15th Triennial World Congress, Barcelona, Spain

DATA FUSION FOR PRECISE DEAD-RECKONING OF PASSENGER CARS

M. Kochem * N. Wagner ** C.-D. Hamann ** R. Isermann *

* *University of Technology, Darmstadt Institute of Automatic Control Laboratory for Control Engineering and Process Automation*

** *International Technical Development Center (ITDC) Adam Opel AG, Rüsselsheim*

Abstract: This paper investigates the possibility of computing the actual position of a passenger car based on different independent internal signals available in a production car. A new front wheel based dead reckoning approach is compared to a model based approach as well as to an Extended Kalman Filter. Depending on the method the input signals are the wheel revolutions of the front wheels, the steering angle and the yaw rate. Experimental results show the quality of all three algorithms compared to position data obtained by a DGPS system and also to the well known rear wheel based dead reckoning approach.

Keywords: vehicle navigation, odometry, Extended Kalman filter, data fusion, automotive, parking assistance system, sensor fusion

1. INTRODUCTION

The realization of a semi-automatic parking assistance system for passenger cars requires the exact knowledge of the vehicle's relative position to the parking spot during the whole parking procedure. Already available parking aids support the driver with optical or acoustical warnings, indicating the distance to a possible rear obstacle. However, the system under development provides the driver with information on whether his car fits into the parking spot and, more important, on how to steer his car to achieve the final position within the spot with minimal steering action. Fully automatic parking has been studied in the past within various projects ((Jiang and Seneviratne, 1999),(Paromtchik and Laugier, 1996*b*), (Cheng *et al.*, 1996)). But, either mobile robots with a fixed axle suspension have been used, or the self-localization has been performed based on ultrasonic sensor data, which was obtained during standstill.

Position estimation can be divided in *internal* and *external* methods (Borenstein *et al.*, 1996). External methods calculate the position with reference to geostationary objects, e.g. satellites in the case of the GPS, or ultrasonic sensors for relative distance measurement, (Paromtchik and Laugier, 1996*a*)). In contrast, internal methods compute the position relative to a starting position based on integration of differential increments, as for example INS or odometry ((Borenstein *et al.*, 1996), (Chung *et al.*, 2001)). Although well known to the mobile robotics society, dead reckoning has not been used much for precise passenger car position estimation so far. Due to the fact, that the axle suspension in passenger cars is not as stiff as in mobile robots. Additionally, the precision is also influenced by other environmental factors such as different road conditions, varying temperature and road surface quality.

This paper presents three different extensions to the well known problem of calculating the position of a ground vehicle based on the measured wheel rotation. One is the position estimation based on measured signals of the Anti-Lock-Braking System (ABS) of both front wheels. The second is a model based estimation method, where the front axle model of the car was obtained using *Modelica*, an object-oriented modelling language ((Elmqvist *et al.*, 1998)), to extend the basic equations of motion of a tricycle. Third, an Extended Kalman Filter (EKF), has been constructed to fuse odometry and gyroscopic measurement data.

The paper is organized as follows. The next section describes the basic vehicle model. In Section 3 the algorithms for the position estimation will be presented in detail. Experimental results in Section 4 show the validity of the proposed methods. The paper concludes with a summary in Section 5.

2. VEHICLE MODEL

The motion of the vehicle is described by the well known non-holonomic equations for a rear wheel driven tricycle (Latombe, 1998):

$$\begin{aligned} \dot{x} &= cos(\theta) \cdot v_R \\ \dot{y} &= sin(\theta) \cdot v_R \\ \dot{\theta} &= \frac{1}{l} tan(\delta_{FM}) \cdot v_R \end{aligned} \quad (1)$$

where v_R is the velocity of the rear wheels, l the wheel base and δ_{FM} the steering angle of the virtual middle front wheel. The equations describe the position of the car coordinate-system, attached to the midpoint of the rear axle, with respect to the world coordinate system as depicted in figure 1. The figure shows a car with the so called *Ackermann*-steering, which means that no lateral forces occur at the front wheels during driving, which is not true for real cars for driving dynamic reasons. Thus, a so called *virtual wheel* is being introduced, comprising the effective steering angles of each front wheel and simplifying the real car to a tricycle model or the so called single track model. It can be further assumed that the no slip condition applies for the wheels. This model is the basis for path planning purposes as well as for the model based position estimation which will be described in detail in the next section. Since this model represents the pure kinematics of a real car, it is valid for low velocities, only.

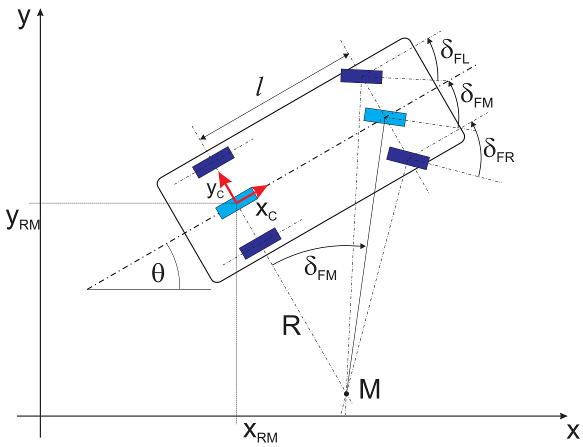

Fig. 1. Vehicle model

3. POSITION ESTIMATION ALGORITHMS

This section covers the proposed methods for the improved position estimation during the parking procedure. The basic equations for dead-reckoning

$$\begin{aligned} x(k+1) &= x(k) + \Delta x \\ y(k+1) &= y(k) + \Delta y \\ \theta(k+1) &= \theta(k) + \Delta\theta \end{aligned} \quad (2)$$

simply state, that the new configuration at step $k+1$ can be calculated based on the old configuration at time step k plus a change of the coordinates and the orientation. It is the basic equation for both the extension of the well known rear wheel algorithm as well as the new algorithm based on the driven path length measured at the front wheels. All following equations apply either for forward or backward driving, dependent only on the sign of the direction, i.e. positive for forward and negative for backward driving.

3.1 Front axle based position estimation

There are two major advantages in using front wheel signals instead of the rear wheel signals only. First, the front wheels provide additional revolution information, independent of the rear wheels. Second, because of the considered vehicle being rear wheel driven, the front wheels have less wheel slip than the rear wheels. However, the position calculation at the front wheels is more difficult than the rear wheel based one, due to the complexity of the front axle suspension in passenger cars. Figure 2 shows the geometry necessary to understand the following equations. The assumption of equal lateral forces at the front wheels holds for equal tires, normal forces and ground conditions. Thus, the vehicle moves along the same trajectory as if it had an Ackermann-steering, i.e. without lateral tire forces. The effective steering angles $\delta^*_{FL,FR}$ for both front wheels can then be combined to the steering angle δ^*_{FM} of

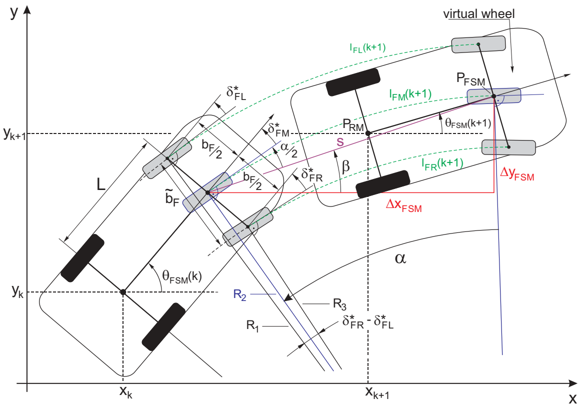

Fig. 2. Geometry for the front axle based position estimation

the virtual wheel in the middle. First, the changes in the front track width b_F and the wheel base L during steering action will be neglected. The driven path length of the *virtual wheel* is then calculated based on the angle δ^*_{FM} and the path lengths of the real front wheels by calculating the mean value

$$l_{FM}\ (k+1) = \frac{R_2}{2} \cdot \left(\frac{l_{FL}(k+1)}{R_1} + \frac{l_{FR}(k+1)}{R_3} \right) \tag{3}$$

with

$$\begin{aligned} R_2 &= \frac{L}{\sin(\delta^*_{FM})} \\ R_1 &= \sqrt{R_2^2 + \left(\frac{b_f}{2}\right)^2 + R_2 \cdot b_v \cdot \cos(\delta^*_{FM})} \\ R_3 &= \sqrt{R_2^2 + \left(\frac{b_f}{2}\right)^2 - R_2 \cdot b_f \cdot \cos(\delta^*_{FM})}. \end{aligned} \tag{4}$$

With this result the angle α of the partial arc is calculated by

$$\alpha = \frac{l_{FM}(k+1)}{R_2} = \frac{l_{FM}(k+1)}{L} \cdot \sin(\delta^*_{FM}). \tag{5}$$

A secondary method of calculating α uses the effective front track width $\tilde{b}_f$ (compare also fig. 2)

$$\tilde{b}_f = b_f \cdot \left(\frac{\cos\left(\frac{\delta^*_{FR}+\delta^*_{FL}}{2}\right)}{\cos\left(\frac{\delta^*_{FR}-\delta^*_{FL}}{2}\right)} \right) \tag{6}$$

and thus yields a relatively simple equation for α

$$\alpha = \frac{l_{FL}(k+1) - l_{FR}(k+1)}{\tilde{b}_f}. \tag{7}$$

With

$$\beta = \theta(k) - \frac{\alpha}{2} - \delta^*_{FM} \tag{8}$$

and

$$s = 2 \cdot R_2 \cdot \sin\left(\frac{\alpha}{2}\right) \tag{9}$$

the change in the position of the front center point P_{FSM} can be computed by $\Delta x = s \cdot \cos(\beta)$ as well as $\Delta y = s \cdot \sin(\beta)$ and $\Delta\theta_{FSM} = -\alpha$. Substituting these values in the basic dead-reckoning equation (2) yields the new position at time step $(k+1)$. Finally, the obtained position has to be transformed to the midpoint of the rear axle. For a linear motion of the car the aforementioned equations simplify to

$$\begin{aligned} \Delta x &= l_{FM}(k+1) \cdot \cos(\theta(k)) \\ \Delta y &= l_{FM}(k+1) \cdot \sin(\theta)(k) \\ \Delta\theta &= 0. \end{aligned} \tag{10}$$

3.2 Model based position estimation

In order to get reasonable results for a model-based position estimation one has to modify the simple tricycle model (eq. (1)) to behave similar to a real car. The inputs of this model are the velocity v_R and the steering angle δ_{FM}. Based on a given steering angle and the wheel base l the driven radius R can be calculated. Unfortunately, this simplification does not apply for a real car, due to nonlinearities especially for extreme steering angles while parking. The main idea of this model based approach is to modify the simple model in a way that the real driven radius R can be derived directly from the steering angle input. In addition to that, the obtained model should be easily adaptable to other vehicles. The inputs of the extended model are, similar to the simple model, velocity v_R and steering angle δ_{FM} (the latter being measured as lateral displacement of the steering bar as described in section 4.1). An object-oriented modelling language (*Modelica*) has been used to derive a kinematic front axle model from construction drawings of a real car, in order to generate several nonlinear mappings modifying the tricycle model. Figure 3 shows the graphical 3D-representation of the object-oriented front axle model. The resulting block structure of

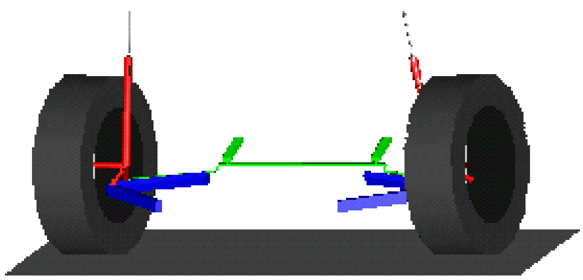

Fig. 3. Graphical representation of the object oriented front axle model

the extended tricycle model is depicted in figure

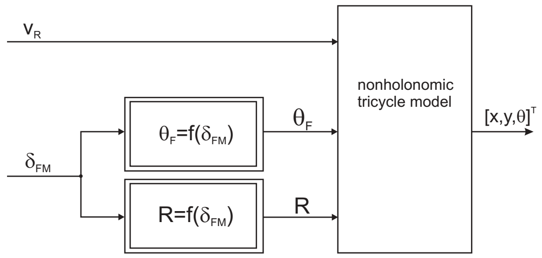

Fig. 4. Extended nonholonomic tricycle model

4. The second nonlinear block $R = f(\delta_f)$ uses the nonlinear mapping in connection with the following equation 11 to compute the real driven circle radius

$$R = \frac{l + \Delta x}{\tan \delta_F} + \Delta y \tag{11}$$

where Δx and Δy are the vehicle specific correction factors obtained by the front axle model. The upper nonlinear block contains the steering angle specific front axle displacement mapping which applies for steering action during standstill only and is not accumulated in equation (2) but nevertheless results in an orientation displacement. This modification has also been calculated based on the external front axle model.

3.3 EKF based position estimation

Basic motivation for the use of a Kalman filter is its ability of fusing independent data from different sources. For nonlinear systems the extended form (EKF) has been already discussed in the literature (e.g. (Welch and Bishop, 2001)). Based on that, other researchers have used the EKF to calculate a mobile robot's position (v. d. Hardt *et al.*, 1996). However, the developed algorithm for generating increments from raw ABS-signals (compare section 4.1) lacks precision for very low velocities (especially at the transition from stop to drive and vice versa), due to the low signal to noise ratio. By using wheel revolution data only, single misplaced increments may lead to an unwanted change in the vehicle orientation. The proposed EKF uses the different path lengths measured at the rear wheels, the velocity calculated based on the front wheel signals and the yaw rate measured independently by a gyro-sensor as measured input signals. The intention is to generate position estimates more precise than by the algorithms previously described. The state-vector is defined as follows: $x_k = [x_{RM}, y_{RM}, \theta, \omega, v]^T$. The time update is

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0) \tag{12}$$

with the nonlinear function

$$f(\cdot) = \begin{bmatrix} vT_0 \cos(\theta + \omega T \cdot 0.5) \\ vT_0 \sin(\theta + \omega T \cdot 0.5) \\ \omega T_0 \\ 0 \\ 0 \end{bmatrix}, \tag{13}$$

that represents, the kinematic vehicle model. The measurement update equation is calculated by

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{14}$$

where K_k is the Kalman gain and $(z_k - h(\hat{x}_k^-, 0))$ being the residual. With

$$z_k = \begin{bmatrix} l_{RL}(k+1) \\ l_{RR}(k+1) \\ \omega_G \\ v_F \end{bmatrix} \tag{15}$$

being the measurement vector and

$$h(\cdot) = \begin{bmatrix} T_0(v - 0.5 \cdot b_r \cdot \omega) \\ T_0(v + 0.5 \cdot b_r \cdot \omega) \\ \omega \\ v \end{bmatrix} \tag{16}$$

where $l_{RL,RR}$ are the driven path lengths, ω_G the measured yaw rate and v_F being the speed of the car measured at the front wheels and subsequently transformed to the center point of the car coordinate system. The Kalman gain K_k is calculated based on A_k, H_k, P_k, Q and R where A_k and H_k are Jacobian matrices from $f(\cdot)$ and $h(\cdot)$, respectively.

4. EXPERIMENTS

This section describes the results of two test runs. One is a forward slalom drive the other a parallel backward parking procedure, consisting of a long backward motion into the parking spot followed by a direction reversal and a short forward motion reaching the final position.

4.1 Experimental system

In order to validate the algorithms a real car (OPEL (GM Europe) Omega Type A, Limousine) has been used as a test vehicle. The rear wheel driven manual gear shift car is equipped with various sensors: a potentiometer for the lateral displacement of the steering bar, which can be transformed into a steering angle measurement, incremental encoders for the wheel rotation (for reference purposes only), measurement devices for the ABS-signals and a Gyro-Box for yaw-, pitch- and roll-rate-measurement. Table 1 shows characteristic values of the test vehicle and some of the sensors.

Table 1. Characteristic values of the test vehicle

Variable	Value	Unit	Comment
b_f	1450	mm	front track width
b_r	1468	mm	rear track width
l	2730	mm	wheel base
r	308.5	mm	wheel radius
n_{ABS}	192	Inc/rev	ABS-signals
$n_{FL,FR}$	400	Inc/rev	front inc. encoders
$n_{RL,RR}$	1568	Inc/rev	rear inc. encoders

A signal processing algorithm extracts 192 increments per wheel revolution without phase delay using the raw data from the production ABS-Sensors. On one hand this information is used to obtain information about the driven path length of each wheel between two sampling steps, on the other hand, the velocity input for the model based position estimation and the EKF is generated using these increments. Obviously, the incremental encoders with 400 and 1568 increments per revolution, allow position estimates of higher precision, but since one goal of this work was to estimate the position using signals available in almost every passenger car, the encoder signals have been used for reference purposes only. All algorithms described above, have been implemented on a realtime signal processing hardware being executed while driving.

4.2 Results

This section shows the results of the position estimation during a parallel parking process and a forward slalom drive using all three algorithms. A Differential Global Positioning System (DGPS) has been used to gather information about the vehicle's "real" position during the parking maneuver. The sampling rate of the DGPS is 1 Hz and the limit of the systems accuracy is around 1 cm. First, fig. 5 shows the whole trajectory for the slalom. It can be clearly seen, that the trajectories drift slightly away from the reference path with increasing track length, one drawback of an odometry based position estimation. Figure 6 depicts the zoomed endpoint of the slalom drive. The circle indicates a radius of 30 cm around the DGPS-measured endpoint. Figure 7 shows the trajectory of the center point of the vehicle coordinate system during the parking procedure, consisting of a backward drive, a stop with direction reversal and a short forward drive to reach the final parking position. The rectangle indicates the dimensions of the parking spot. Figure 8 shows the reversing point of the parking process, where the backward motion stops and is turned into a forward motion. The circle indicates a radius of 5 cm around the reversal point. The front axle ABS-based as well as the model based trajectory (with gyro) are well within an acceptable range

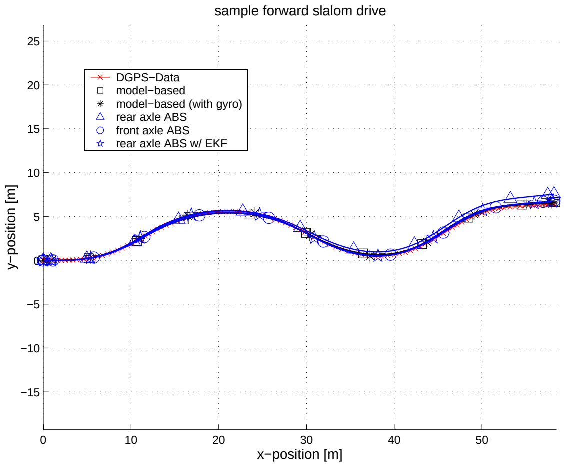

Fig. 5. Trajectory of the vehicle for a simple forward slalom drive

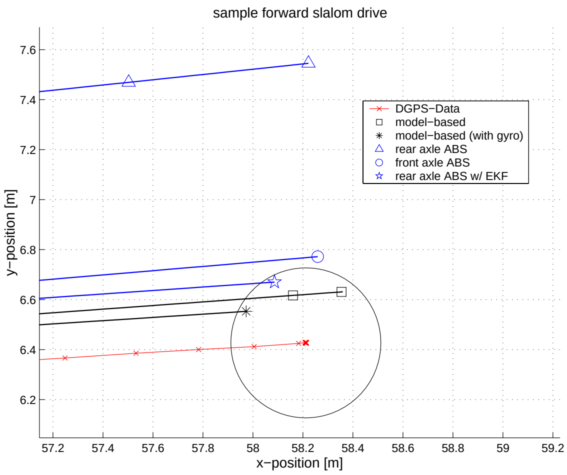

Fig. 6. Motion endpoint from fig 5

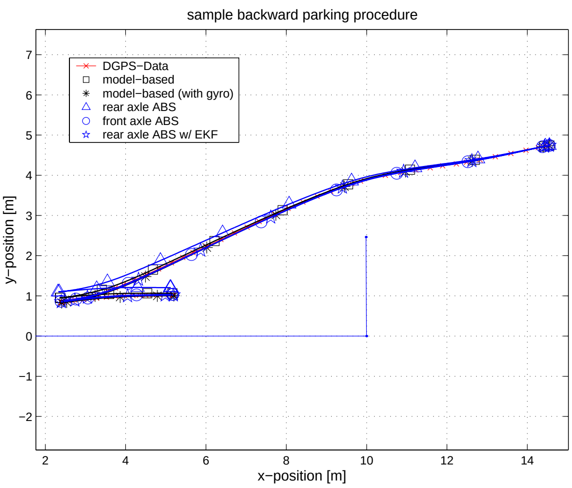

Fig. 7. Trajectory of the vehicle during a backward parking process

for the parking assistance system. A closeup of the final parking position depicts figure 9. Again the circle indicates a 5 cm radius around the final position measured by the DPGS. At this point also the model based position estimate (without gyro) lies within an acceptable range. It can be

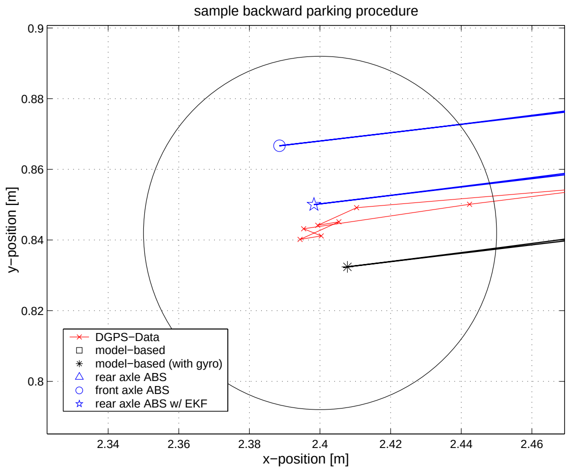

Fig. 8. Closeup of the reversing point in fig. 7

clearly stated that the EKF based position estimation shows extraordinary performance.

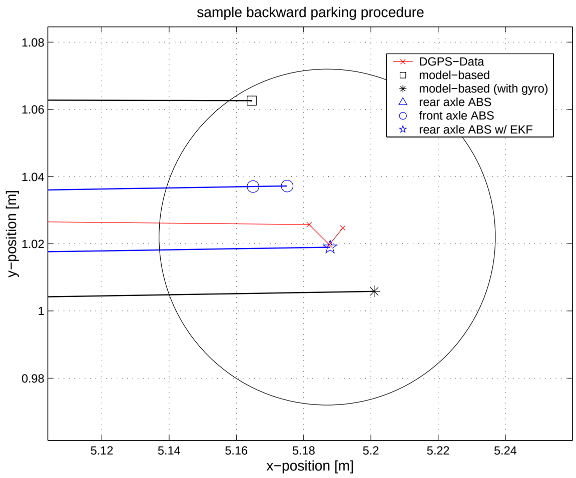

Fig. 9. Zoom of the end point from fig. 7

5. SUMMARY

Continuous precise position estimation is vital for a parking assistance system, which is designed to provide the driver with exact information on how to steer the vehicle into the desired parking spot. In this paper different internal algorithms for obtaining precise position estimates for passenger cars based only on signals already available in production passenger cars have been proposed and evaluated. It was clearly shown, that front axle based dead reckoning, the model based position estimation algorithm and the EKF approach are superior to a simple rear axle based dead reckoning. Modern passenger cars usually provide all necessary signals, for the advanced algorithms. The experiments have proven that basically all three position estimation algorithms yield reasonable results for the parking assistance system. They provide position estimates within an accuracy of 5 cm. Only the EKF shows superior precision of about 1 cm.

REFERENCES

v. d. Hardt, H., D. Wolf and R. Husson (1996). 'The dead reckoning localization system of the wheeled mobile robot ROMANE'. *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems.*

Borenstein, J., H. R. Everett and L Feng (1996). Where am I? Sensors and Methods for Mobile Robot Positioning. Technical report. The University of Michigan, Department of Mechanical Engineering and Applied Mechanics, Mobile Robotics Laboratory.

Cheng, C.-W., Shih-Jie Chang and Tzuu-Hseng Li (1996). Parallel-parking control of autonomous mobile robot.

Chung, H., Lauro Ojeda and Johann Borenstein (2001). 'Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope'. *IEEE Transactions on Robotics and Automation* **17**(1), 80–84.

Elmqvist, H., Sven Erik Mattson and Martin Otter (1998). Modelica - the new object-oriented modeling language. In 'The 12th European Simulation Multiconference, ESM '98'.

Jiang, K. and L. D. Seneviratne (1999). A sensor guided autonomous parking system for non-holonomic mobile robots.

Latombe, J.-C. (1998). *Robot Motion Planning.* number 5.. Kluwer Academic Publishers.

Paromtchik, I. E. and C. Laugier (1996*a*). Autonomous parallel parking of a nonholonomic vehicle. In 'IEEE Intelligent Vehicles Symposium'. pp. 13–18.

Paromtchik, I. E. and C. Laugier (1996*b*). Motion generation and control for parking an autonomous vehicle. In 'IEEE Int. Conf. on Robotics and Automation'. Minneapolis, MI. pp. 3117–3122.

Welch, G. and Gary Bishop (2001). 'An introduction to the kalman filter'. *Department of Computer Science, University of North Carolina* pp. 1–16.