An approach towards smart fault-tolerant sensors

Sebastian Zug and Jörg Kaiser
Institut for Distributed Systems
Otto-von-Guericke-University Magdeburg
{zug,kaiser}@ivs.cs.uni-magdeburg.de

***Abstract*—Acquisition and processing of sensor data has to cope with measurement uncertainties and complex failure modes. Additionally, multiple sensor types and modalities may be used to improve reliability of environment perception. Our work aims at providing an architecture for fault-tolerant sensors and offering a uniform interface to the application. In the paper, we present our fault-tolerant virtual sensor concept that is based on combining model-based estimation and redundant sensor data. To illustrate and evaluate our concept we simulate a mobile robot in an instrumented environment which integrates several smart position sensors. By using a mathematical model to evaluate sensor data we achieve a more reliable position estimation. The paper presents results of the fusion process and discusses methods for generalization.**

***Index Terms*—Abstract sensors, fault-tolerance, data fusion, intelligent sensors**

I. Introduction

Autonomous robots rely to a large extent on a robust and reliable perception of the conditions in their physical environment. Sensors perform this perception of the real world. Two fundamentally different worlds meet at the sensor interface: the real world, characterized by continuous time and continuous valued phenomenon, and on the other side of the sensor interface, the discrete world of a computer, a quantized model of time and a discrete approximation of real-world data. Additionally, the sensor itself may exhibit a sophisticated behaviour when converting the energy of the physical process to the representation, which can be handled in a digital computer. Kopetz at al. [1] speak about (real-time) entities in the real world and the respective (real-time) images represented in a computer.

Usually, once available in binary form, the rt-images are treated like normal time-value entities of the computer. Schemes that deal with fault-tolerance in a sensor-based system often start at that point and use replicated digital data derived from multiple sensors for some form of comparison and voting scheme. These approaches rely on a couple of important assumptions:

Firstly, they assume that replication is possible at all. In many cases like ultrasound distance meters, laser-scanners or other active sensing system this is not easily feasible beside the fact that it is costly.

Secondly, it is assumed that the replicated sensors deliver data within a more or less small range and that faulty sensors can be detected in the form of outliers that substantially exceed the predefined margins. Unfortunately, sensors sometimes have an inherently noisy output and complex failure modes [2]. Hence, distinguishing between a good or faulty behaviour is not as easy when considering just a single sensor value.

Thirdly, it is assumed that all sensor values are available at the same time. However, due to the specific sampling characteristics of a raw sensor, this assumption may not be valid and sensor values may not be accessible at arbitrary points in time [3]. Therefore, they have to be represented in form of some state variables that always constitute the last valid measurement. This state-variable can be accessed whenever the system needs to do so, but of course the temporal validity have to be concerned. In a way, this represents a temporal decoupling between the conversion process and the internal computations. Those three points - mathematical replication, temporal decoupling fault detection and handling - have to be considered by smart sensor model.

In our paper, we strive for exploiting an analytical model of the environment for validating sensor readings. We propose to build a hierarchy of sensor abstractions that compare the information coming from some raw sensor with the expected value derived from virtual sensors based on continuous analytical models. In this way we try to bridge the gap between the discrete and the continuous world and also have some means to detect sensor failures and distinguish them from noise and temporarily incorrect readings. The contribution of the paper is a generic model of such a fault-tolerant sensor architecture and an evaluation of the benefits.

II. Related work

During the last decade a large body of research has been established in the area of distributed preprocessing of raw sensor data. This work strives to hide the complex and individually very diverse behaviour of different sensors and sensor types. Hence, a broad spectrum of mechanisms has been devised for preprocessing sensor data. The different developments aim to one or more of the following main purposes:

a) Fault tolerance: Many authors like [3], [4] aim at fault tolerant distributed measurement systems that react to different sensor or communication faults in a flexible way. Fault tolerance can be reached by sensor and/or mathematical redundancy. Concepts and algorithms depend on the knowledge about the faults and fault probability.

b) Reducing measurement uncertainty: The modules provide a adaptable fusion or filter algorithm to reduce measurement uncertainties, e.g. [5]. Most of the presented work like [6] assumes a unique measurement and sensor type.

c) Programming abstraction: A common interface and programming abstraction makes the concept applicable in a large number of application scenarios [5], [7]. Some implementations offer a specific configuration language for rapid developments [8] or try to support a high level database query interface [9] based on the combination of different sensor types.

d) Context information: Some authors discuss the need of enriching sensor data with context attributes during the first filter and monitor step. Such additional information could be a sensor ID, timestamps, an uncertainty function of the output, validity states of the sensor, sensor position etc. [10]

e) Assignment optimisation: In conjunction with energy efficiency in Wireless Sensor Networks (WSN), [11] tries to optimize the assignment of filter and fusion tasks to individual nodes.

In the following paragraphs we compare the basic terms and concepts.

1) Abstract Sensors: Marzullo [3] defines an abstract sensor as one that consumes the output of a physical transformation process (concrete sensor) and calculates a validity interval. He assumes a continuous uniform distributed sensor measurement. A fusion unit

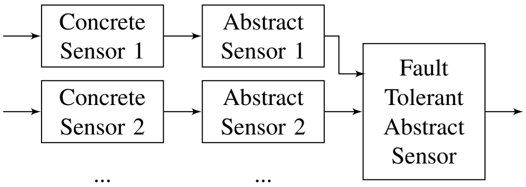

Fig. 1. Marzullos Fault Tolerant Sensor Fusion

receives the results of n abstract sensors and utilizes the redundant validity intervals to distinguish between faulty abstract sensors and correct working abstract sensors. Prasad et. al. [12] and other authors implement the role of abstract sensors in the same way, but vary the error detection scheme. Fig. 1 illustrates the general fusion scheme. The abstract sensor is defined by two properties: Firstly, the knowledge about the sensor's physical behaviour is encapsulated in the abstract sensor and secondly it provides an enriched and application related output interface.

2) Virtual Sensors / Soft-Sensors / Property Estimators: Introducing this classification is based on the fact that quite often the respective physical environment cannot be measured by concrete sensor [7] directly. Therefore, a virtual sensor combines different physical values - pressure and volume for example - to calculate the application relevant unit, e.g. a temperature. In extreme temperature scenarios a concrete temperature sensor may not be available. Another reason could be the precision or the duration of the direct physical measurement in relation to the virtual state acquisition of the application running in the digital computer. Virtual or soft-sensors, as introduced in [13], are used in industrial applications in the context of Model Predictive Control (MPC) [14] since 20 years. Bose et al. [9] enhance the traditional definition of virtual sensors and define a number of subclasses for different hierarchical ordered purposes. The first level singleton virtual sensor accepts only individual measurements and assigns sensor position, sensor ID etc. A basic virtual sensor combines multiple singleton sensors of the same type and provides a better reliability. A derived virtual sensor integrates different basic sensors and provides abstract queries.

3) Logical Sensors: Henderson et al. propose a hierarchically applicable fusion/filter units and call them Logical Sensors [15], [16]. They developed a complete tool-chain for logical sensors description, configuration and code generation. An important goal was the support of a user by a Logical Sensor Specification Language aiming at error detection, diagnosis and recovery. Henderson emphasizes the adaptability of Logical Sensors. The suggested sensor selection mechanism is capable to compensate missing individual sensor measurements by redundant readings.

III. Fault Tolerant Fusion Structure

Our approach integrates the abstract sensor model of Marzullo in a flexible virtual sensor model for dynamic fusion scenarios. Unlike the abstract sensor idea of Marzullo that obtains real measurements of concrete sensors (see Fig. 1) we propose additionally to exploit the global results of the fusion engine in individual abstract sensors. The global virtual results are the outcome of calculating a fusion result based on all available abstract sensor outputs. Inside an abstract sensor the global virtual result is transformed in an appropriable structure.

The general structure of our fault-tolerant sensor framework is depicted in Fig. 2. Considering the application scenario in Fig. 4, a couple of diverse concrete sensors are used to obtain the primary actual position data. The raw data is then merged in the abstract sensor with the expected position data derived from the mathematical model and the outputs of the previous global position estimation of the fusion engine. This procedure is detailed in the subsequent section.

Concrete and abstract sensors as well as the fusion engine are integrated in a virtual sensor. Recursively, this virtual this virtual sensor could be combined with others to a new virtual sensor of a higher abstraction. Fig. 2 illustrates the interaction of the different modules.

A. Concrete sensor

Concrete sensors are transducers representing the interface of a digital system to the environment. A concrete sensor captures and transforms physical values to a digital representation. This conversion process is subject to many uncertainties due to the sensor's physical characteristics that cannot be captured by simple redundancy schemes in the digital part of the system.

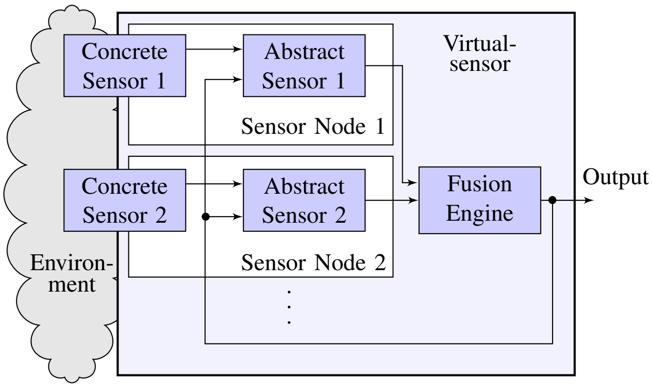

Fig. 2. Extended Fault Tolerant Sensor Fusion Framework

B. Abstract sensors

Abstract sensors consume the measurements of a single concrete sensor (or sensor array of a unique type). They produce an estimate of the current state by applying a form of analytic redundancy. Additionally, the abstract sensor consumes the previous results of the fusion engine and scaling it to the new point in time according to a system and environment model. Thus abstract sensors provide an improved perception of the current environment. An abstract sensor also constitutes a uniform interface for further dissemination of the preprocessed sensor data e.g. in a subsequent fusion node of the current virtual sensor.

The structure of an abstract sensor is depicted in Fig. 3. As shown in Fig. 2 the abstract sensor obtains real measurements and the virtual results of the global fusion. For error detection the measurements are compared against an expected input interval that is either defined by the sensor's specifications or by an application using some knowledge about the environment. For example, measurements of longer distances made by infrared sensors are not considered due to their very low reliability. This information is contained in the attributes associated to the sensor data. In a second step the noise spectrum of the measurements is checked by stochastic tests. A sensor fault leading e.g. to an approximately constant sensor output can be detected by testing the deviation from an expected probability distribution. The main elements of the abstract sensor

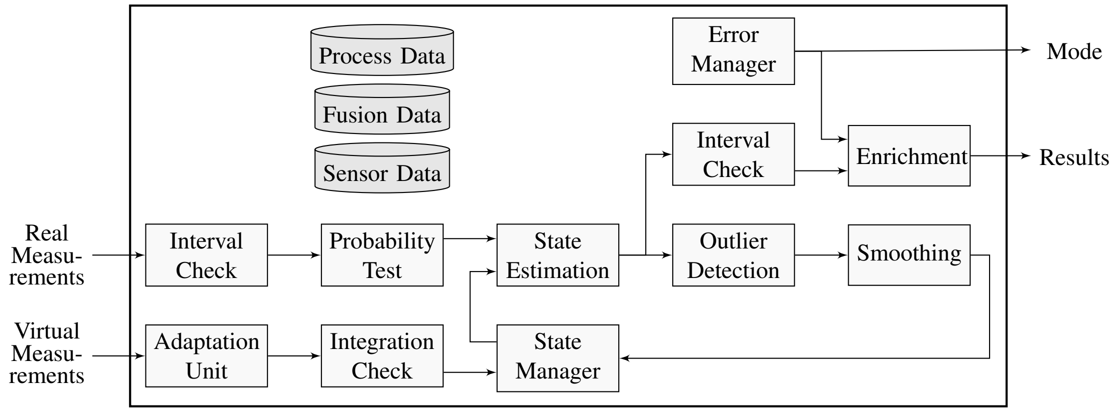

Fig. 3. Abstract sensor structure

are the state estimation and the smoothing algorithm. To ensure a high probability of correct error detection and to smooth the measurements, a mathematical model of the observed process is used, which is defined in the process data sheet. We use this analytic, mathematically founded redundancy to increase the robustness of the abstract sensor. A broad field of methods and algorithms for this purpose were developed. Depending on the application, the developer has to choose an appropriate model of the observed system. For measurements that are disturbed by Gaussian noise only, a Kalman filter can be integrated. More general assumptions about the process and sensor uncertainties and noise may result in using a Bayesian filter unit. For less demanding requirements or systems with substantial performance and memory constraints, simple exponential smoothing or weighted average functions may be sufficient.

Each abstract sensor calculates and transmits its estimation at the fusion point in time. This knowledge is specified in the electronic description of the appropriate virtual sensor and used for the configuration of an assigned abstract sensor. When this point is reached, the new state estimation is calculated and an interval check validates the result. The abstract sensor puts the sensor data into an appropriate format and adds attributes like timestamp and confidence. An error manager continuously monitors the system and may trigger a mode change on detecting an irregular situation.

C. Virtual sensor

Virtual sensors combine a number of abstract sensors and a fusion engine. The fusion engine integrate the "measurements" of all abstract sensors fed by multiple concrete sensors of potentially different type. Due to the synchronisation of all local estimations a large number of fusion algorithms can be adopted. A virtual sensor is defined by an electronic data sheet similar to [17], [18] which contains the appropriate abstract sensor configurations, fusion period and the fusion algorithm.

Virtual sensors can be executed on one of the sensor nodes or on independent hardware. In an environment where the sensor infrastructure may change dynamically, virtual sensors have to adapt to a varying number and diverse properties of the included abstract sensors. A selection algorithm excludes abstract sensors if they exhibit a high degree of uncertainty.

IV. Simulation Scenario

Let us assume, that a mobile robot drives along a line up to 140 cm and comes back afterwards as sketched in Fig. 4. The velocity of the robot is defined by a sine wave with a maximum speed of 40 cm/s and additional noise modelled by a Gaussian distribution. This process is observed by 3 different position sensors: an Ultrasound sensor, a Laser scanner and a Camera system. Fig. 4 illustrates the setup and the defined operating ranges of each sensor system by dashed lines or triangles.

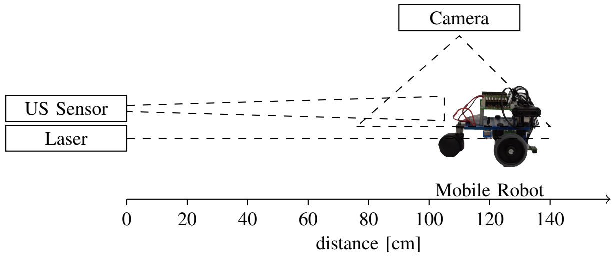

Fig. 4. Mobile Robot Localisation Scenario

All three sensors produce a periodic output of the robot position. The position data is superimposed by a sensor specific constant Gaussian noise. Table I summarizes these parameters. As expected, the measurements of the ultrasound system are more likely subject to measurement uncertainties than the camera output. The assumed amount of noise and its distribution as well as additional disturbance factors are based on a previous experience and evaluation with the respective sensor types.

TABLE I
SENSOR SPECIFICATION

Name	Period	Range	Deviation σ
US sensor	0.05s	0 - 110cm	10.0cm
Laser	0.05s	0 - 140cm	2.0cm
Camera	0.10s	70 - 140cm	0.5cm

The output of such simulated sensors are depicted in Fig. 5 to 7. Each sensor captures the "real" robot movement (dashed (blue) line) according to the range (marked by the darker (red) shaded area) by a noisy measurement (solid (green) line).

The behaviour of the laser sensor, depicted in Fig. 6, is disturbed by two additional factors. Due to the physical measurement principles additional outliers are possible, whose probability and amplitude are controlled by two uniform distributions. The peaks superimposed to the measurement illustrate this fact. Beside the measurement range of each system, that defines a static limit of each sensor, sensor failures like complete crashes are possible. The laser scanner failed 4.2 seconds after the measurements start and from this moment it transfers “0” to its abstract sensor.

The fusion process operates with a period of 0.1 seconds. As described in section III all abstract sensors check their observation state at this point in time and calculate a position, referring to a certain point in time, a deviation estimation and a validity value. Fig. 7 clearly shows the effect of a delay caused by the sensor period. The processing time of the camera delays the measurement. This fact has to be considered and encapsulated in the abstract sensor. Hence, an appropriate synchronisation that allows a precise identification of the measurement time is necessary for any assignment of estimation and measurement. In this first implementation we are not concerned with problems of clock synchronization and assume a global time for all participants. Future work will cope with a realistic uncertainty margin of global timestamps.

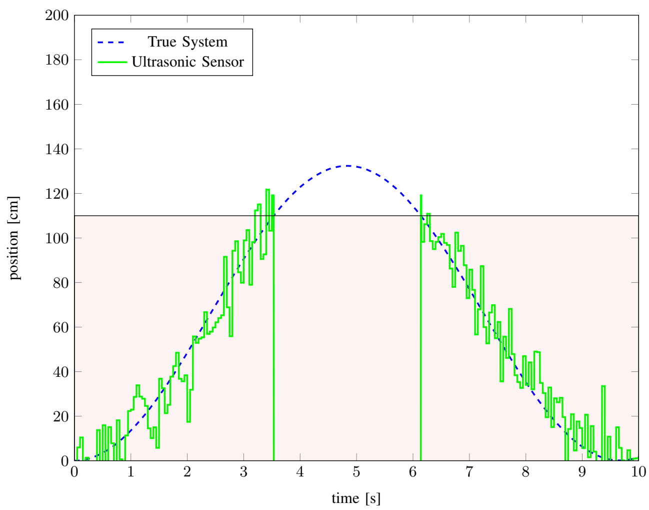

Fig. 5. Ultra sonic measurements with a limited observation range

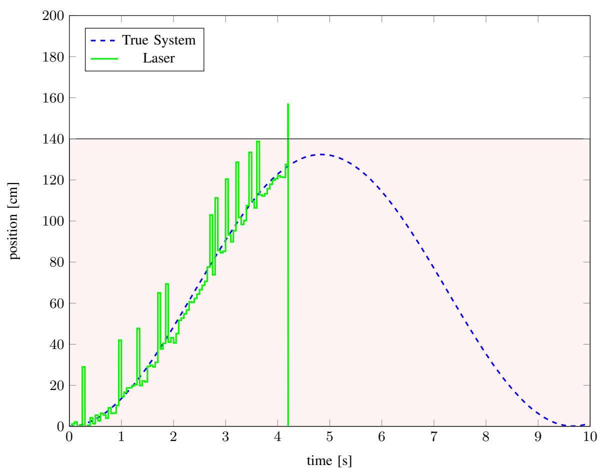

Fig. 6. Laser Measurements with outliers and sensor crash at 4.2 s

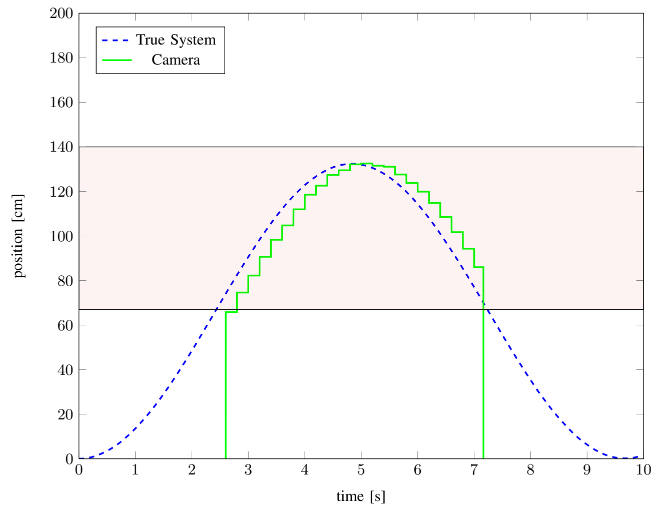

Fig. 7. Camera based localisation of the robot

A. System Model

As described for example in [19] a noise acceleration model is used in each abstract sensor for smoothing and estimation. The manoeuvre model of the robot movement is defined for two state variables - position and velocity $x(t) = [p\ \dot{p}]^T$ - in a discrete time system

$$x\left(t_{k+1}\right) = \left[\begin{array}{cc} 1 & \Delta T \\ 0 & 1 \end{array}\right] x\left(t_k\right) + v\left(t_k\right) \quad (1)$$

where ΔT is the sampling interval of the sensor or the temporal distance from the last sensor measurement to the current estimation. Variable $v\left(t_k\right)$ represents a zero-mean white Gaussian process noise with the covariance

$$cov\left\{v\left(t_k\right)\right\} = \left[\begin{array}{cc} \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^2/2 & \Delta T \end{array}\right] q \quad (2)$$

The Kalman filter produces good results for $q = 2$ in relation to the simulated movement.

Of course, the model does not fit to the non-linear sine wave based “real” robot motion. For the concrete scenario we had to find a more sophisticated model. We do not aim at perfectly adapted mathematical description of the process. We want to show, that a quite general model can be used in an abstract sensor with good results.

The observation model for the sensors is defined by

$$y\left(t_k\right) = \left[\begin{array}{cc} 1 & 0 \end{array}\right] x\left(t_k\right) + w\left(t_k\right) \quad (3)$$

where w is a zero-mean white Gaussian process with the covariance

$$cov\left\{w\left(t_k\right)\right\} = R_i \quad (4)$$

The initial state of the system is assumed to $x(t) = [0\ 0]^T$ for all system variations.

B. Fusion engine algorithm

The fusion engine in the virtual sensors combines all abstract sensor results by a simple weighted average algorithm. The weights are derived from the uncertainty of a measurement which is determined in the covariance Q_n of the estimation of the nth abstract sensor. The algorithm proposed in [20] allows a successive combination of a variable number of abstract sensor estimation vectors x_n to $\hat{x}_n$ with a common covariance $\hat{Q}_n$.

$$\hat{x}_n = \hat{x}_{n-1} + \hat{Q}_{n-1}\left[\hat{Q}_{n-1} + Q_n\right]^{-1}\left(x_n - \hat{x}_{n-1}\right) \quad (5)$$

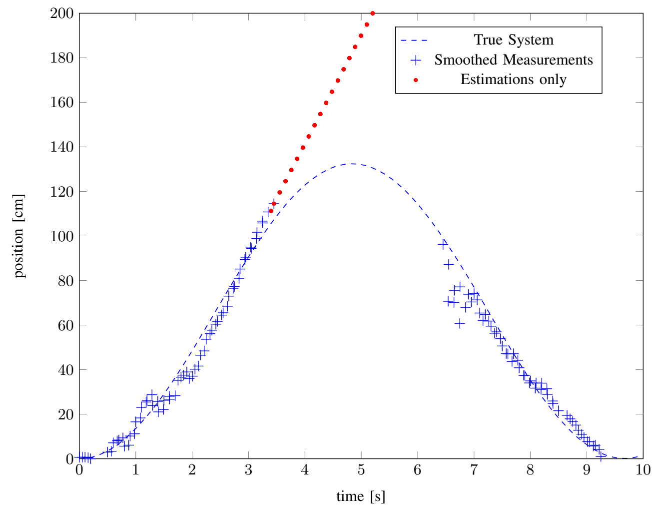

Fig. 8. Abstract Ultra-Sonic sensor results without virtual measurements

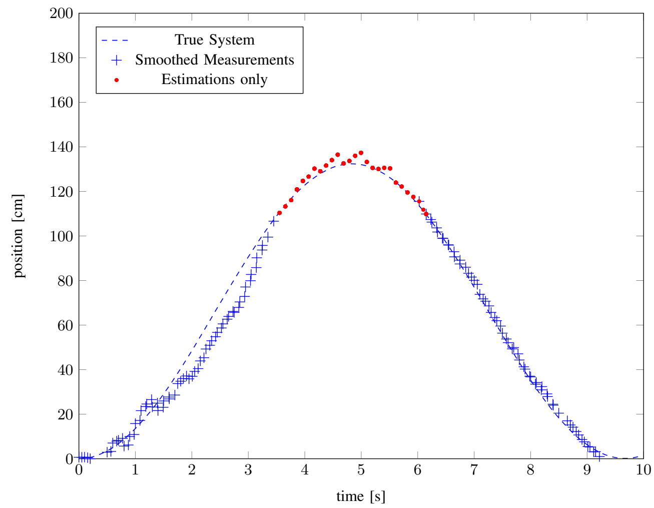

Fig. 9. Abstract Ultra-Sonic Sensor Results with virtual measurements

The current covariance $\hat{Q}_n$ is calculated successively too.

$$\hat{Q}_n = \hat{Q}_{n-1} - \hat{Q}_{n-1}\left[\hat{Q}_{n-1} + Q_n\right]^{-1}\hat{Q}_{n-1} \qquad (6)$$

Precondition of this algorithm is a Gaussian distribution for all measurements. The equation depends on the actual configuration and has to be adapted at each fusion step. For example, if sensor 1 and 3 are working correctly we need two iterations of the central fusion.

C. Implementation

The scenario was implemented in Matlab/Simulink. A continuously simulated block provides the "true" robot movement. The position and velocity output is transformed according to the sensor specification by the 3 simulated sensors in a discrete simulation. Fig. 5 to 7 show the measurements up to 10 seconds. The dedicated abstract sensors smooth the disturbed measurements and calculate an estimation of the position at each fusion step. The results of each independent abstract sensor are transmitted to the fusion engine that merges the estimations according to their uncertainty. The feedback loop transmits the knowledge of the current global estimation to all abstract sensors.

Sensor types, errors and specific behaviours are defined and controlled by Matlab scripts, this structure allows flexible and automated test cases.

D. Results

The Simulink Model mentioned above was validated in two ways: with and without considering of the current global fusion result by the abstract sensors. The benefit of incorporating the global estimation can be seen by comparing of Fig. 8 and Fig. 9. The dashed blue line represents the real movement of the mobile system. The blue crosses mark the estimation based on a valid measurement (see Fig. 5). The first 2.5 seconds of the scenario are characterized by a transient oscillation of the system, afterwards the filter works steady.

When the robot reaches the limit of the observable area the algorithm cannot use real measurements for correcting the estimations. If the robot returns and enters the observation area again, the filter has to recognize that the model based calculation was wrong. It needs some time (until second 7 in the diagram) for stabilizing the state estimation. To overcome this problem, we include the global estimation, that supply each abstract sensor with the recent commonly perceived position and velocity estimation $\hat{x}(t) = \left[\hat{p}\ \dot{\hat{p}}\right]^T$ of the fusion engine. Because the individual abstract sensor knows now the current estimation at each fusion point it can process the restarted measurements without any strong oscillation. Fig. 9 shows this situation in the interval between the seconds six and seven. The behaviour of the abstract sensor close to the point where the observation resumes has a strong impact on the difference between real position of the robot and the estimated one. Fig. 10 shows the cumulative error probability for both variants and an optimal estimation. For example, the probability of an error smaller than 5 cm is around 90 percent for the system using the global estimation. An implementation without this knowledge shows a lower probability to produce an error smaller than 5cm (around 75 percent). The example clearly shows an increased position error without virtual measurements. To highlight the benefit of our approach, we compare to a centralised fusion scan-to-track approach. All measurements are processed in the central fusion node. This fusion mode represented

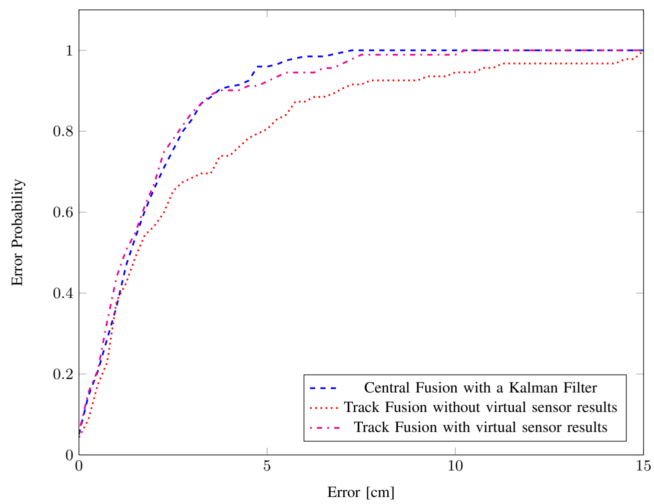

Fig. 10. Cumulative Error Probability

by a dashed line in Fig. 10 possesses an optimal result and the lowest probability to obtain larger estimation errors. But the centralised approach requires a global knowledge of all details about the sensor configurations - observation areas, variances etc. and all sensor measurements have to be received. Hence, the complete centralised fusion wastes a lot of communication bandwidth. The abstract sensor including virtual measurements produces estimations near to the optimal error level but without the communication overhead.

The track-to-track fusion implementation with and without global knowledge (dotted red and dashed magenta line) encapsulates the sensor information and avoids those transmission. Only the current standard deviation has to be attached to each measurement. Especially for a high number of fast sensors the communication effort decreases in relation to a centralised fusion approach. The advantage of the integration of the global estimation as mentioned above is illustrated by a faster convergence to one and is closer to the optimal result (dashed line). The maximum error for the system with virtual measurements is 10.1 cm and for the implementation without this feature we get an error of 19.2 cm. All approaches seem to produce a similar number of estimations close to the real state. Around 12 percent of all estimations posses an error of 0.5 cm.

V. Conclusion and Future Work

We have shown that our approach using an abstract sensor system with embedded in a virtual sensor is capable of improving the sensor characteristics in the presence of noise and failures. We argued that the feedback of virtual measurements to the abstract sensor improve global result. Due to the distributed organisation, this approach is able to handle failed sensors, limited operation areas, noisy measurements and different sensor periods in a flexible way. Furthermore, the results are very close to a purely centralised fusion that produces optimal results at much higher costs. We believe that adding an analytic model in the proposed architecture will considerably improve the reliability of smart sensors and also will be a viable approach for resource constraint embedded systems in future.

Currently we are working on the extension of the presented scheme in the following directions:

- Including communication delays: At the moment we do not consider communication delays in our system. However, due to the distributed nature of our sensor infrastructure, we will need to include these delays as a systematic error.
- Introducing Electronic Data sheets: The "data sheets" of sensors integrated in our scenario so far are simple Matlab structures. A more general solution are presented in [17] and [21]. Based on the XML scheme described models of the abstract sensor should be generated automatically.
- Exploiting real hardware: The current implementation is a simulation only. The Matlab/Simulink environment offers a code generation tool-chain for embedded devices. We work on the transfer of abstract sensor concept to real hardware in a next step. This will also proof that the concept is viable in resource constraint sensor nodes.
- Adding a selection process: The fusion algorithm does not assess the importance of the available sensor data. Very uncertain estimations of an abstract sensor will be excluded by a selection scheme from subsequent fusion steps.

Acknowledgement

This work has partly been supported by the Ministry of Education and Science (BMBF) within the project "Virtual and Augmented Reality for Highly Safety and Reliable Embedded Systems" (VierForES).

References

[1] H. Kopetz, R. Zainlinger, G. Fohler, H. Kantz, P. Puschner, and W. Schutz, "The design of real-time systems: from specification toimplementation and verification," *Software Engineering Journal*, vol. 6, no. 3, pp. 72–82, 1991.

[2] H. F. Durrant-Whyte, "Sensor models and multisensor integration," *The International Journal of Robotics Research*, vol. 7, no. 6, p. 97, 1988.

[3] K. Marzullo, "Tolerating failures of continuous-valued sensors," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 4, pp. 284–304, 1990.

[4] E. Cho, S. S. Iyengar, K. Chakrabarty, and H. Qi, Eds., *A new fault tolerant sensor integration function satisfying local Lipschitz condition*, 2000.

[5] T. C. Henderson and E. Shilcrat, "Logical sensor systems," *Multisensor integration and fusion for intelligent machines and systems*, p. 81, 1995.

[6] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for target tracking," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[7] S. Kabadayi, A. Pridgen, and C. Julien, Eds., *Virtual sensors: Abstracting data from physical sensors*, 2006.

[8] David E. Bakken and Zhiyuan Zhan, "Middleware support for voting and data fusion," in *In Proceedings of the International Conference on Dependable Systems and Networks*, 2001, pp. 453–462.

[9] Raja Bose, Abdelsalam Helal, Vishak Sivakumar, and Shinyoung Lim, "Virtual sensors for service oriented intelligent environments," in *Proceedings of the third conference on IASTED International Conference: Advances in Computer Science and Technology*. Phuket, Thailand: ACTA Press, 2007, pp. 165–170.

[10] A. Herms, M. Schulze, J. Kaiser, and EdgarNett, "Exploiting publish/subscribe communication in wireless mesh networksfor industrial scenarios," in *Proceedings of Emerging Technologies in Factory Automation (ETFA '08)*, Hamburg, Germany, September 2008, pp. 648–655.

[11] R. Kumar, M. Wolenetz, B. Agarwalla, J. S. Shin, P. Hutto, A. Paul, and U. Ramachandran, Eds., *DFuse: a framework for distributed data fusion*, 2003.

[12] L. Prasad, S.S. Iyengar, R. Rao, and R.L. Kashyap, "Fault-tolerant integration of abstract sensor estimates using multiresolution decomposition," in *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, 1993, pp. 171–176 vol.5.

[13] S. Park and C. Han, "A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns," *Computers and Chemical Engineering*, vol. 24, no. 2-7, pp. 871–877, 2000.

[14] S. Joe Qin and Thomas A. Badgwell, "An overview of industrial model predictive control technology," 1997, pp. 232–256.

[15] Thomas C. Henderson and Mohamed Dekhil, "Instrumented sensor system architecture," *The International Journal of Robotics Research*, vol. 17, no. 4, pp. 402–417, 1998. [Online]. Available: \url{http://ijr.sagepub.com/cgi/content/abstract/17/4/402}

[16] T.C. Henderson, M. Dekhil, S. Morris, Y. Chen, and W.B. Thompson, "Smart sensor snow," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 3, 1998, pp. 1377–1382 vol.3.

[17] J. Kaiser and H. Piontek, "CODES: Supporting the development process in a publish/subscribe system," in *Proceedings of the fourth Workshop on Intelligent Solutions in Embedded Systems WISES 06*, Vienna, 30. June 2006, pp. 1–12, iSBN: 3-902463-06-6.

[18] I. . S. Committees, *IEEE Std 1451.2-1997, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators*, 1997.

[19] Yaakov Bar-Shalom, Xiao-Rong Li, and Thiagalingam Kirubarajan, *Estimation with applications to tracking and navigation*. John Wiley and Sons, 2001.

[20] Ashley W. Stroupe, Martin C. Martin, and Tucker Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *IEEE International Conference on Robotics and Automation, May, 2001*. IEEE, 2001.

[21] J. Kaiser, S. Zug, M. Schulze, and H. Piontek, "Exploiting self-descriptions for checking interoperations between embedded components," in *International Workshop on Dependable Network Computing and Mobile Systems (DNCMS 08)*, Napoli, October 2008, pp. 41–45.