A Kalman Filter for Wall Following

Emilio Remolina

November 6, 1997

Abstract

We have implemented a wall following control law which takes its input signal from a kalman filter. The kalman filter comprises a wall model as well as the control law dynamics. The benefits of using this kalman filter approach include: (i) the effect of sensor's noise on the control law is lessen, (ii) a compromise between the control's law operational conditions[1] and the actual environment is achieved, (iii) reliable methods can be implemented to detect when the control law should no longer be executed.

We propose a system architecture that allows to detect inconsistences between the environment and the kalman filter's dynamic and measurement models. We develop a strategy for a smooth change to a new wall model (if possible), whenever the current wall model is no longer valid.

1 Introduction

Following a boundary (wall) when navigating an environment is one of the most common tasks for an autonomous robot. Different control laws can be designed to robustly perform this task. However, the operational conditions under which these control laws are supposed to work are not necessarily met by the environment or the robot's sensorimotor aparatous. In particular, sensor noise and unexpected changes in the boundary can cause the control law to produce wrong control signals. In this paper we describe the implementation of a wall following control law which takes its input signal from a Kalman filter. The use of a Kalman filter allows to explicitly deal with the robot's sensorimotor noise as well as to detect when the control law should no longer be used.

In order to follow a wall we specify the distance the robot should keep from the wall (i.e. the `setpoint`). At specific intervals of time, the robot determines how much it has to turn according to the following control law

$$w = K_e e + K_\theta \theta$$

where `w` is the angle to turn, `e` is the current `error` with respect to the given setpoint[2] and θ is the orientation with respect to the wall (see figure 1). The control law is used to define a control input `[d,w]` such that the robot rotates `w rad` followed by a forward translation of `d mm`.

[1]the ideal condition under which the control law works

[2]the `error` is the difference between the setpoint and the actual distance to the wall

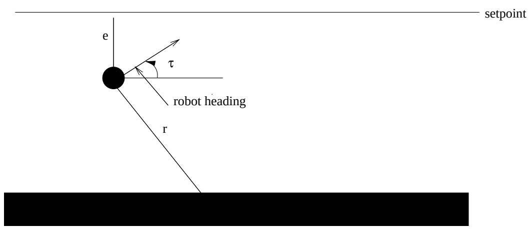

Figure 1: Wall following control law. The setpoint is the distance to keep from the wall. e denotes the difference between the robot's current distance from the wall and the setpoint. θ denotes the robot's orientation with respect to the wall. r denotes the sonar reading.

Previous implementations of this control law calculated the values of e and θ directly from sonar readings (see [4]). In these implementations, heuristics were use to deal with sonar noise, specially with sonar's specular reflection. In this report, we describe the use of a kalman filter in order to calculate the current values of e and θ. We argue that this approach allows for the use of more reliable methods to deal with sensor noise as well as for detecting qualitative changes in the execution of the control law.

2 Kalman Filter

Since the function of the Kalman filter is to produce the input to the control law, we model the state of the system by the tuple (e,θ), where e and θ are defined as above.[3]

Dynamic model. The control input associated with the control law defines the state transition function of the Kalman Filter. The corresponding equations are as follows:[4]

$$\begin{aligned} e_{t+1} &= e_t - d * sin(\theta_t + K_e e_t + K_\theta \theta_t) + w_e \\ \theta_{t+1} &= \theta_t + K_e e_t + K_\theta \theta_t + w_\theta \end{aligned}$$

where d is the forward displacement the robot does between two consecutive applications of the control law, w_e and w_θ are the random noise associated with e and θ respectively.

Measurement model. We only use the reading of the right (left) sonar to calculate the distance to the wall. Let r_t denote the reading of this sonar at time t. Assuming that a sonar behaves like a ray-trace scanner, then

$$r_t = \frac{*setpoint * -e_t}{cos\,\theta_t} + (\frac{\theta_t}{15^o})^2 w_r$$

where w_r is the random noise associated with r.

Notice that the kalman filter's dynamic model corresponds to the dynamics of the control law, while the measurement model is derived from a model of a wall and its relation with the robot's location. The term $(\frac{\theta_t}{15^o})^2 w_r$ is included in the measurement model in order to deal with sonar's specular reflection. This term captures the idea that, when the orientation is not in the interval $(-15^o, 15^o)$, sonar reading are not reliable. Conversely, when the orientation is in such interval, sonar readings should be trusted.[5]

[3] see [2] for an introduction to Kalman filters

[4] More sophisticated vehicle kinematics could be used but this simple `point kinematics model` suffices for our purposes.

[5] The angle of 15^o was experimentally determined.

3 Implementing the Control Law

In this section we describe the overall architecture to implement a wall following control law using a Kalman filter. Notice in figure (2) that , in addition to the kalman filter we have two more modules: the pre-filtering and monitor modules. The pre-filtering module filters out sonar readings that do not agree with the measurement model while the monitor module detects when the dynamic model is no longer valid.

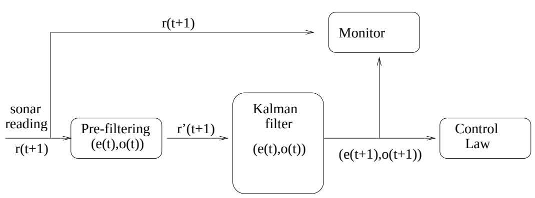

Figure 2: System overall architecture.

3.1 Pre-filtering the sonar reading

One of the problem when using Kalman filters is that the innovation[6] can be very big and consequently the state predicted by the filter is not a valid one. For example, whenever an intersection is present at the end of the wall, the sonar reading will be very big compared to the one expected by the kalman filter. If we input such reading to the filter, it will estimate an erroneous state which in turn will cause the control law to produce a sharp turn angle. The pre-filtering module detects inconsistencies between the actual reading and the kalman filter measurement model. We use a `variant size window` approach to dectect these inconsistencies as explained next.[7]

Given the current reading **r**, the current state `(e,`θ`)`, and the current variance on the estimations of `e` and θ, we define a *confidence interval*, `[min,max]`, around the expected reading associated with the state (e,θ). In order to decide the output of the pre-filtering module we proceed as follows:

1. If $r \in [min, max]$, return r.
2. If $r \leq min$, return min.
3. otherwise, return the expected reading associated with (e,θ).

The justification for (2) and (3) above is that short sonar readings are more reliable than large ones.

3.2 Detecting the end of the control law

Because of the pre-filtering module, the kalman filter does not have access to the actual reading. However, readings rejected by the pre-filtering module could indicate the end of the control law. The monitor modules detect the end of the control law by looking at the `correction` associated with the actual reading. This correction is compared with the previous ones in order to decide whether it is qualitatively different. When various consecutive qualitative changes in the corrections are detected, the end of the control law is signaled. Next we explain how this is done.

[6]The innovation is the difference between the actual measurement and the expected one.

[7]This window correspond to the **validation gate** used in [3].

The Kalman's filter update rule has the form

$$x_t = x_t^* + K * (r - r_t^*)$$

where x_t denotes the state of the system at time t, x_t^* is the dynamic's model projected state at time t, K is the kalman gain, r is the current measurement, and r_t^* is the measurement's model projected reading at time t. The term $r - r_t^*$ is call the `innovation`. The term $K*(r-r_t^*)$ is the `correction` at time t. Notice that the correction is a vector of the same dimension as the system state vector. In our particular application, it is a two-dimensional vector whose first component is the correction in the error and whose second component is the correction in the orientation. The end of the wall following control law is detected as follows:

1. Let N and M denote two natural numbers, and R a real constant.
2. Let `avN` denote the average of the error's correction associated with the last N readings.
3. Given a new error correction, `er`, er is `good` if it belongs to the interval $((1-R)avN, (1+R)avN)$, otherwise er is `bad`.
4. Update `avN` to include `er`.
5. Whenever M consecutive bad error corrections are detected, the end of the control law is signaled.

4 Experiments

In this section we compare the use of a Kalman filter as opposed to the use of raw sonar readings to calculate the error and orientation needed by the control law. For such purpose, we consider the problem of following a 5m long cardboard wall at a distance of `400 mm`(i.e. setpoint = 400 mm). The robot started aligned with the wall at 200 mm from the wall (i.e initial state = (e, θ) = (-200, 0.0)). The robot then followed the wall at a constant speed of 100 mm/sec until an intersection was found. Sonar readings were taken at 1sec intervals. We ran the experiments using two different methods to calculate the error and orientation: the first method used the kalman filter as described above[8]; the second method used raw sonar readings as explained next.

Calculating the error and orientation directly from sonar readings. Let r_1 and r_2 be two consecutive (right) sonar readings, and let `d` denote the distance traveled during these readings[9] Then, the orientation can be calculated by the equation

$$\theta = tan^{-1}\frac{r_2 - r_1}{d}$$

Once the orientation is calculated, the error is determined by

$$error = *setpoint* - r_2 * cos(\theta)$$

Figures (3) through (6) show the evolution of the error and orientation when the robot followed the wall. Figures (3) and (4) show the evolution of the error and orientation as predicted by the kalman filter. Figures (5)

[8]In the actual implementation, two set of readings are taken separated by a 0.5sec period. Consequently, the robot moves 150mm forward between two consecutive applications of the Kalman filter (i.e. d=150).

[9]In the actual implementation we use the following approach to identify sporious readings. The average, `av`, of the previous readings is kept. Whenever r_i is such that $2 * av \leq r_i$, the average reading value `av` is used instead of r_i.

and (6) show the evolution of the error and orientation as predicted by the kalman filter. In all cases, the dashed lines indicate the error and orientation evolution as predicted by the control law under "ideal" conditions.

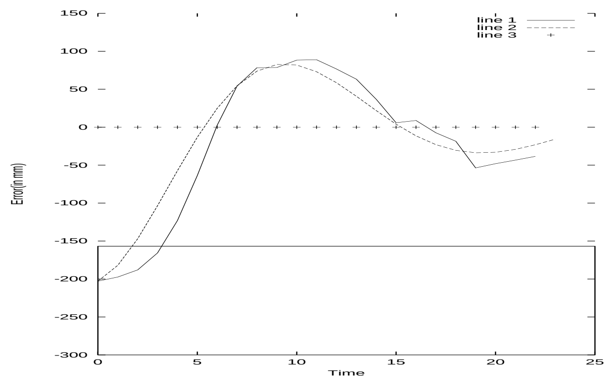

Figure 3: Kalman's Filter predicted evolution of the error. The dashed line indicates the control's law expected error evolution.

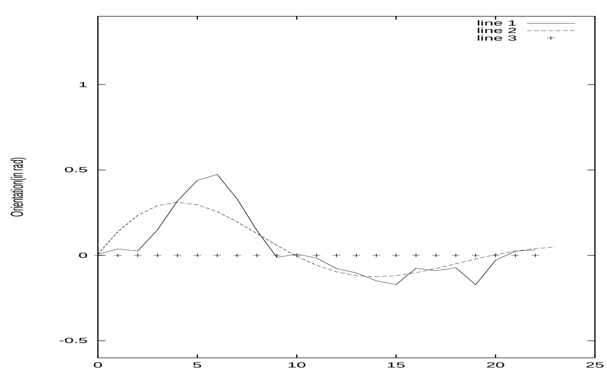

Figure 4: Kalman's Filter predicted evolution of the orientation. The dashed line indicates the control's law expected orientation evolution.

Discussion. It can be observed that the control's law input signal generated by the Kalman filter fits the expected control's law operational conditions. This is not surprising since the Kalman's filter dynamic model corresponds to the control's law dynamics. Differences between the expected and observed conditions are due to noise in the sonar readings as well as odometry errors in the robot's motor aparatous. The Kalman filter compensates these errors while

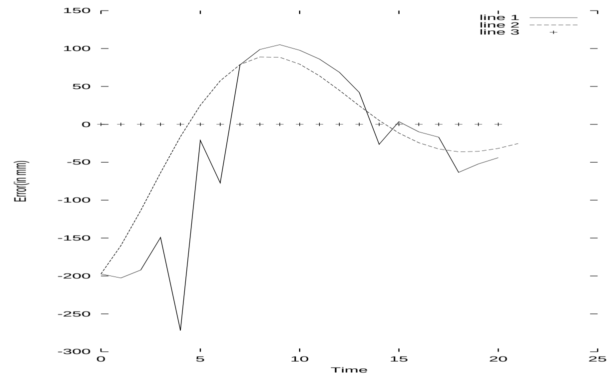

Figure 5: Sonar Based predicted evolution of the error.The dashed line indicates the control's law expected error evolution.

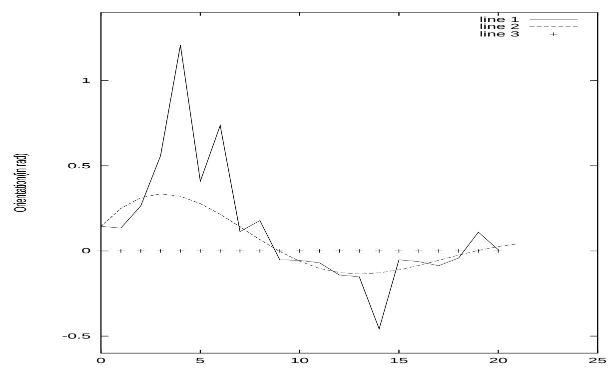

Figure 6: Sonar Based predicted evolution of the orientation. The dashed line indicates the control's law expected orientation evolution.

fiting the control's law dynamics.

The error and orientation calculated from raw sonar readings present an erratic behavior while preserving the "general behavior shape" expected by the control law. This erractic behavior causes the control law to output consecutive contradictory turn angles that make the robot shake while following the wall.

Notice also that there exist significant differences between the control's law expected orientation and the calculated one. In the Kalman filter case, these differences are explained by errors when the robot rotates. The current robot cannot rotate in small angles[10] commanded by the control law. When using raw sonar reading, these differences are due to sonar noise and translational odometry errors when taking the readings used to calculate the orientation.

5 Related work

Leonard and Durrant-White [3] present an extensive use of Kalman Filters for mobile robot navigation using sonar. After pointing out the main characteristics of sonar, they introduce the notion of `regions of constant depth` (RCD) as a qualitative representation of the information content in a sonar scan. RCDs are used for robot localization by matching a predicted RCD to an observed RCD. In addition, they are used for map building by matching multiple RCDs observed from different locations based on different target assumptions. Next we summarize some of the main points in [3] related to the methods presented in this paper.

1. In [3], targets (i.e. walls, edges, corners, etc.) are represented with respect to a global frame of coordinates. A local frame of coordinates suffices for our wall-following control law.

2. RCDs and not raw sensor data are used as input to the kalman's filter measurement model for robot localization. "A RCD provides a means of reducing a target's angular uncertainty, as multiple adjacent returns to the same target constraint the possible true bearing to the target" [3].

3. Measurements has to be validated before using them for position estimation. The `validation gate` approach in [3] reduces to our pre-filtering method when only one reading is available.

4. The RCD's methods used for identifying targets are different from the one we use to detect the end of the control law. In particular, the `circle test`[11] could be used to detect the end of the control law instead of our monitor module.

6 Conclusions and future work

The key ideas presented in this paper are:

1. The effect of sensorimotor noise in the control's law output can be aminorated by using a Kalman filter to generate the control's law input signal.

[10] angles in the interval (-0.1,0.1)

[11] RCDs which correspond to a plane will all be tangent to the plane. See [3] page 99.

2. A Kalman filter allows to integrate the control's law dynamics and a `local` model of the environment. Consequently, a compromise between the control's law operational conditions and the actual conditions of the environment can be achieved.

3. By modeling uncertainty in the current state of the system, reliable methods for detecting inconsistencies between the sensory input and the environment model can be detected. Heuristics for handling sporious sensory input can be replaced by or combined with more solid mathematical tests.

4. Differences between the Kalman filter models and the actual environment have to be detected by additional mechanisms. A prefiltering step identifies differences between the current measurements and the filter's measurement model while a monitor module identifies when the filter's dynamic model is no longer valid.

We are planning to use the methods presented in this paper to implement more complex trajectory-following control law (for example, a middle-line following behavior) as well as add the use of Kalman Filters for autonomous learn behaviors as presented in [1]. In addition, we would like to adapt the RCD's methods presented in [3] in order to detect the end of a control law.

References

[1] Pierce D. and Kuipers B. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–227, 1997.

[2] Welch G. and Bishop G. Scaat: Incremental tracking wiht incomplete information. 1997.

[3] Leonard J. and Durrant-White H. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, 1992.

[4] Lee W.Y. *Spatial Semantic Hierarchy for a physical mobile robot.* PhD thesis, The University of Texas at Austin, 1996. Technical Report AI96-254.