstein and D. Wehe H.R. Everettand L. Feng. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 1996.
2. Bong-Su Cho, Woo sung Moon, Woo-Jin Seo, and Kwang-Ryul Baek. A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. *Journal of Mechanical Science and Technology*, 2011.
3. Vikas Choudhary and Krzysztof Iniewski. *MEMS: Fundamental Technology and Applications*. CRC Press, 2013.
4. Ahmed El-Rabbany. *Introduction to GPS The Global Positioning System*. Artech House, 2002.
5. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
6. Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
7. Shu-Li Sun and Zi-Li Deng. Multi-sensor optimal information fusion kalman filter. *Automatica*, 2004.
8. Hisashi Tanizaki. *Nonlinear Filters: Estimation and Applications*. Springer-Verlag Berlin Heidelberg, 2013.
9. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.