Cooperative Localization Algorithms for Improved Road Navigation

Eric Richter, Marcus Obst, Veit Leonhardt, Robin Schubert, Norman Mattern, and Gerd Wanielik
Professorship of Communications Engineering
Chemnitz University of Technology
Reichenhainer Straße 70, 09126 Chemnitz, Germany
Email: {eric.richter, marcus.obst, gerd.wanielik}@etit.tu-chemnitz.de

***Abstract*—The estimation of a vehicle's dynamic state is one of the most fundamental data fusion tasks for intelligent traffic applications. It has been shown that the quality of GNSS based systems can be improved by the usage of *Smoothed Pseudorange Double Differences*. This paper presents a decentralized data fusion system, which uses a general method for incorporating information from several connected nodes. Both, local and transmitted estimations are represented by particle sets. Simulations show that the position estimate for every vehicle is is in terms of accuracy superior compared to the GNSS solution.**

Keywords: Decentralized Data Fusion, Localization, Cooperative System, Vehicle-2-Vehicle, GNSS, Particle Filter

I. Introduction

Determining the position of a vehicle is an important task in Intelligent Transportation Systems (ITS), both for vehicular safety systems and navigation purposes. Despite of the extensive research which has been carried out to improve the performance and reliability of positioning systems, there are still a lot of unsolved problems. In particular, a high position accuracy can with today's technologies only be achieved by using expensive equipment (e. g. dual frequency receivers or vehicle sensors).

On the other hand, great effort in ITS research is being spend on Cooperative Systems which are based on Vehicle-2-Vehicle- or Vehicle-2-Infrastructure-communication, respectively. Numerous research projects on European or national level are focusing on this topic in order to improve traffic safety and efficiency. Thus, this paper proposes an approach to exploit cooperation between vehicles in order to improve positioning accuracy.

The idea is based on the exchange of GNSS pseudo ranges between several vehicles via an existing communication link. Provided that the same subset of satellites is visible for all vehicles, systematic errors can be partly compensated and, thus, accurate relative position vectors between the vehicles can be determined. Having performed those calculations, all available information (i. e. GNSS navigation solutions, relative position vectors, and velocities/yaw rates measured by vehicle on-board sensors) can be fed into a sophisticated data fusion algorithm. The algorithm, which will be presented in detail in this paper, is based on the Particle Filter framework. It assumes that all vehicles are moving according to the *Constant Turn Rate and Velocity* motion model. With the above-mentioned measurements, it is able to determine a position estimate for every vehicle which is in terms of accuracy superior compared to the GNSS solution.

The paper describes the details of the cooperative positioning approach, in particular of the decentralized data fusion algorithm. Results of simulations are presented and discussed. Furthermore, the influence of different limitation factors such as limited bandwidth or disturbed information from one of the vehicles will be analyzed. Thus, the paper contributes to the further improvement of road navigation systems.

The paper is organized as follows: Section II gives a rough overview of the overall system architecture, section III shows detailed explanations of the *Networked Information Fusion* followed by simulation results and conclusions.

II. System Architecture

The system presented in this paper is a decentralized network of independent nodes, which exchange their knowledge about their own position. Hence, none of the nodes knows about the whole network state and yields an estimation of its own state only. Each node of the decentralized data fusion system is structured as shown in figure 1. Several local sensors like GNSS, yaw rate, and velocity are fused in a "traditional" way using the non-linear *Constant Turn Rate and Velocity* (CTRV) motion model, which would lead to localization errors as shown in [1]. Additionally, *Smoothed Pseudorange Double Differences* [2] and received localization estimations from connected nodes are used to reduce the estimation error. After incorporating new local sensor information, the estimation can be transmitted to the connected nodes.

The main difficulty arises from the fact that the information flowing through the network are not independent and therefore correlated, which would lead to inconsistent and erroneous estimation results if these correlations would not be treated [3]. Section III gives an overview of several consistent data fusion methods proposed in literature and a detailed explanation of the method used in this paper.

III. Networked Information Fusion

When doing *Networked Information Fusion*, one of the main decisions to make is whether to use centralized or decentralized methods for combining observations and estimations from different nodes of the network. While centralized methods

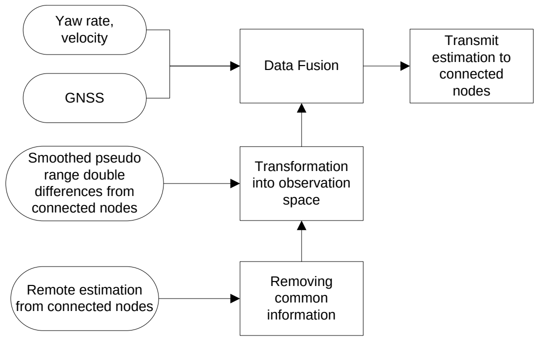

Figure 1. System Architecture

yield some advantages like implicit handling of cross correlations and reduced computational load at nodes by delegating the main effort to the master, decentralized methods promise a higher scalability and robustness without the need for a master instance or additional infrastructure.

In [4], a centralized localization approach is presented, where correlations between nodes are implicitly handled by one large state space and the according linear observation matrix.

In [5] and [6], a decentralized approach for cooperated localization is shown, where each vehicle maintains a large state vector containing the estimations of all connected nodes. These state is updated using local sensor information only and sent to the connected nodes. Additionally, each vehicle holds another large state vector which is updated with the information received by other nodes, but not sent back to the network.

When using decentralized methods, one has to be aware of correlations between estimations from different nodes as they might contain knowledge from other nodes, which has to be considered while incorporating a received estimation in the local one. In [3], [7], and [8] a consistent decentralized method known as *Covariance Intersection* is shown which provides a way of consistently combining two covariances even if there is no knowledge about the cross correlations. As the name indicates, only Gaussian estimation representations can be fused.

A more general solution was introduced by [9] and [10], where discrete particle sets represent the estimations at each node, which enables the ability for representing almost every distribution and non-linear state and observation transitions without linearization.

A. Filter Technology

Particle Filters belong to the class of Bayes filters which recursively estimate the state x_k of a certain system from one time step t_k to the next t_{k+1}. While *Kalman Filters* represent the probability density function (PDF) through parameters (mean and covariance) *Particle Filters* are based on Sequential Monte Carlo Methods (SMCM) and therefore heavily rely on samples also called particles. Instead of the *Unscented Kalman Filter* [11] which represents the PDF with a fixed number of deterministically sampled particles, *Particle Filers* use a large number of randomly generated samples to represent the PDF or density of the system. Due to the use of this non-parametric representation *Particle Filters* are very suitable for non-linear and non-Gaussian applications.

A typical particle filter (SIR) algorithm consists of the following steps:

- sampling step: generation of new particles where each particle is drawn from an importance function $\pi(x)$.
- update importance weights: calcuation of partciles weights $\omega^{(i)}$.
- resampling step: draw M particles $\tilde{x}^{(i)}$ from set S_k according to resampling alogrithm.

First of all a representation of a system and measurement model

$$x_k = f(x_{k-1}) + u_k \tag{1}$$
$$z_k = g(x_k) + v_k \tag{2}$$

is assumed. Here, x_k is the state vector of interest, while z_k represents the vector of observations. u_k and z_k are both independent noise vectors with known distributions. $f(.)$ and $g(.)$ are known (maybe non-linear) functions. There is also a state transition probability $p(x_k|x_{k-1})$ and a likelihood function $p(z_k|x_k)$.

Each particle consists of a certain state x_i and an importance weight ω_i. The single particles which are spread around the state space are combined to the set

$$S_k = \left\{ \left\langle x_k^{(i)}, \omega_k^{(i)} \right\rangle | i = 1, \ldots, N_p \right\} \tag{3}$$

which represents the pdf $p(X_k, Z_k)$. Here, $X_k = \{x_i, i = 0, \ldots, k\}$ contains all states up to time step k. For a large number of particles the approximation of the pdf $p(X_k, Z_k)$ is then given by

$$p(X_k|Z_k) \approx \sum_{i=1}^{N_p} \omega_k^{(i)} \delta(X_k - X_k^{(i)}) \tag{4}$$

where the weights are normalized such that $\sum_i w_k^i = 1$. The denser the particles are concentrated in a certain area of the state space the higher is the probability of the pdf to be approximated.

In a typical initialisation phase the particles are sampled from the initial distribution $p(x_0)$:

$$x_0^{(i)} \sim p(x_0) \tag{5}$$

while the weights $\omega_0^{(i)}$ are set to $1/N_p$.

For a Bayesian Filter framework the transitional density $p(x_k|x_{k-1})$ corresponds to the system update equation. It is used to predict the particles to the next time step t_k.

When a new measurement z_k arrives, all particles need to be judged based on the likelihood function $p(z_k|x_k^{(i)})$:

$$\omega_k^{(i)} = w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)}) p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)} \tag{6}$$

Here, $q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)$ is the proposal distribution where to sample from. The new posterior pdf $p(x_k|Z_k)$ with incorporated measurement z_k is then approximated by:

$$p(x_k|Z_k) \approx \sum_{i=1}^{N_p} \omega_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{7}$$

The described approach is problematic since the variance of the weights $\omega^{(i)}$ will increase over time which leads to the so called degeneracy phenomenon. In consequence only a few particles will contribute to the approximation of the posteriori pdf while most have an importance weight near zero. To overcome this limitation the resampling step is necessary. In short, resampling should duplicate particles having a high weight and eliminate particles containing weights near zero. After resampling the weights of all remaining particles are usually set to $1/N_p$. Typical resampling algorithms are *multinomial sampling* and *systematic sampling* [12]. In this paper *KLD-sampling* [13] is used since it offers quite equal performance and the additional benefit of an adaptive number of particles. Especially relating to bandwidth limitations of the communication channel it may be necessary to reduce the amount of particles after the filter has reached a steady state. *KLD-sampling* uses the Kullback-Leibler (KL) divergence which is a measure of difference between two probability distributions p and q:

$$K(p, q) = \sum_x p(x) log \frac{p(x)}{q(x)} \tag{8}$$

In contrast to [13] we used a slightly different algorithm. The prediction and resampling were separated into two independent steps. Hence, a constant number of particles is firstly predicted via the motion model, while the subsequent resampling step may adapt the number of particles.

B. Distributed Decentralized Particle Filtering

In a decentralized scenario where several nodes communicate with each other and exchange information, data incest can occur if naive approaches are used to fuse the information. This is due to the fact, that in a network estimates based on local information may come back to the originating sender, which would lead to an inconsistent estimation if not handled properly.

In case of particle filters the approach proposed by [9], [10] provides a reasonable method for removing common information, which is in general

$$p\left(x|Z_l \bigcup Z_r\right) \propto \frac{\overbrace{p(x|Z_l)}^{\text{local}}\overbrace{p(x|Z_r)}^{\text{remote}}}{\underbrace{p\left(x|Z_l \bigcap Z_r\right)}_{\text{common}}}. \tag{9}$$

Two operations, multiplication and division, need to be realized implementing (9) using particle representations. Since the support of two probability densities is not necessarily equal, first step prior both operations is to guarantee a common support through a conversion of one of the distributions into a continuous representation.

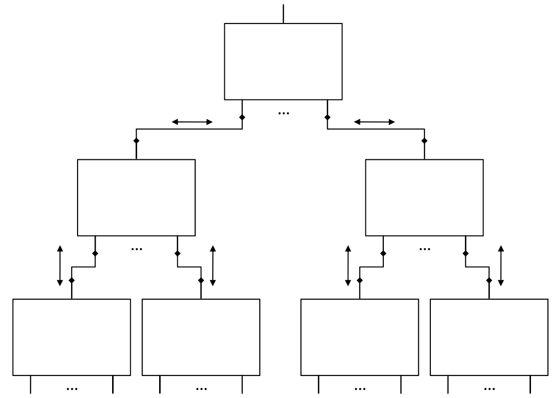

Figure 2. Tree network structure provides a way for removing common information by knowing which information was formerly sent to which node

Equation (9) requires that the common information between two distributions is hold in order to remove it before incorporating into the local distribution. Tree-based network architectures as shown in figure 2 provide a way for easy tracking of common information between two nodes.

C. Implementation

The system presented in this paper estimates the position and the heading of a vehicle using a decentralized distributed filter structure. Each vehicle possesses an independent filter estimating its own position and heading only, not knowing about the existence of other vehicles. The state space consists of three random variables

$$\vec{x} = \begin{pmatrix} x & y & \theta \end{pmatrix}^T, \tag{10}$$

where x/y denote the position and θ the heading of the vehicle.

1) Motion Model: As the application environment is automotive, participants of the cooperative localization system are vehicles, whose movement is bounded to some constraints, derived from their technical architecture. Therefore, movement models can be applied in order to dynamically estimate the parameters from (10). An overview of motion models can be found in [1].

Within this paper, the motion model *Constant Turn Rate and Velocity* (CTRV) from [14] is used, which assumes that velocity v and turn rate [1] ω are piecewise constant. The nonlinear state transition using an input $\vec{u}(t)$ is

$$\vec{x}(t + T) = f(\vec{x}(t), \vec{u}(t)), \tag{11}$$

where

$$\vec{u}(t) = \begin{pmatrix} v & \omega \end{pmatrix}^T. \tag{12}$$

[1]Within this paper, turn rate and yaw rate represent the same physical value, the rotation around the z-axis of the vehicle

Using the constraints of the CTRV model, the resulting state transition becomes

$$\vec{x}(t+T) = \begin{pmatrix} \frac{v}{\omega}\sin(\omega T + \theta(t)) - \frac{v}{\omega}\sin(\theta(t)) + x(t) \\ -\frac{v}{\omega}\cos(\omega T + \theta(t)) + \frac{v}{\omega}\sin(\theta(t)) + y(t) \\ \omega T + \theta(t) \end{pmatrix} \tag{13}$$

for $\omega \neq 0$ and

$$\vec{x}(t+T) = \begin{pmatrix} v\cos(\theta(t))T + x(t) \\ v\sin(\theta(t))T + y(t) \\ \theta(t) \end{pmatrix} \tag{14}$$

for $\omega = 0$.

2) Incorporation of Remote Information: The incorporation of remote information represented through a probability density function with particles is done by the following steps:

(a) Synchronizing estimates: As the remote distribution is usually not synchronized with the local one, the local distribution is predicted to the remote distribution's time using the motion model from section III-C1.

(b) Transformation of received distribution into local marginalized state space $p^+(x^m|Z)$: The smoothed pseudo range double difference distribution $p(d_p)$ is represented by a Gaussian. The received distribution is shifted by $p(d_p)$.

(c) As shown in [10] the division operation

$$p(x) = \frac{p_A(x)}{p_B(x)} \tag{15}$$

leads to weights

$$\omega(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})q(x^{(i)})}, \tag{16}$$

where $q(x^{(i)})$ is chosen to be $p_B(x^{(i)})$

$$p(x) = \frac{p_A(x)}{p_B(x)^2}. \tag{17}$$

Here, $p_B(x)$ is represented by the common last information $p^{ci}(x|Z)$. Due to the particle representation, $p^{ci}(x|Z)^2$ is simply done by squaring the importance weights.

(d) Transformation of last common information into continuous representation $p^c(x|Z)$: The discrete particle representation $p^{ci}(x|Z)^2$ is now transformed into a continuous distribution $p^c(x|Z)$ by applying a Gaussian kernel

$$K_h(x) = \frac{1}{h^{n_x}} K\left(\frac{x}{h}\right) \tag{18}$$

to each sample. The bandwidth h is chosen to be

$$h_{opt} = \frac{4}{n_x + 2}^{\frac{1}{n_x+4}} N_p^{-\frac{1}{n_x+4}} \tag{19}$$

with n_x is the number of dimension and N_p the number of particles (see [15]).

(e) Division of $p^+(x|Z)$ by $p^c(x|Z)$: The devision is done by sampling from $p^c(x|Z)$ with particles of $p^+(x|Z)$ resulting in likelihoods $l^{+(i)}$. The new information is calculated by

$$\omega^{+-(i)} = \frac{\omega^{+(i)}}{l^{+(i)}}. \tag{20}$$

(f) Fusion of new information $p^{+-}(x|Z)$ with local state: At first, $p^{+-}(x|Z)$ needs to be transformed into a continuous representations $p^{c+-}(x|Z)$ as done in step d. Afterwards sampling from $p^{c+-}(x|Z)$ with particles of local distribution $p(x|Z)$ is done resulting in likelihoods $l^{+-(i)}$. The importance weights of the final estimate including remote information are:

$$\omega^{F(i)} = \omega^{(i)} \cdot l^{+-(i)} \tag{21}$$

(g) Transmission of new state to network: The new estimation can either be used for incorporating new local sensor information or be transmitted to connected nodes. In the latter case the common information between communicating nodes needs to be set to the estimation sent.

number of vehicles	1	2		3			4			
vehilce v_i	0	0	1	0	1	2	0	1	2	3
error e	4.1	3.5	2.8	3.3	2.6	2.4	3.2	2.6	2.4	2.5
mean error $\bar{e}$	4.1	3.15		2.76			2.67			

Table I
RMSE FOR DIFFERENT NUMBER OF COMMUNICATING VEHICLES AFTER 300 SAMPLES

IV. Simulation Results

To confirm our assumptions some simulations have been done. In figure 3 the *Root Mean Squared Error* (RMSE) of the position of two vehicles is drawn. The red curve represents vehicle v_0 which only uses local sensor information for the estimation process and transmits the result to veicle v_1 (green line). Since vehicle v_1 starts some seconds later compared to v_0 (simulated delay) it can benefit from the already acceptable estimate of vehicle v_0 which leads to a smaller error at the beginning. The fact that fusion of remote information not only lowers the position error but also produces a smoother estimated trajectory is shown in figure 4. The black line corresponds to the ground truth of both vehicles while the green line shows the trajectory of vehicle v_1 which is slightly smoother compared to v_0 (pink line).

In table I it has been shown that mean error calculated by the RMSE of all involved cars after 300 samples decreases while the number of vehicles grows. The vehicles $v_i, i = 1, \ldots, 3$ made use of the networked information fusion algorithm as proposed in this paper which means that they were transmitting their local estimates while also incorporating remote information.

The influence of the covariance of the smooth double differences is pointed out in table II. For all three simulated vehicles the RMSE is calculated at four different times and compared. A more noisy double difference leads to a smaller

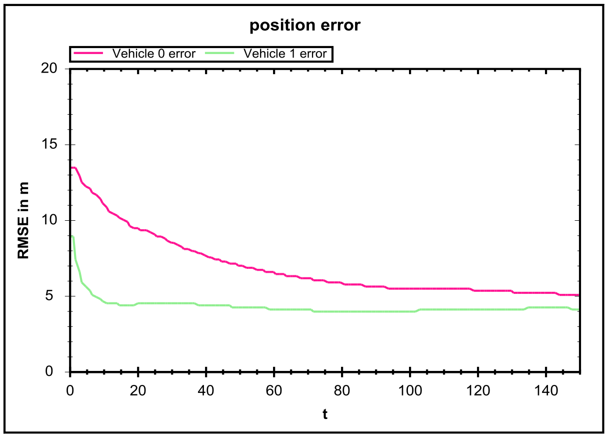

Figure 3. RMSE of two vehicles, one with exclusive local estimates only (pink), the other with fused remote information (green).

σ [m] \ samples	10	50	100	300
1	6.6	4	3.4	2.5
2	6.7	4.3	3.6	2.5
5	7.3	4.7	4.3	3.9

Table II
MEAN RMSE OF THREE VEHICLES AT DIFFERENT TIMES WITH σ VARYING.

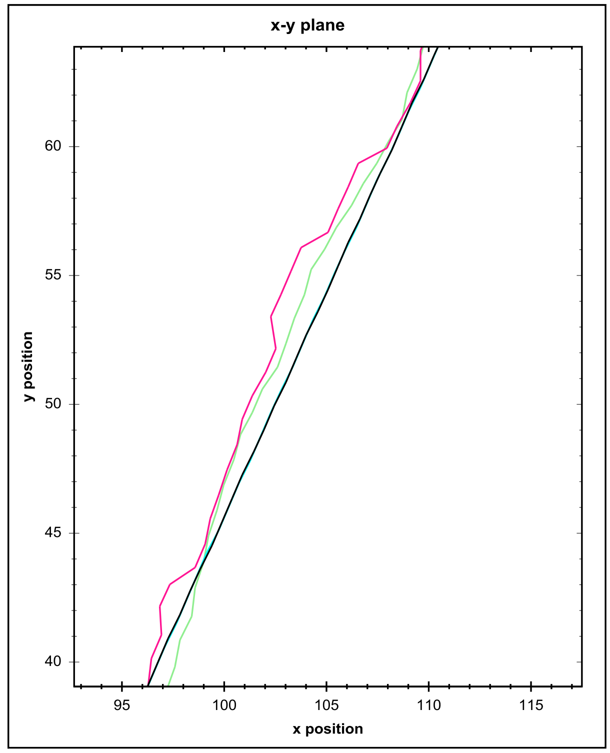

Figure 4. Clipped trajectory of vehicle v_0 (pink) and v_1 (green) compared to ground truth (black).

improvement through the received remote information and therefore to a lower estimation performance.

V. CONCLUSIONS

In this paper, it has been shown that decentralized networks provide a scalable way of fusing information produced by moving groups of vehicles, which leads to improved position estimates. Data incest can be avoided by removing common information using a general method.

The next step will be an evaluation of the proposed system in real scenarios.

REFERENCES

[1] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. 11th International Conference on Information Fusion*, June 30 2008–July 3 2008, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4632283&isnumber=4632178

[2] F. Graas and M. Braasch, "GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results," *Navigation: Journal of the Institute of Navigation*, vol. 38, No. 4, Winter 1991-92, 1991.

[3] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. American Control Conference the 1997*, vol. 4, 4–6 June 1997, pp. 2369–2373.

[4] J. E. Sanguino, "Enhanced waw location-based service positioning," in *The 11th International Symposium on Wireless Personal Multimedia Communications (WPMC 2008)*, 2008.

[5] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Cooperative multi-vehicle localization," in *Intelligent Vehicles Symposium, 2006 IEEE*, 13-15 June 2006, pp. 564–570. [Online]. Available: http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1689688&isnumber=35629&punumber=11114&k2dockey=1689688@ieeecnfs&query=%28karam+n.%3Cin%3Eau%29&pos=6

[6] ——, "Collective Localization of Communicant Vehicles Applied to Collision Avoidance," in *Intelligent Transportation Systems, 2006. Proceedings. 2006 IEEE*, Sept. 17-20, 2006, pp. 442–449.

[7] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *Proc. 8th International Conference on Information Fusion*, vol. 1, 25–28 July 2005, p. 7pp.

[8] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. Fifth International Conference on Information Fusion*, vol. 2, 8–11 July 2002, pp. 901–904.

[9] L.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent methods for decentralised data fusion using particle filters," in *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Sept. 2006, pp. 85–91.

[10] L.-L. Ong, T. Bailey, H. Durrant-Whyte, and B. Upcroft, "Decentralised particle filtering for multiple target tracking in wireless sensor networks," in *Proc. 11th International Conference on Information Fusion*, June 30 2008–July 3 2008, pp. 1–8.

[11] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004. [Online]. Available: http://www.cs.ubc.ca/~murphyk/Papers/Julier_Uhlmann_mar04.pdf

[12] J. Hol, T. Schön, and F. Gustafsson, "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop*, Cambridge, United Kingdom, Sep. 2006.

[13] D. Fox, "Adapting the sample size in particle filters through kld-sampling," 2003. [Online]. Available: citeseer.ist.psu.edu/fox03adapting.html

[14] R. P. S. S. Blackman, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[15] C. Musso, N. Oudjane, and F. L. Gland, *Sequential Monte Carlo Methods in Practice*. Springer, 2001, ch. Improving Regularised Particle Filters, pp. 247–253.