1. Introduction

Navigation of autonomous vehicles and robots can be divided into two categories: indoor navigation and outdoor navigation. In general the outdoor navigation is more difficult and complex task because often the environment does not have characteristic points that can be identified and with respect to which robots could relate their position. If environment is unstructured (e.g. off-road environment), problems related to autonomous navigation are very difficult. Problems related to navigation in the unstructured environment could be divided into mapping, localization, collision avoidance and trajectory tracking.

Mapping and Localization

In an ideal case an autonomous robot has all information regarding the surrounding environment (local) as well as global environment. Information usually is presented in form of map. If map is static that means that the characteristic features of the map are not changing in time and robot can design its trajectory before it moves from original location to its final destination. Very often the map is dynamic, which represents moving objects (e.g. other vehicles, peoples, changing configuration of permanent objects
etc.). In that case an autonomous robot has to create a map as it moves along and navigates in any given environment. This is a local map of the surrounding world, objects, and features in neighborhood of moving robot. The map is acquired using variety of sensors. Some sensors would give information about robot’s position with respect to the nearest structure or obstacle (e.g., laser range finders, sonar, radars, vision systems) while the other sensors would give measurements about distance traveled from the original location (e.g. encoders, INS). This measurement method is known as odometry. Once information about the world around the robot is acquired, the robot can use it for its own localization.

In general two main methods of mapping could be distinguished: first, metric, and, second, topological. Metric maps acquire geometrical features and proprieties of the environment, while the topological methods are focused on connectivity points. The metric methods are dominated by probabilistic methodologies. At present, there is a broad consensus that probabilistic approaches give the best results in mapping and its application in autonomous vehicle navigation. There are several problems with probabilistic methods in mapping. First problem is that algorithms based on those methods require large computational effort caused by high dimensionality. The second problem is related to data association caused by uncertainty if the mapped region can be identified and match with real world. The third problem is connected to measurement noise. In many statistical and probabilistic methods it is assumed that the measurement noise is so called white noise with Gaussian distribution and zero mean value. Unfortunately, this assumption is untenable for robotics vehicle for which measurement noise is correlated and depends on robot position, attitude and design. In short, measurement noise is colored noise. Over many years large number of probabilistic mapping methods have been developed. In this paper only some of them will be addressed. The earliest approach is occupancy grid mapping method with its derivatives. The second are methods based on Kalman filtering and include, between others, simultaneous localization and mapping (SLAM) method. Nowadays, more and more, SLAM name is used in conjunction with methods that are not based on Kalman filtering. The Kalman filtering methods are essentially restricted to linear or linearized systems with measurement white noise assumption. Within the second group there are advanced methods based on adaptive Kalman filtering. The idea behind it related to extension of Kalman filter applicability to measurement with non-white noise. The gain of the Kalman filter is adapted to prevent divergence caused by colored noise. The third approach includes particle filters. Particle filters can be applied for non-linear systems with colored noise.
Sensor fusion is a measurement integration procedure. As the robot navigates, many different measurements and information are obtained from multiple sources (sensors). All information should be integrated using one of the sensor fusion methods. These algorithms can be classified into three into three different groups. First, fusion based on probabilistic models, second, fusion based on least-squares techniques and third, intelligent fusion. The probabilistic model methods are Bayesian reasoning, evidence theory, robust statistics, recursive operators. The least-squares techniques are Kalman filtering, optimal theory, regularization and uncertainty ellipsoids. The intelligent fusion methods are fuzzy logic, neural networks and genetic algorithms.

Collision avoidance

An autonomous robot navigating in crowded (cluttered) environment has to avoid variety of obstacles. Those obstacles may be static or dynamic. In case of static obstacle an appropriate algorithm has to be developed. Very often the method used is a simple proportional, or constant angle navigation. In case of dynamic obstacles a special collision avoidance algorithm has to be developed. Collision avoidance with dynamic obstacles is a much more complex procedure. In this case mapping and active collision avoidance and related trajectory planning is an indispensable part of robot control system.

Trajectory tracking

There are three systems required for the autonomous vehicle to follow the designed path. Those systems are navigation, guidance, and control system. In navigation problem, two basic position-estimation methods usually applied: absolute and relative positioning. For positioning, two types of sensors are available, internal and external sensors. Internal sensor will provide physical data and information that can be measured on the vehicle. The examples of these sensors are encoders, gyroscopes, accelerometers and compasses. External sensors measure relationships between the robot and its environment, which can be composed of natural or artificial objects. The examples of external sensors are sonar, radars and laser range finders.
Measurements of internal sensors are quite accurate for short period. However, for long-term estimation, the measurements usually produce a typical drift. External sensors do not produce the drift, however, because of their characteristics, the measurements are not always available. This problem may be solved by using multi sensors in navigation. Internal sensors can be used to estimate the position of the vehicle during a particular period. External sensors are then implemented to correct the errors that come from internal sensors. Both of those types of sensors have bound errors, and therefore a simple reset of internal errors is not sufficient. A better way is to fuse those two measurements in such a way so it will produce the best desire estimation.

The sensor fusion methods can be divided into three different groups. First, fusion based on probabilistic models, second, fusion based on least-squares techniques and third, intelligent fusion. The probabilistic model methods are Bayesian reasoning, evidence theory, robust statistics, recursive operators. The least-squares techniques are Kalman filtering, optimal theory, regularization and uncertainty ellipsoids. The intelligent fusion methods are based on fuzzy logic, neural networks and genetic algorithms.

This paper presents an example of autonomous robot navigation in specific conditions and environment. In this particular case, a security type robot was used to navigate through gates. This presentation was done on the basis of [1,2] papers.

2. Mobile robot navigation through gates

A robot shown in Fig. 1 was described and used in [1]
The robot was deployed in security applications. This type of robots can carry out patrol tasks in the critical environments such as airport, nuclear power plant stations, gas plants and other industrial establishments. The problem of gate recognition and crossing shown below is a part of this intelligent security mobile robot system. During patrol task of the mobile robot, GPS provides the most global level navigation and often is used for navigation at local level. However, there are some disadvantages of using a GPS sensor. They could be:

1. Periodic signal blockage due to obstruction;
2. Special situations, such as the navigation of through the gate, in which an error in GPS location signal may be too large and have serious consequences.

Those problems are particularly important if differential GPS (DGPS) cannot be used, either because it is not available in the area or it is not desired due to logistic requirements. A few meters precision of basic localization is not sufficient for guiding through the gate or narrow passage between two buildings, where typical width is of 4-6 meters and required precision is of 0.3-05 m. Considering the above reasons, combining GPS with other navigation technologies would be very effective. This method was using in several applications. One of the possible applications of sensor fusion was shown in [3, 4]. This original method was based on adaptive Kalman filtering with fuzzy logic adaptation procedure. This procedure ensured stability and convergence of filter gain.

The gate crossing problem addressed here consists of detecting the entrance using a proximity sensor and then reactively navigating through this entrance. There are several methodologies that can be used to solve this
problem [5]. The gate crossing problem differs from range based wall-following, since it requires the transition from an open field GPS or fusion-based navigation to range-based navigation. This transition itself requires a feature extraction and recognition phase usually not necessary in the environments as underground mines, where range-based wall-following is sufficient [6].

The extraction of robust geometrical features from the data provided by the LMS sensor is not an easy task [7, 8, 9]. The proposed solution for guidance through a gate consists of several steps: environment representation, LMS data filtering, gate recognition, localization and motion control.

The robot platform, ARGO conquest 6x6, Fig.1, is manufactured by Ontario Drive & Gear Ltd. (ODG). It was retrofitted, for computer control and equipped with NovAtel GPS, MicroStrain inertial sensor, built-in wheel odometry, and LMS-221 laser scanner (manufactured by SICK).

The problem described in this paper consists of guiding a vehicle through the area delimited by two objects of known shape called posts located at the known distance from each other (entrance width). The geographical location of the whole structure (called gate) is supposed to be only roughly known. This means that the longitude/latitude of the entrance center point can be transformed to its position on the area map with few meters accuracy, but the geometry of the gate can be known with few centimeters accuracy.

The typical task to be performed consists of navigating along the path that leads to the gate area and passing through the gate. Typical entrance width is 6 m, typical size of the post is about 1 m. Experimental vehicle used for a proof of concept stage is about 2 m wide and 3.5 m long.

**Experiment**

Processing of data provided by a laser scanner is a crucial element for gate navigation. LMS operates according to the time-of-flight principle. The working principle is simple: a laser pulse emitted for a short time is reflected by a target object. A count starts while the pulse is transmitted, and stops when the reflected signal is received. The emitted pulse is diverted by a rotating mirror in the scanner (LMS 2xx User Manual, SICK AG, 2003). The angular resolution of the scanner was set to 0.5°. The maximum range is 80 meters, although we have limited it to 30 m in order to get reliable and meaningful data. Therefore, a full scan of 180° provides 361 range values (indexed according to a scanning angle). The data is transferred to the controlling computer by RS422 interface (500 Kbaud) at which data transferring of a full scan is within 13.3 ms. According to the
manufacturer’s specifications, the scanner’s error is up to ±1 cm within the maximum range 80 meters.

**Gate Identification Procedure and Signature Concept**

Two wooden boxes (cubes of 1.20 m) have been used to construct the gate. The distance between the boxes – the gate entrance - is set to 6.00 m. The vertices are indexed for reference as shown in Fig.2 (the edge is referenced by delimiting vertices).

Depending of the position of the sensor, the perception of the gate obtained using LMS scanner is different. To address the perception of the gate geometry by the sensor, the neighborhood of the gate is divided into proximity zones. Fig.3 shows the ten partitioned zones of the gate.

In each zone some posts' edges are observable while the others are hidden. Fig.4 illustrates visible and hidden edges for the scanner located in zone V: the edges 1-2, 2-3, 5-6, and 6-7 are observable and the edges 3-4, 4-1, 7-8, and 8-5 are hidden.

![Fig. 2 Gate dimension and vertex points (all dimension are in meters)](image1)

![Fig. 3 Gate approach zones for navigation](image2)
The obtained scan (observable image of the gate) is shown in the inserted rectangle. As can be easily seen, each scan contains two portions (corresponding to two posts); each portion contains an isolated segment or two adjacent segments (a corner). Fig.5 is another scanning illustration from zone IX.

Scanning from zones (I-IV, X in Fig.3) gives poor representations of the gate as illustrated by icons in the table 1. For Zone I and II there is only one visible edge, in zone III and IV the reflection degrades since the part of the surface is illuminated at a very small angle. Our control algorithm will try to avoid guiding through those zones I-IV, using zones V-IX instead. Zone X also gives a poor representation of the gate, but this zone is inevitably visited by the vehicle after it has almost crossed the gate area. Recognition is not an issue by that time – the gate is almost crossed but the special procedure is needed to estimate the relative pose of the vehicle in this zone. Gate visibility from all ten zones has been presented in Table 1.
Table 1 Gate visibility from the respective zones

<table>
<thead>
<tr>
<th>Zone</th>
<th>Gate Segments</th>
<th>Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1]</td>
</tr>
<tr>
<td>11</td>
<td>_ _</td>
<td>_ _</td>
</tr>
<tr>
<td>_</td>
<td>_ _</td>
<td>_ _</td>
</tr>
<tr>
<td>_1</td>
<td>_ _</td>
<td>_ _</td>
</tr>
<tr>
<td>_x</td>
<td>_ _</td>
<td>_ _</td>
</tr>
</tbody>
</table>

In order to compute the vehicle’s location relative to the gate, the zones have first to be distinguished from which the scan is obtained. The concept of the “gate signature” has been proposed. Signature vector is computed considering the length of each observed segment, the distance between two continuous portions of the scan (gap), and the distance between the scan start point and the scan end point (size).

Two examples of signature computation is given in Fig.6: in the upper rectangle, there are four segments, \( ab, bc, ef \), and \( fg; ce \) is a gap between two posts and \( ag \) is the total width of the scan. A vector becomes \([ab, bc, -ce, ef, fg, ag]\). The negative sign of the third component emphasizes the fact that it corresponds to the gap and not to the edge. Similarly, for the low rectangle, the signature would be\([bc, cd, -de, ef, fg, ag]\).
Table 2  Signature of the gate scans

<table>
<thead>
<tr>
<th>Zones</th>
<th>Image</th>
<th>Signature</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td><img src="image1.png" alt="Image" /></td>
<td>[41,42,43,44,45,46]</td>
<td>1x6</td>
</tr>
<tr>
<td>VII</td>
<td><img src="image2.png" alt="Image" /></td>
<td>[41,42,43,44,45]</td>
<td>1x5</td>
</tr>
<tr>
<td>IX</td>
<td><img src="image3.png" alt="Image" /></td>
<td>[41,42,43,44,45,46]</td>
<td>1x6</td>
</tr>
</tbody>
</table>

Table 2 shows the possible canonical signature vectors for the scans obtained from the zones V, VII, and IX. One can see that even the dimension of the signature varies from zone to zone. Canonical signatures are computed for every zone. By comparing this on-line signature against the set of canonical zone signatures we may roughly determine the robot location with respect to the gate (corresponding zone) and then proceed to the gate recognition and finally to the refinement of the localization result and guidance.

**Experimental procedure**

An algorithm of gate recognition is described in this section. The algorithm contains several modules: the GPS/INS/LMS data collection module, the map building module, the map filtering module, the map fitting module, the gate signature calculation module and the unexpected error check module. The flowchart of the algorithm is shown in Fig. 7.

Data is acquired from GPS/INS/LMS firstly with GPS/INS/LMS data collection module. The map-building module computes the map of the current environment using row data from LMS. The preliminary map includes all the visible objects within the range of the laser scanner. Some of these objects are false positives; they do not correspond to real objects but to the noisy LMS measurements (physically attributed to multi-path reflections,
dust etc.). An example of the preliminary map from the raw data is given in Fig 8. The encircled object is a gate. Generally speaking, the preliminary map cannot be used directly because of too many non-gate objects. All the noisy points and non-gate objects from the preliminary map would be removed by the map filtering module. It evaluates all the detected objects by their dimension and relative position keeping only objects corresponding to the gate posts. The approximations for post size and entrance width are used for this computation (50% difference in size and width is tolerated). The map-filtering algorithm is described in details as:

**Step 1:** Set the effective range to \(~30\) m (software limit) while the maximum range of LMS sensor is set to 80 meters (hardware limit). Thus, all points with ranges higher than 30 meters are deleted;

**Step 2:** Delete all the objects consisting of only one isolated point (usually false positives corresponding to noise);

**Step 3:** Determine the dimension of each object and delete those objects that, based on their dimensions obviously do not correspond to the gate (50% tolerance);

**Step 4:** Calculate the distance between two consecutive (scanning from left to right) objects and delete those which, based on their relative distance are clearly not the gate posts (50% tolerance).

Fig. 7 Flowchart of gate navigation module
Fig. 9 shows the filtered map corresponding to a preliminary map from the Fig.8. After the above steps, if the filtered map still contains more than two objects, the tolerance conditions for the dimension and distance are tightened and the filtering algorithm is re-applied.

The points obtained from the scan are discrete, but represent continuous edge lines. Each edge is delimited by vertex points and can be identified by those points. The vertex points are located on the border of the point cloud (have neighbors only on the right or only on the left hand side) and thus correspond to the corner of the object. The estimated gate image is obtained by connecting the vertex point one by one.

The detail of the segment fitting algorithm may be described in the following steps:

**Initial step**: Determine which points belong to the gate. In the module, we did not consider the effect from environmental disturbances and assume that the only object in the filtered map is the gate. In an example shown in Fig.10, the gate includes all the points marked as shown in the lower rectangle of Fig.6 from point (b) to point (g).
Step 1: Scanning from left to right, find the last point of the left hand post, point (d), and the first point of the right hand post, point (e). This is done by comparing the distance between each two consecutive points and the canonical sizes of the post and of the entrance;

Step 2: To find the "corners" (points (c) and (f) in Fig.6 notation), if they exist, we search for points which have the largest distance to the line connecting the first and the last points of each post ((b),(d) and ((e),(g)). For the case illustrated in Fig.10, two corners for the two posts exist.

When all the vertexes/corners are identified, the current signature for the scanned gate is constructed as explained in section III. For example in Fig.5, the current signature is a vector of dimension 6 (full): [bc, cd, -de, ef, fg, ag]. Comparing this current signature with canonical signatures we assume that the gate is recognized if a canonical signature matches the current signature. For matching decision the squared norm of the difference between the current and canonical signature was used.

Let the canonical signature be:

\[ A_0 = [a_{01}, a_{02}, -a_{03}, a_{04}, a_{05}, a_{06}] \]

and, the current signature be: \[ A = [a_1, a_2, -a_3, a_4, a_5, a_6] \]

The criterion norm of \( A - A_0 \) is:

\[ e = (a_1 - a_{01})^2 + (a_2 - a_{02})^2 + (-a_3 + a_{03})^2 + (a_4 - a_{04})^2 + (a_5 - a_{05})^2 + (a_6 - a_{06})^2 \]

For matching the value \( e \) is compared against the threshold, \( e_{\text{max}} \). If, \( e < e_{\text{max}} \) two vectors are matched otherwise they are not. The positive matching result (\( e < e_{\text{max}} \)) means the gate is recognized. The next map fit-
ting step consists of establishing correspondence between observed points and the gate segments (edges of the posts).

Computing relative position of the vehicle with respect to the gate is a crucial step for being capable of designing an appropriate controller for guiding the vehicle through the gate. The localization module outputs the data to the gate crossing control module. Here the final step of processing localization data is described. Let us define a coordinate frame \( \{OXY\} \) attached to a vehicle with the original point \( O \) – being a point of the platform where LMS is located, \( OY \) – being a longitudinal axe of the vehicle aligned with 90° LMS beam. Fig. 10 illustrates the layout. The gate recognition procedure is based on the extraction of points \{1, 2, 3, 4, 5, 6, 7, 8\}. Using this information the middle point \( C \) of the entrance and the direction of the straight line containing \{2, 3, 6, 7\} (from the left post to right post) can be estimated. More precisely, we are estimating the direction of the beam pointing from the left to the right post – let us call it entrance direction, \( P_{\text{enter}} \). An angle (\( \psi \)) of this beam with \( OX \) axe is the first parameter to be output to the control low being designed. Let us defined a target point \( T \) somewhere shifted from \( C \) along the direction orthogonal to \( P_{\text{enter}} \), called the gate direction, \( P_{\text{gate}} \).

Second output parameter is an angle, \( \alpha \), between \( OT \) and \( OY \). As can be easily seen for the vehicle perfectly aligned with the gate both angles are zero. Fig.12 shows extraction of the middle point and entrance direction from the real LMS image.

![Fig. 11 The layout of localization and control method](image-url)
Control System

In order to guide the vehicle through the gate one may apply either planning/correction approach or nonlinear control approach. We have chosen the latter, exploiting the similarity of the problem of guiding a vehicle towards and through the gate with the parking problem addressed in [7]. In both cases we have to drive the vehicle to a desired pose (position/orientation). For the “gate through” problem the desired position is a middle point of the entrance segment and orientation – orthogonal to this segment. In contrast with original settings of [7], in our case the target point is not known in advance but has to be continuously estimated online. This constitutes a remarkable difference between gating and parking problems. Another difference is that we do not constrain the vehicle speed by zero at the target point, but rather set this speed to the desired value (usually lower than in the field). In order to address these differences, we (i) estimate the entrance middle point $C$, (ii) we define a goal point $T$, (it could be any point located in the middle of the gate and little away from the rear side of the gate post,) as a point moving from the entrance middle point along the gate direction (see definition above). This procedure allows us to keep the vehicle offset from the goal point (singular point, [7]), thus ensuring a stable motion across the gate.

An intermediate control output is defined as follows:

$$u = K_x \alpha + K_y \psi$$

(1)
where, $K_1$ and $K_2$ are chosen using a pole placement technique. For our field experiments $K_1 = 0.9$ and $K_2 = -0.4$. This corresponds to both poles set to -1 on the Z-plane.

In order to compute the final control output, a particular steering design of the ARGO vehicle (skid-steering) is taken into account. In order to avoid unnecessary reactions on the small offsets that can even lead to unstable behavior a dead zone $(-u_{\text{min}}, u_{\text{min}})$ for $u$ is introduced and final control output is computed as follows:

$$U = N (\text{sign}(u - u_{\text{min}}) + \text{sign}(u + u_{\text{min}})) / 2$$

(2)

where $N$ stands for a value to be applied for initiating a vehicle turn.

**Results and Discussion**

Using simulation, all the developed gate-recognition and guiding-through algorithms are tested first with synthetic data. In addition, the filtering and signature-based recognition modules have been verified with LMS data collected along the manual driving through the gate. Finally, the algorithm has been fully integrated on-board and tested in the field. Fig.13 shows the whole trajectory of ARGO from GPS data during the experiment. The area in the dashed circle is gate recognition and crossing area.

![Fig.13 GPS trajectory of experiment](image)

The map filtering module continuously monitors the environment based on the available LMS data. If the gate is actually farther than a soft-upper-limit range the filtered map is empty. When the gate posts become observ-
able they completely appear in the filtered map after relatively short transition period corresponding to the partial visibility of the posts, noisy data etc. Our experiments show that this period is usually about 1 second and never exceeds 3 seconds. As soon as the gate becomes consistently visible, the recognition module computes the gate signature, searches for a closest canonical signature, and estimates the vehicle pose relative to the gate. The localization module computes the gate middle point $C$, and gate direction $P_{GATE}$ and sends them to the motion control module, which calculates the control outputs $U$ and send it to the hardware (low level controller). Fig.15 illustrated motion through the gate. Small circles correspond to the vehicle trajectory, stars to few consecutive vehicle locations with gate images acquired from those locations.

![Fig.14Illustration trajectory of ‘gate through’](image1)

![Fig. 15 Recognition error](image2)
The uncertainty (of about 2m) in absolute (GPS/INS/Odometry-based) position of the gate could be seen. In spite of this uncertainty the LMS-based position of the vehicle relative to the gate is precise enough (0.1m, 3 deg) and a vehicle successfully navigates through the gate using our algorithm described above. Stable gate recognition is crucial for subsequent steps of the algorithm. Fig.15 shows the recording of the recognition results while the vehicle is autonomously entering the gate. The offset (2-norm of the difference) is between the current signature and a canonical signature (taken from the data base according to the zone where the vehicle is located). It may be seen that the error is reduced significantly in accordance with the distance to the approaching gate. The soft-upper- limit range has been set based on theoretical and experimental analysis. If the posts are located too far apart although below the hard-range limit, the recognition results deteriorate. This can be roughly explained as follows: observing a small object (a singular point in the limiting case) we do not get enough information for positioning – lateral and orientation offsets are coupled. In practice the computation remains possible but measurement errors are amplified and results become unusable.

The developed concept of gate signature provides an effective method to estimate the relative position between the vehicle and the gate. The motion control algorithm based on the nonlinear transformation to polar coordinates proposed in [10] coupled with on-line estimation of the vehicle pose and enhanced by the moving target point for avoiding singularities ensures the stable gate crossing with acceptable lateral errors of about 0.3 m. In this work the more general fusion of range measurements with GPS/INS/Odometry data have not been addressed. Using methods proposed in [11,12] the absolute positioning of the vehicle along the "gate crossing" portion of the path can be improved. Such a "tightly coupled" absolute and relative localization is expected to improve the reliability of the navigation system providing better global position estimates along the whole path and smoothing the phase of approaching the gate. However, there are interesting methodologies that would allow navigation and the gate crossing with yet higher accuracy. Using sensor fusion method described in [3,4] navigation with high positioning accuracy is possible.
The numerous field experiments performed this year with retrofitted ARGO platform have proven that the signature concept, recognition algorithm, and gating controller work very well and provide an effective mean for carrying out the gate navigation tasks. Fig.16 illustrates the vehicle passing through the gate in an experimental field.

CONCLUSIONS

Navigation of autonomous ground vehicles through the gate and/or around similar characteristic structures or the environment’s features were proved to be difficult. The performed experiments, demonstrated validity and usefulness of presented concepts for a single gate case. More experiments are needed to verify the conditions for the multi-gate case. The possible improvements of the recognition module would let the navigation algorithm recognize more than one gate and then make its decision according the requirement from the upper level control. The requirement may consist of an entry into a particular gate or the sequence of gates in any given sequence.

Some limitations related to the hardware (laser scanner) have been encountered. Conditions for object recognition are difficult in some particular situations. Creating an environment map using multi-sensor information should be considered. In particular, multi-laser systems or a laser scanner linked with radar sensors, sonar, or video cameras ought to be considered. The system improvement with the multi-sensor and sensor fu-
sion procedure would make the recognizing procedure more effective and the overall system more robust.

References

5. Thrun, S., Burgard, W., Fox, D., Probabilistic Robotics, MIT Press, 2005