Adaptive environment perception in Cyber-Physical Systems

Sebastian Zug, André Dietrich, Christoph Steup, Jörg Kaiser*
Otto-von-Guericke Universität Magdeburg
Faculty of Computer Science
39106 Magdeburg, Germany
{zug,dietrich,steup,kaiser}@ovgu.de

ABSTRACT

The concept of an adaptive acquisition of environment data in distributed scenarios promises a number of benefits. If an application aggregates and uses all available sensing information in an intelligent environment it may provide a higher precision and an increased fault-tolerance. Unfortunately, the application developer has to cope with a number of additional challenges compared to static sensor evaluation. It is not possible to generate an optimized sensor application schedule for a dynamic system at design-time. Due to the adaptive selection process, this has to be executed at runtime. In this paper we propose a general approach for this problem based on a two-level analysis. The first level compares sensor parameter sets (periods, offsets, delays) and application requirements (number of measurements, quality) based on a worst/best case analysis. If a more precise evaluation is necessary, the second level needs to be started. This one considers additional, situation-specific properties like phase shift of sensor periods, communication delays and jitter. At the end, it provides an online optimization of common goals e.g., minimization of the age of data and a constant number of input counts.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer-Communication Networks—*Distributed Systems*

General Terms

Measurement, Reliability

Keywords

Sensor network, Adaptive fusion, Smart sensor, Sensor scheduling, Sensing quality

*Sebastian Zug, André Dietrich and Jörg Kaiser are with the Department of Distributed Systems. Christoph Steup does his research for the Department of Institute of Knowledge and Language Engineering

1. INTRODUCTION

1.1 Motivation

Previous sensor-actuator systems were designed as a combination of statically configured components interacting with each other in a predetermined pattern. If we want to design a mobile robot accordingly, the engineer has to

- anticipate carefully all possible environment situations (shape and velocity of obstacles, crossing humans, environment conditions),
- chose a suitable set of sensors (cameras, laser scanners, RGB-D devices)
- define an appropriate schedule for sensor polling, data filtering and processing.

The specific knowledge about environment conditions and tasks allows the development of highly optimized mobile systems. Sensors for instance can be adjusted precisely to each other in space and time. This allows a minimization of the response time, an increase of output quality, a maximization of the covered area or the limitation of the required processing performance.

This approach does not consider the information generated by external stationary sensors (building automation, monitoring cameras, sensors inside machines) or by other mobile entities (robots, smartphones, wearable sensors). Due to the fact, that the number of sensors increases in our homes or in industrial contexts, we need to overcome the "island perspective" based on local sensor outputs. Why should we detect the state of a door by laser scanner or camera information in an elaborate process when the door control system provides this information? Is it necessary to track a human by RGB-D measurements, if the position of their smartphone is available? This point of view addresses the paradigms of Cyber-Physical Systems (CPS) [1] and the Internet of Things [2]. Following these ideas, we no longer design a statically configured, local perception layer but integrate interfaces to aggregate external perceptions in order to improve task specific parameters adaptively. Following this approach, a mobile robot is equipped with a number of sensors to guarantee its basic functionality. Additional external sensors enable additional services or higher performance (higher speed, more precise localization, more efficient trajectories). In the end, we will be able to build cheaper robots with increased robustness in the context of changing environments and unintended tasks. These properties will meet the requirements of future scenarios like assistant robots in industrial applications or in the private sector much more than a static configuration.

1.2 Basic Concept

Replacing the static sensor configuration with an adaptive perception layer poses a number of challenging problems. First of all we need an abstraction mechanism for encapsulating individual sensor parameters, like measurement range, uncertainty level, update rate, and position. They are not known at design-time but necessary to interpret and evaluate the measurements at run-time. The literature offers different approaches ranging from uniform data types in communication protocols to electronic data sheets [3], [4]. Based on these representations, we are able to distinguish between relevant and non-relevant measurements, to evaluate measurement quality information and to synchronize the input. Additionally, the application has to be adapted and optimized with regard to the current number of data sets. Consequently, a Cyber-Physical System needs an additional component to apply these services. For this purpose we developed the concept of an Adaptive Sensing Controller (ASC).

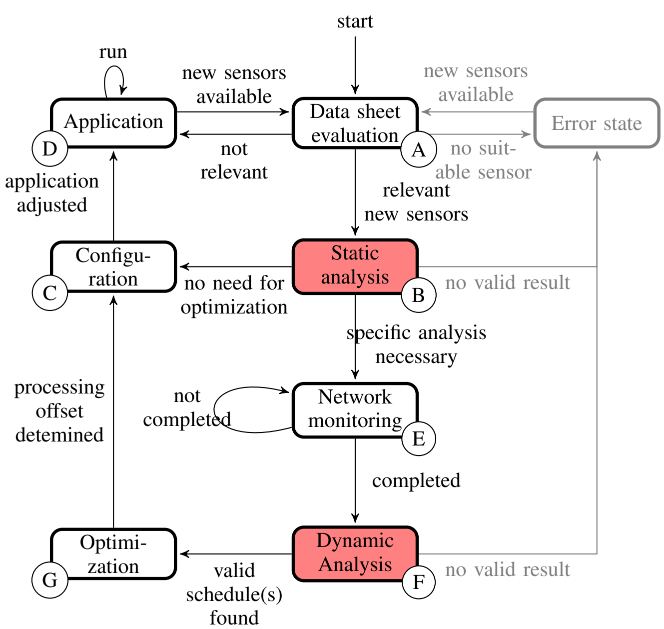

Figure 1: Online evaluation of the available sensor sets combining a static analysis and an optional optimization loop in the Adaptive Sensing Controller (ASC)

Fig. 1 illustrates its basic functions in a state machine. The application (D) announces the type and quantity of sensor data. Afterwards the ASC is started. It continuously monitors all available sensors, evaluates their data sheets in relation to the relevance and adjusts the application. One important aspect is the generation of an optimized schedule. Hence, the ASC compares the quality and quantity of requested sensor data in relation to existing measured data and prepares the application to process the current sensing configuration. In case of mismatches an error status occurs.

The electronic data sheet of all available sensing devices is analyzed in the first evaluation step (A). If a relevant sensor is detected, the static analysis will be started and calculates the matching level in relation to the minimum number of sensors per application cycle, maximum age, level of uncertainty, e.g. This is the worst case analysis that just uses the static parameters of the sensors and does not consider the variable properties (delays, phase and jitter) of the distributed scenario. If all demands are fulfilled, the configuration phase will be started. In case of a more challenging requirement set, we have to start a scenario specific, second level evaluation. The network monitoring component determines the network characteristics by evaluating the time series of measurement data (E). The results are included in the dynamic analysis (F) and the subsequent optimization (G). At this level, a more detailed temporal model is used to define an optimal schedule. Consequently it provides a more precise and adjusted result but requires more time to monitor the network and to analyze the specific sensor configuration.

This paper addresses one important aspect of the ASC: the optimization of the application schedule in relation to the sensor periods and phase shifts. It illustrates the two-stage evaluation process, the low level worst-case static analysis (B) and the dynamic analysis (F). The mathematical model for both components is described shortly. At the end, we illustrate the benefits using an exemplary scenario.

2. OFFSET ANALYSIS AND EVALUATION

2.1 General Problem - Phase Shift between Sensor Periods

The ASC optimizes the different time frames of the sensors related to the application periods. In statically configured systems these adjustments are done at design-time. For runtime solutions the reference literature describes several approaches.

- A static schedule is assumed in many control applications. The different sensor schedules are copied either by ignoring (in case of systems with a limited dynamic) or by synchronizing the measurements based on a mathematical model [5]. However, the estimation of in-time sensor data based on older measurements increases the level of uncertainty. With regard to adaptive control applications, a number of authors developed models that provide a flexible adaptation of the control period related to the sensor state [6].

- An interesting example of an adaptive Kalman-Filter is given in [7]. The authors adjust the period of the filter in relation to the connected sensors. An optimization of the measurement age is not applied.

- Some frameworks describe an adaptation of the sensor periods at run-time according to predefined patterns [8]. However, these approaches only consider single application scenarios and are not applicable to our distributed systems with multiple independent applications with different periods.

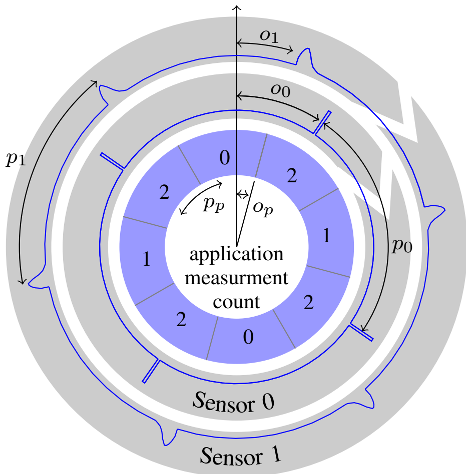

Figure 2: Arbitrary adjustment of 2 sensors and the fusion component. The diagram shows a super period containing 4 or 6 periods of each sensor. The application offset o_p determines the count of measurements per application cycle.

In this paper we only consider periodically working sensors and applications. Even with globally synchronized time within the distributed system, each device triggers its measurement generation or processing algorithm independently. An appropriate synchronization method was presented in [9]. The start-up time or phase shift of a sensing devices o_n, o_p (Fig. 2) cannot be controlled by the application, since there might be more than one application using the sensor.

We want to illustrate a set of metrics for the static and dynamic analysis focused on an adjustment application period offset o_p to sensor periods. At the end, the measurement count and age should be stabilized and the uncertainty level should be minimized. Fig. 2 illustrates the corresponding challenge in an abstract way. For the sake of simplicity, we only consider two sensors and one processing task at this point. The processing task merges the received measurements and calculates a combined result that is applied in an application. The circle illustrates the super-period of this example containing four periods p_0 from sensor 0 and six periods p_1 from sensor 1. The fusion task runs eight times during the super period.

The sensors generate one measurement per period, its propagation depends on internal and network delays. The occurrence is modeled using a probability density function, illustrated by the thin blue graphs. For this example we assume that both sensor devices are connected with different networks. Sensor 0 shows a tall window with uniform probability, while measurements from sensor 1 are delayed according to a Rayleigh distribution.

We suppose a constant phase shift between the two sensors $o_1 - o_2$ determined by the different start times of the devices. The ASC is only able to control the phase offset of the application (o_p) that is completely independent. Its definition affects the time adjustment of sensor and processing periods and configures the age and the number of the incoming measurements. The number of received measurements fluctuates between 2 and 0 per cycle in Fig. 2. During the application cycle and without a measurement, the application has to use estimated values. Accordingly, the quality of the environment representation decreases. If we suppose a slightly smaller offset of the application (o_p) and rotate the inner circle respectively, we can find an adjustment that guarantees at least one measurement per cycle. Additionally, we have to consider the age of each measurement. Due to the periodical execution of the application, an additional delay occurs. As you can see in Fig. 2, the measurements are received at different points in time during the application cycles. If we monitor highly dynamic processes, the measurement value represents a backward view, and its uncertainty increases besides the pure sensing noise.

Hence, the application defines a minimum measurement count and uncertainty. The ASC uses the data sheets of the available sensors and calculates an individual and common occurrence probability. If the minimal value covers the requirements, it is not necessary to run the dynamic analysis or the optimization. The definition of the application‘s phase offset is irrelevant in this case. Otherwise the specific offsets have to be involved in the dynamic analysis in order to calculate an optimized o_p.

2.2 Mathematical model

In contrast to a previous paper [10] we apply a mathematical model for the evaluation process. In this section we derive metrics for the static and dynamic analysis module. At this stage of the project we do not consider any additional delays. The maximum age of a measurement is determined by the length of the sensor period only.

In a single sensor scenario the whole system can be described by using just three variables: p_p processing period, p_s sensing period and a phase shift or offset o_s between them. The minimum and maximum number of measurements per application cycle is defined by

$$m_{max} = \left\lfloor \frac{p_p}{p_s} \right\rfloor + 1 \text{ and } m_{min} = \left\lfloor \frac{p_p}{p_s} \right\rfloor \tag{1}$$

The probability to meet the higher/lower number of measurements, is defined by

$$P(m_{max}) = \frac{p_p}{p_s} - \left\lfloor \frac{p_p}{p_s} \right\rfloor \text{ and} \tag{2}$$

$$P(m_{min}) = 1 - \frac{p_p}{p_s} + \left\lfloor \frac{p_p}{p_s} \right\rfloor \tag{3}$$

At this stage we are able to calculate both value pairs for a single sensor. If more than one sensor is available we have to calculate the convolution between all discrete distributions.

A second criterion of the analysis is the age of a measurement related to its processing time at the end of an application period. We have to determine the maximum value d_{max} out of a set of delays $d_1, \cdots, d_n$ with $n = n_p$. All values in this tupel are unique for a sensor period to application period configuration but the order is determined by the offset o. The difference between neighboring delay values is always defined by the greatest common divisor (gcd) of sensor and processing period. The offset o also shifts all delay values within this range $\mathrm{mod}(o_s, \gcd(p_s, p_p))$. Hence, the maximum and the minimum delay that may occur can be calculated by

$$a_{max} = p_s - \mathrm{mod}(o_s, \gcd(p_s, p_p))$$
$$a_{min} = \gcd(p_s, p_p) - \mathrm{mod}(o_s, \gcd(p_s, p_p)) \quad (4)$$

It has to be noted, that o_s is integrated in the following equations. The static analysis does not know the offset configuration and can only calculate a worst-case-assumption with

$$a_{max} \approx p_s - \gcd(p_s, p_p)$$
$$a_{min} \approx 0 \quad (5)$$

The dynamic analysis can access the network monitoring results and determine a specific result.

3. EXAMPLE

3.1 Scenario

To illustrate the impact of the approach, we consider a mobile robot scenario. In a previous European Research Project [1], our group build an experimental setup for automotive scenarios based on miniaturized cars. The arena has a size of 5x5m and shows a closed loop trajectory with a width of 80cm. A section is visible in Fig. 3. The cars are tracked using several cameras mounted on the ceiling of the testing ground. The image capturing and processing is executed on different embedded processors. All cameras provide a 2D position and orientation information.

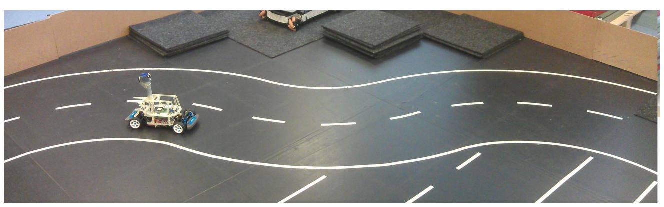

Figure 3: Experimental setup with a miniaturized car. It is tracked using several cameras to optimize the control algorithms.

For our exemplary scenario, we consider just two cameras of the same type to estimate the location and orientation of a car. In the context of this paper, they can be considered an intelligent environment, and all sensor characteristics are available from an electronic data sheet. As explained, the sensors work completely independent and neither sensor period nor phase shift can be manipulated by an application. The phase shift Δo between both sensor periods is arbitrarily defined by the manual activation. The camera systems are not synchronized and generate output data with a constant frame rate of $25Hz = 1/(40ms)$.

The position measurements x_n are disturbed by a Gaussian noise with a standard deviation of $\sigma_s = 2cm$. Additionally we have to consider the movement of the car after an image was taken. Consequently, the fusion algorithm synchronizes all received measurements $x_1, x_2, \cdots, x_k$ in a first step and predicts the related positions $\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_k$ at the end of the current processing period $k \cdot t_p$. For this purpose the algorithm evaluates the following equations:

$$\hat{x}_n(k \cdot t_p) = x_n(t) + f(k \cdot t_p - t) \quad (6)$$
$$\hat{\sigma}_n(k \cdot t_p) = \sigma_s + \sigma_v \cdot (k \cdot t_p - t) \quad (7)$$

The specific estimation function $f()$ is not important for the context of the paper. It provides the position progress depending on the current velocity. Additionally, we obtain an uncertainty information ($\hat{\sigma}_n$) depending on the time window of the prediction and on the sensor uncertainty. The uncertainty of the velocity was defined by $\sigma_v = 0.06cm/ms$. At the end of each fusion period the algorithm merges all position estimations $\hat{x}_n$ in a common result. Each estimation is weighted related to the reciprocal value of its uncertainty level.

$$\hat{x} = \sum_{n=i}^{i} \hat{\sigma_n}^2 \cdot \sum_{n=i}^{i} \frac{\hat{x}_n}{\hat{\sigma_n}^2} \quad (8)$$

$$\hat{\sigma}^2 = \frac{1}{\sum_{n=i}^{i} \frac{1}{\hat{\sigma_n}^2}} \quad (9)$$

The adjustment of the car control algorithm defines the period of the fusion to $p_p = 60ms$.

The main goal for the integration of the fusion algorithm was to reduce the motion blur and to provide a position information with a standard deviation $\hat{\sigma} < 2.5cm$. This value reflects the uncertainty level of the camera. In contrast, the worst case of a single measurement during a fusion period generates $\hat{\sigma} = 2cm + 0.08cm/ms \cdot 40ms = 5cm$ in worst case. Obviously, one measurement cannot fulfill the requested uncertainty. A second measurement reduces this value by half, a third one results in one third. If the static analysis is able to guarantee at least 3 sensor measurements per cycle, we do not need to run a complete dynamic evaluation of the sensing configuration

3.2 Static analysis

Both sensors work with a period of $40ms$. According to Equ. 1 each sensor generates at least $m_{max} = 1$ and at most $m_{min} = 2$ measurements per application cycle. Both counts occur with the same probability $P(m = m_{max}) = P(m = m_{max}) = 0.5$ based on Equ. 3. The common measurement count for both sensors can be calculated by a convolution operation

$$P_{com}(m) = \sum_{k=-\infty}^{k=\infty} P(k)P(m-k) \quad (10)$$

to $P_{com}(2) = 0.25$, $P_{com}(3) = 0.5$ and $P_{com}(4) = 0.25$. Hence, the two sensors generate at least 2 measurements per application cycle with a probability of 0.25. 3 measurements occur in 50 percent and 4 in 25 percent of the fusion periods. A dynamic evaluation of the offset configuration cannot be avoided due to the requested minimum number of 3 measurements per cycle. Hence, we have to evaluate the phase shift and determine an application offset for meeting the application requirements.

3.3 Dynamic analysis

Related to the configuration of our simplified model, the network monitoring determines only the phase shift of each

[1] Homepage on `http://www.karyon-project.eu/`

sensor $[o_1, o_2]$. In a more realistic case, we would have to monitor the timestamps of the sensor measurements and to estimate the corresponding delay distribution. Afterwards a search algorithm examines all sensor application adjustments $0 \leq o_p < p_p$, calculates the age of each measurement set and determines the expected uncertainty level. Fig. 4 illustrates the result for all possible phase sensor and application shifts. For simplicity the two sensor offsets are combined in a single $\Delta o = o_1 - o_2$ measuring the phase shift between the cameras. The colors illustrate the maximum uncertainties $\hat{\sigma}$ at the end of each fusion cycle calculated by Equ. 9. A certain phase configuration $[o_1, o_2]$ is represented by a horizontal line in the diagram. For $\Delta o = 10ms$ the corresponding uncertainty ranges from $2.2cm$ up to $3.2cm$. Consequently, an application offset of $o_p = \{1, 21, 41\}$ provides the best fusion results. In conjunction with the color bar it becomes clear that we can reach our uncertainty goal in all cases by choosing the fusion offset carefully.

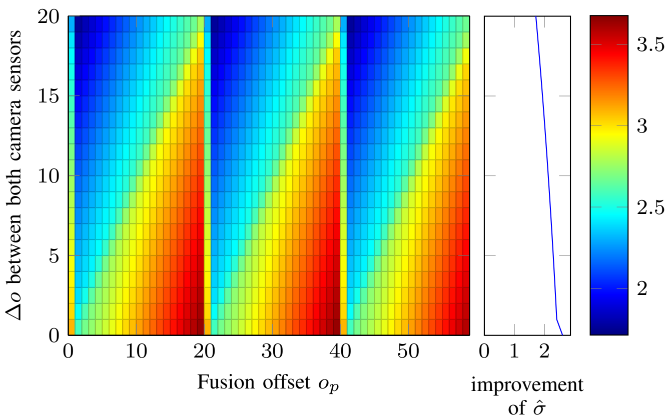

Figure 4: Resulting fusion uncertainty $max(\hat{\sigma})$ for different fusion offset configurations considering various phase shifts between the camera periods

Compared to the low level static analysis the dynamic analysis generates precise result and allows a fine grained evaluation between available and requested quality. The trade-off is given by the duration of the monitoring process and the delay caused by the analysis and optimization process. In realistic scenarios this may effect the applicability.

4. FUTURE WORK

The paper addresses on important task of the ASC, the estimation of the expected quality of the result and the corresponding monitoring and control of the sensor data aggregation. For this purpose, the ASC analysis the number and age of the incomming measurements on two levels and optimizes the schedule of the application. At the moment the mathematical model is simple, we do not consider additional delays or variable measurement uncertainties. As next steps, we want to develop a general solution for the phase optimization problem. It has to

- consider sub-goals (validity levels) and other metrics (precision, accuracy),
- find an abstract description of application quality demands,
- include an effective monitoring component for determining sensor phase shift $o_{\{1,2,\ldots,n\}}$,
- solving the optimization problem in an effective manner.

Finally, the approach needs to be validated using real hardware considering different network types and configurations.

Acknowledgment

This work has been partially supported by the DFG, through project INST 272/221-1 “Mobile Cooperative Robotics” (MoCoRo).

References

[1] J. Shi, J. Wan, H. Yan, and H. Suo, “A Survey of Cyber-Physical Systems,” in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, IEEE, 2011, pp. 1–6.

[2] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] S. Zug, M. Schulze, A. Dietrich, and J. Kaiser, “Programming abstractions and middleware for building control systems as networks of smart sensors and actuators,” in *Proc. of Emerging Technologies in Factory Automation (ETFA 2010)*, Bilbao, Spain, Sep. 2010.

[4] K. Lee, “IEEE 1451: A standard in support of smart transducer networking,” in *Proc. 17th IEEE Instrumentation and Measurement Technology Conference IMTC 2000*, vol. 2, 2000, 525–528 vol.2.

[5] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance.* Springer Verlag, 2006.

[6] K. G. Shin and C. L. Meissner, “Adaptation and graceful degradation of control system performance by task reallocation and period adjustment,” in *Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on*, IEEE, 1999, pp. 29–36.

[7] P. Ulbrich, F. Franzmann, C. Harkort, M. Hoffmann, T. Klaus, A. Rebhan, and W Schroder-Preikschat, “Taking control: modular and adaptive robotics process control systems,” in *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*, IEEE, 2012, pp. 55–60.

[8] Edward L. Waltz and James Llinas, *Multisensor Data Fusion.* Artech House, Inc., 1990, Available on http://portal.acm.org/citation.cfm?id=575126.

[9] C. Steup, S. Zug, J. Kaiser, and A. Breuhan, “Uncertainty Aware Hybrid Clock Synchronisation in Wireless Sensor Networks,” in *The 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, IARIA, Rome, Italy, Aug. 29, 2014, pp. 246–251.

[10] S. Zug, A. Dietrich, C. Steup, T. Brade, and T. Petig, “Phase optimization for control/fusion applications in dynamically composed sensor networks,” in *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, 2013, pp. 130–135.