Monitoring of Hybrid Systems using Behavioural System Specification

L. Zouaghi, A. Wagner and E. Badreddin

Institute for Computer Engineering, Automation Laboratory, University of Heidelberg
Mannheim, Germany
leila.zouaghi@ziti.uni-heidelberg.de
achim.wagner@ziti.uni-heidelberg.de
essameddin.badreddin@ziti.uni-heidelberg.de

Abstract:

This paper proposes hybrid monitoring of lower level actions for dependable (mobile) control systems including closed-loop control. We address a special class of hybrid dynamical systems, called switched systems. Here, the notion of switching is used to describe the state change when reaching a given bound or due to an external action that may be performed by a human operator. The approach is based on the consideration of both discrete and continuous aspects within a single unified framework by combining Petri-net and numerical filter theory. The model uses a specification of the desired system behaviour during the system design in order to deliver an on-line estimation of the current system state. The main goal of this approach is to get a generic formal method to create monitors for hybrid systems using the behavioural specification and a special class of mixed numerical and symbolic Petri-Net. To show the feasibility of the approach we apply it in the mobile robot area to an intelligent wheelchair. Such system can be best represented by hybrid models and presents a number of interesting new challenges for monitoring and diagnostic methodologies, which could be easily generalized to any autonomous mobile robot.

Keywords: monitoring, state estimation, Petri-nets, behaviour, specification, mobile robot.

1. INTRODUCTION

Computer-based systems such as control systems for automobile, aircraft, medical and intelligent mobile robots, are widely used in complex and safety critical applications which demand safe operation, high performance and very high dependability. All of these requirements have made system monitoring an inevitable component of system operations. Such safety critical systems are best described as hybrid systems because of interacting continuous-time dynamics and discrete-event dynamics. In the recent years, growing efforts have been made for diagnosing complex dynamical systems with hybrid (discrete/continuous) behaviours induced by the dynamics of the system and by external actions that may be performed by a human operator. The most current approaches used to deal with hybrid monitoring and fault diagnosis are represented as hybrid state estimation and they are based on the idea of predicting the outcomes of the system's actions by using some sort of predictive model and comparing the predicted outcomes with the observed ones. Some of these methods use a multi-model filtering such as a bank of kalman-filter (Veeraraghavan and Papanikolopoulos 2004) or a multi-model particle filtering (Funiak and Williams 2003). These methods track multiple hybrid estimates over time, but require many filters as much as the number of possible modes of the system. Other approaches are based on the use of automata (Hofbaur and Williams 2002) or on Bayesian nets (McIlraith 2000) linked to some numerical evolution models. Hybrid monitoring based on Petri-nets has been developed, for example, in Koutsoukos et al. (1998) where a timed Petri-net was used to model temporal discrete-event behaviour of (hybrid) systems. However, this kind of behaviour prediction does not really consider the hybrid system's nature consisting of a combination of discrete-event and continuous state evolution. The only existing hybrid estimation approach based on hybrid Petri-net and particle filter was used for the analysis of flight procedures and deals with situation monitoring (Lesire and Tessier 2005).The work focuses on the decisional autonomy and the monitoring of the high level actions i.e. producing a task plan and supervising its execution. However this kind of monitoring is not real-time critical and is based on qualitative properties and does not involve any continuous measure. The novelties of our work are (i) the use of an extended form of Petri-net and particle filter for the on-line monitoring of the lower levels of a system, which includes all the basic behaviours and the control loops (obstacle avoidance, motion control, etc.) (ii) the formulation of such a monitoring in a behavioural framework, where the behaviour of a system is considered as the main object of study (iii) the use of the behavioural specification of the system to determine if it behaves as desired. For the demonstration of the developed specification-based monitoring approach we apply it to an intelligent wheelchair, which has to reach specific goal points in an indoor environment.

2. PROBLEM FORMULATION

In order to monitor any dynamical system, in the meaning of comparing its expected behaviour, given by a model, with its actual behaviour, known through on-line observations, we need first to understand the behaviour of the system and to model, in a sufficiently abstract way, its interactions, which can be both discrete and continuous. Our problem formulation for monitoring is stated in a probabilistic behavioural framework using some basic definitions from Willems' behavioural systems theory, Petri-net theory and the Bayesian framework:

Definition: (Willems 1991) A dynamical system Σ is a triple $\Sigma = (T, W, B)$ with $T = \Re$ (continuous) or $T = \mathrm{N}$ (discrete) time axis, W the signal space, and $B \subseteq W^T$ the behaviour. $\Sigma_S = (T, W, X, B_S)$ is called the state-space (Input/state/output) representation of Σ, in which the behaviour B_S satisfies the axiom of state. The behaviour of Σ is now defined as $B = \{w \mid \exists x$ such that $(w, x) \in B_S\}$.

Based on this definition, we formulate the problem addressed in this paper as follows: given the dynamical state-space system $\Sigma_S = (T, W, X, B_S)$ of a dynamical system Σ and its specification $Spec$, it is required to built a monitor $\Sigma_{monit} = (T, W, X, B_{monit})$, which estimates the hybrid state X considering uncertainty in order to determine if the system Σ_S satisfies the given specification $Spec$. Throughout this paper, a unique product decomposition $W = U \times Y$ is considered, where U denotes the set of input signals, and Y the set of output signals. The solution approach for the problem statement will be based on the following steps (i) to construct a generic specification template $Spec' \subseteq Spec$ containing only the elements of $Spec$ that are considered essential for monitoring, such as the to-be-monitored variables, a given behaviour specification $B_{S,Spec}$ of B_S representing the set of acceptable signals $B_{S,Spec} \subseteq W^T$, the system noise constraints, accuracy constraints, etc. (ii) Embedding $Spec'$ into the monitor Σ_{monit} in order to estimate the system state and to compare it to the current state.

Here the problem of hybrid monitoring is represented as hybrid tracking and state estimation. We assume that Σ_S is realizable by a Petri-net and using $Spec'$ and the particle filter theory we develop a monitor $\Sigma_{monitor}$, which captures, in a single formalism, the discrete changes with the continuous changes.

The paper is organized as follows: In Section 3 the hybrid monitoring problem is stated in the behavioural framework and a generic monitor based on system specification is presented. In section 4 we apply the method on the mobile robotic area to an Intelligent Wheelchair in order to monitor its position and orientation change. Behavioural descriptions of each monitor and experimental results are presented. Some concluding remarks are offered in the final section.

3. MONITORING IN A BEHAVIOURAL FRAMEWORK

3.1 Behaviours and Petri-net

This section aims at introducing the monitor model through the behavioural formalism of system theory by describing it as a state-space dynamical system. In the context of such an input/state/output system-representation, the behaviour of the dynamical system $\Sigma_S = (T, W, X, B_S)$ is specified by behavioural equations. These are often differential or difference equations (Willems 1991 & Moor et al. 1998) consisting of a transition function $f : X \times U \to X$ that maps current states and inputs unto next states in the sense that B_S will be described in the discrete time domain (by sampling with a constant sampling period) by the following behavioural equations:

$$x_{c,k+1} = f_k(u_k, x_{c,k}) \quad , \tag{1}$$

$u \in U$ and $x_c \in \Re^{n_c}$ stand for the input signal and the continuous state respectively. The structure of the proposed hybrid monitor model Σ_{monit} is composed of a discrete model (Petri-net) and a continuous model (particle filter) that are in interaction. As depicted in Fig. 1, the discrete part receives the discrete input σ and the discrete output ψ of the dynamical system Σ_{system} together with the estimated continuous state $\hat{x}_c$ in order to determine the marking M of the Petri-net, which represents the discrete state x_d of the system. Then the continuous part uses the estimated discrete state together with the continuous input u and the continuous output y in order to determine the numerical state x_N. Such a state is represented by some places of the Petri net called numerical places P_N, which represent assumptions about the current behaviour of the system given by the difference equations (1). The places of the Petri net representing the configuration or the operation mode of the system are called symbolic places P_S or symbolic states x_S. The Petri-net marking M composed of numerical tokens evolving in x_N and symbolic tokens in x_S represents a hybrid state $x \in X$ of the system Σ_S. Let $B_{S,Spec}$ be a given specification of the external behaviour of Σ_s defined as:

$$B_{S,Spec} := \{w \in W^{\mathrm{N}} \mid \exists x \in X : \forall k \in \mathrm{N}: (x(k), w(k), x(k+1)) \in \xi \;\; and\, x(0) \in X_0' \}, \tag{2}$$

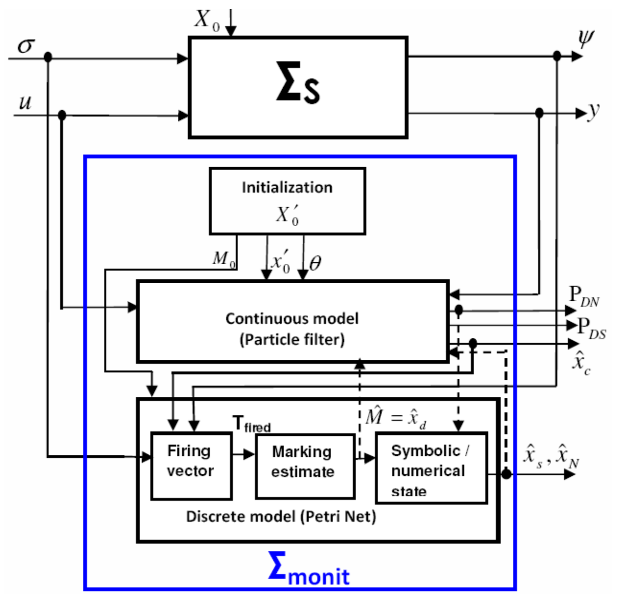

Fig. 1. Hybrid monitoring model

where $W, X, X_0' \subseteq X$, $\xi \subseteq X \times W \times X$ denote the external signal space, the hybrid state space, the set of initial conditions and the next state relation respectively. We assume that the Petri net $PN_S = (\mathrm{P}, \Gamma, \mathrm{Pr}\,e, Post, M_0)$ is a realization of $B_{S,Spec}$ by creating for every state $x \in X$ a place p(x), where $\mathrm{P} = \{p(x) : x \in X\}$ is the set of places partitioned into numerical places P_N and symbolic places P_s and for every action $e \in \xi$ from state x to x' creating a net transition $\tau(e)$, where $\Gamma = \{\tau(e) : e \in \xi\}$ is the set of transitions. The initial marking M_0 is matched with the initial state of Σ_s. Pre and $Post$ are the pre-incidence and the post-incidence matrices respectively, of dimension $|P| \times |\Gamma|$. The incidence matrix of the net is defined as: $C(p,\tau) = Post(p,\tau) - \mathrm{Pr}\,e(p,\tau)$ and represents the interaction between the discrete and continuous parts. The evolution of the marking from a given marking M_k at time k to a reachable marking M_{k+1} at the instant k+1 is obtained from:

$$M_{k+1} = M_k + C \cdot T_{fired,k}\,, \tag{3}$$

where $T_{fired,k}$ is the firing vectors at time k (Fig. 1). The state vector x is composed of discrete (binary) states x_d and continuous states x_c. The evolution of the continuous state x_c can be described by (1) and x_c is estimated using the particle filter (see Section 3.2). The evolution of the discrete state x_d is given by:

$$x_{d,k+1} = \delta(x_{d,k}, x_{c,k}, \sigma_k)\,, \tag{4}$$

with $x_d \in \{0,1\}^{n_d}$, σ is the discrete input of the system, x_c is the continuous state and δ is the function specifying the possible events.

3.2 Particle filter for the tracking of continuous states

The problem of tracking the continuous state of the system is formulated in terms of estimating the posterior probability density over the state space of the dynamic system Σ_s using a particle filter. The key idea of this technique is to represent probability densities by sets of samples known as particles with associated weights, that simulate the probabilistic model of the system. Detailed description of particle filtering methods for estimation of dynamical systems can be found in Arulampalam (2002). Using (1) we specify the time discrete evolution of the continuous state x_c, which is represented by a set of particles $\pi_{k+1|k}^{(i)}(p)$, meaning particle number i on the numerical place $p \in P_N$ at time k+1 knowing the observation at time k as follows:

$$\pi_{k+1|k}^{(i)}(p) = F(p, k, \pi_{k|k}) + v_k(p)\,, \tag{5}$$

with $\pi_{k|k} \in M_k(p)$ and v_k is the process noise. This step is called prediction step and computes the next state estimate from the current state. A correction step compares the actual observation with the prediction and corrects the predicted state using the observation model:

$$z_k = h(k, x_k) + \eta_k \quad , \tag{6}$$

where z_k is the observation at time k and η_k is the observation noise. The results of the correction step are respectively a probability distribution P_{DN} (as weighted particles) over the numerical state vector and a probability distribution P_{DS} over the symbolic state vector (Zouaghi et al. 2010).

3.3 Monitoring using embedded behavioural specification

Given is a system Σ_s and a subset of its specification $Spec$ denoted by $Spec'$, which is comprised of the essential elements for the state-estimation based monitoring listed below such as the specification of the behaviour $B_{S,Spec}$ described by behavioural equations, a specification of uncertainty in the model (as a noise) and in the variable given by ε as the maximal allowed deviation from a reference value, etc. We can construct a monitor Σ_{monit} as a Petri-net model corresponding to the behavioural description file depicted in Fig. 2. This behavioural description contains both the specification $Spec'$ and some necessary building elements for the monitor Σ_{monit} like the number of the particles θ and their initial values x_0', which can be defined a priori or generated with a random number generator. Fig. 2 shows an excerpt from the general template of the behavioural description file, which can be used for different systems by making a few small changes. This Behavioural file has the following structure:

1. The definition of the system state $x \in X$: the "Netparameters" represent the monitored numerical state vector x_N. The "Configurations" (i.e. the operation mode) represent the monitored symbolic state vector x_S.
2. The maximal allowed deviation ε representing the accuracy constraints over the continuous-valued state x_c.
3. Behaviour specification and operation modes: the numerical states "NUM" (numerical Petri net places) and their corresponding difference equations coefficients, specifying the behaviour B_s of Σ_s (e.g. the coefficients A an B of the difference equation ($x_{k+1} = A \cdot x_k + B \cdot u_k$). The symbolic states "SYM" represent the operation modes of the system (i.e. ON or OFF) or the environment states (i.e. areas where a vehicle navigates)
4. A list of all transitions numerical and symbolic, which represent the events occurring in the system. The input conditions associated to the numerical transitions can be simple conditions, which use comparative operators, including "<", ">", ">=", "<=" or "=" or combined with the logical operators "AND" or "OR". In case of a reference trajectory we use other kind of conditions called "Interval Condition". Such interval denoted $[I^-, I^+]$ can be represented as $V_{ref} + [-\varepsilon, \varepsilon]$, where V_{ref} is a reference value belonging to the reference trajectory and defined as

$$V_{ref} = \frac{I^- + I^+}{2} \;;\; \varepsilon = \frac{I^+ - I^-}{2} \quad . \tag{7}$$

5. The incidence matrix represents the model structure, which describes the interrelationship between the states and the evolution of the tokens.

6. The prediction noise from the motion model and the observation noise from the observation model.

7. The initial conditions: the number of tokens (particles and configurations), their initial values and the initial Marking denoting the initial state of the system.

```
>Netparameters number_of_param          ← xN
[param_1] [param_2] ...
>Configurations number_of_config        ← xS
[config_1] [config_2] ...
>Max_Deviation number_of_dev_param      ← ε
[Dev_param][dev_param=dev_value]
>Places number_of_places
[P0][NUM][A11 A12][B1]      } BS,Spec
[P1][NUM][A21 A22][B2]
...                                          Spec´
[OP_MODE_1][SYM][]
[OP_MODE_2][SYM][]
...
>Transitions number_of_transitions      ← u
[numerical_Trans1][NUM][T][param_1>value_a]
[numerical_Trans2][NUM][A][param_2>value_b & param_2<value_c]
[numerical_Trans3][NUM][O][param_3>value_d || param_3<value_e]
....
[OP_MODE_1][SYM][MODE = 0.0]  } σ
[OP_MODE_2][SYM][MODE = 1.0]
...                                                Σmonit
>Incidence matrix                       ← C
-1 1 0 -1 0
1 -1 -1 0 1
...
>Pred_noise [value_p_noise]             ← η
>Obs_noise [value_n_noise]              ← ν
>Marking number_of_particles
[P0][NUM][init_value]
[P0][NUM][init_value]
...                           } Initialization X0´
...
[OP_MODE_1][SYM][1.0]
```

Fig. 2. Behavioural description file containing system and monitor specification

4. MONITORING OF AUTONOMOUS ROBOTS

To show the feasibility of the approach we apply it to two different kinds of monitors on an Intelligent Wheelchair (IWC) as a special type of autonomous mobile robot. The model for the orientation monitoring uses a purely numerical Petri-net, while the model for the position monitoring is a numerical-symbolic Petri-net.

4.1 Wheelchair system description

The intelligent wheelchair (IWC) depicted in Fig. 6 is a conventional powered wheelchair with two motorized rear wheels and castors in front. It can move forward and backward, drive straight and turn left or right. It is equipped with different sensors: ultrasonic sensors arranged around the wheelchair providing collision avoidance functionality, incremental encoders on the wheel axes for the odometry and a gyro measuring the angular rate of the wheelchair orientation. The wheelchair's behaviour is produced using a nested control architecture (Badreddin 1989), which decomposes the system into behavioural-subsystems in a nested manner. In this paper we present only the monitor of the position-control level, which implements the controller for position and orientation and drives the robot along the desired trajectory.

Fig. 3. The wheelchair system

4.2 Kinematics Model of the Wheelchair

The wheelchair's pose at time k represents its state and is described by the vector $\vec{X}(k) = (x_{pos}(k), y_{pos}(k), \phi_{pos}(k))$ corresponding to two position coordinates $x_{pos}(k), y_{pos}(k)$ and the orientation $\phi_{pos}(k)$ transformed from the wheelchair coordinate to the world coordinate system. The state of this behavioural level is tracked with a particle filter using the motion model $\vec{X}_{k+1} = f(\vec{X}_k, u_k)$, which represents the behavioural model of Σ_s (1). The model defines the relationship between the new pose with respect to the current pose and to the input given by the odometry sensors on the wheelchair as follows:

$$\begin{aligned} \vec{X}_{k+1} &= f(\vec{X}_k, u_k) + v_k \\ &= \begin{bmatrix} x_{pos,k} + \Delta s_k \cdot \cos(\phi_{pos,k}) \\ y_{pos,k} + \Delta s_k \cdot \sin(\phi_{pos,k}) \\ \phi_{pos,k} + \Delta\phi_k \end{bmatrix} + v_k \end{aligned} \quad , \qquad (8)$$

where $u_k = (\Delta s, \Delta\phi)$ is the control input, Δs and $\Delta\phi$ are the displacement and the rotation of the wheelchair respectively during the same sample interval and v_k is a noise assumed to be zero mean Gaussian with covariance Q_k modelling the position uncertainty of the wheelchair.

4.3 Experiment and simulation results

a) Orientation monitoring

The IWC has to reach specific goal points in an indoor area. In order to make it drive a given path, a sub-goals trace has to be defined offline. The definition of such bypass-points will not be necessary if a path planner is available because it delivers on-line the sub-goals lying on the path. We suppose that no collision avoidance behaviour is available therefore safe trajectories are a-priori planned. A trajectory is then a combination of rotation und straight drive which can be predicted by monitoring the orientation change of the wheelchair based on the behavioural model of this level. The behaviour of the Wheelchair can be specified i.e. in the simple case with three states: „ driving straight", "Turn left" and "Turn right" (Fig. 4). We consider here only the forward motion. The places model the possible states of the wheelchair and the

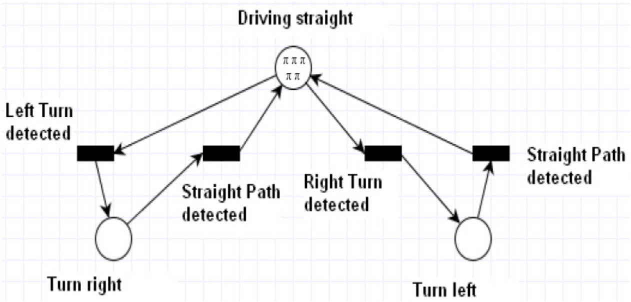

Fig. 4. The Petri-net model for the orientation monitoring

transitions model the state change. The behavioural description file of the model is depicted in Fig. 5. It contains only a numerical part, which uses the particle filter to estimate the current wheelchair state (orientation) and to describe how the wheelchair will actually behave.

```
>Netparameters 1
[DELTA_PHI]
>Configurations 0
>Max_Deviation 1
[E_PHI][ε=0.01]
>Places 3
[P0][NUM][1.0][1.0]
[P1][NUM][1.0][1.0]
[P2][NUM][1.0][1.0]
>Transitions 4
[T0][NUM][T][DELTA_PHI<-0.01]
[T1][NUM][A][DELTA_PHI>-0.01 DELTA_PHI<0.01]
[T2][NUM][T][DELTA_PHI>0.01]
[T3][NUM][A][DELTA_PHI<0.01 DELTA_PHI>-0.01]
>Incidence matrix
-1 1 0 0
1 -1 -1 1
0 0 1 -1
>Pred_noise [0.001]
>Obs_noise [0.07]
>Marking 20
[P0][NUM][0.0]
[P0][NUM][0.0]
........
........
```

Fig. 5. Behavioural description file for orientation monitoring

b) Position monitoring:

In order to bypass and to avoid obstacles we defined for the first part of the problem particular points called bypass points, which are described in a database and known only for the system. Now we introduce for the position monitoring a new kind of via-points, called Waypoints and which are surrounded by areas. The idea is as follows: the operator specifies offline the reference trajectory as a sequence of waypoints (WP) from a start point to an end point i.e. $WP_1 \rightarrow WP_2 \rightarrow WP_5$, which means that WP_1 must be reached before WP_2 and WP_2 before WP_5. Taking under account the maximal deviations ε_x and ε_y from the specification, the waypoints will be expanded automatically to waypoint-areas (9) to take into account the location uncertainty of the Wheelchair. The wheelchair needn't go exactly through the i-th waypoint $WP_i = (x_{ref,i}, y_{ref,i})$, which represents the middle point of the i-th area but it suffices to enter his correct area and to begin to adjust its orientation to the next sub-goal. The areas are defined as follows:

$$Area_i = \{x_{ref,i} - \varepsilon_x \leq x \leq x_{ref,i} + \varepsilon_x \,;\; y_{ref,i} - \varepsilon_y \leq y \leq y_{ref,i} + \varepsilon_y\} \quad (9)$$

The Petri-net model is extended to a symbolic part (Fig. 6) consisting of symbolic places, which represent the waypoint-areas. The events on the symbolic transitions update the symbolic marking of the Petri-net and indicate when navigating in an area and when going towards the next area.

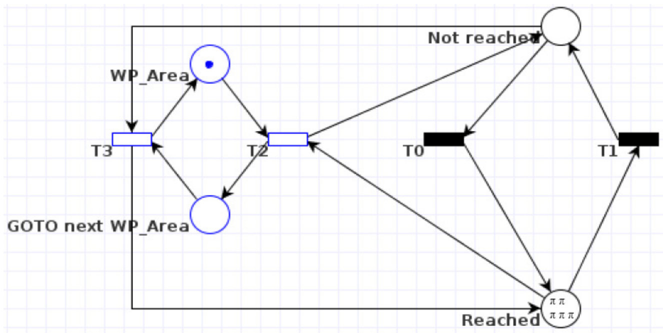

Fig. 6. The Petri-net model for the position monitoring

The numerical part of the Petri-net net tracks the position change of the wheelchair, which is modelled into the numerical states "reached" and "not reached" meaning an area is entered or not. On the numerical transitions are the intervals defined, indicating when the waypoint must be reached. The behavioural description file of the position monitor is depicted in Fig.7. The model contains a numerical

```
>Netparameters 2
[x]
[y]
>Configurations 1
[Area]
> Max_Deviations 2
[E_x][εx=1]
[E_y][εy=1]
>Places 4
[Not reached][NUM][1.0 0.0|0.0 1.0][0.0]
[Reached][NUM][1.0 0.0|0.0 1.0][0.0]
[WP_Area][SYM][]
[GOTO next WP_Area][SYM][]
>Transitions 4
[T0][NUM][V][A][A][x>=[3.5 5.5 8.5 ...] x<=[4.5 6.5 9.5 ...]
                 y>=[1.5 8.1 14.6 ...] y<=[2.5 9.1 15.6 ...]]
[T1][NUM][V][O][A][x<[3.5 5.5 8.5 ...] x>[4.5 6.5 9.5 ...]
                 y<[1.5 8.1 14.6 ...] y>[2.5 9.1 15.6 ...]]
[T2][SYM][Area=0.0]
[T3][SYM][Area=1.0]
>Incidence matrix
-1 1 1 -1
1 -1 -1 1
0 0 -1 1
0 0 1 -1
>Pred_noise [0.03]
>Obs_noise [0.07]
>Marking 21
[Reached][NUM][0.0 0.0]
[Reached][NUM][0.0 0.0]
[Reached][NUM][1.0 1.0]
..............
[WP_Area][SYM][1.0]
```

Fig. 7. Behavioural description file for position monitoring

part, which uses the particle filter to estimate the wheelchair position and a symbolic part, which indicates the current navigation area. The results of the estimation process are shown in Fig. 8. We can notice that the predicted behaviour (red trajectory) fits well with the observed one (blue trajectory). The estimation of the orientation change corresponding to this trajectory is shown in Fig. 9. The red points represent the observations every second and the blue crosses are the particles, representing the estimate of the orientation change. Negative values correspond to a right turn and positive values imply a left turn. The probabilities of the states are shown in the same plot (right axis). As shown the estimation smoothly fits the observations and corresponds to the correct discrete state (Driving straight, left turn or right turn). The dashed red circles show that also when two states take place successively in a short time the monitor is still able to detect the intermediate state i.e. a right turn (pink line) then a left turn (black line) with "Driving straight" (green line) as an intermediate state

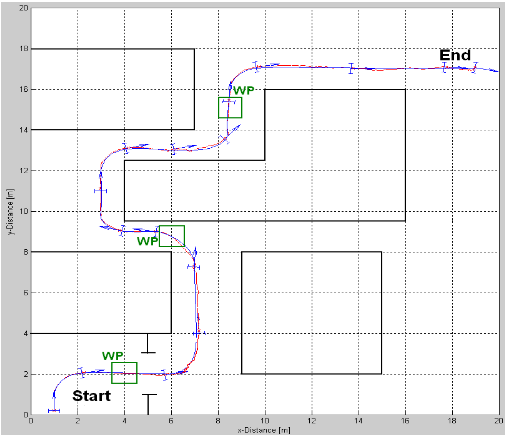

Fig. 8. The trajectory estimation of the IWC

(dashed circle on the left). We are also interested in a specific interval representing the state "Driving straight", which is usually symmetric with respect to 0 and representing the conditions associated to the numerical transitions of the Petri-net. The scaling of such interval (dashed black lines) is determined using ε (the maximal allowed deviation) from the specification, the reference value $v_{ref} = 0$ and the relations given in (7). If the real deviation of Σ_S is larger than specified then this leads to violation of the specification as shown in Fig. 10. The monitor is still able to track correctly

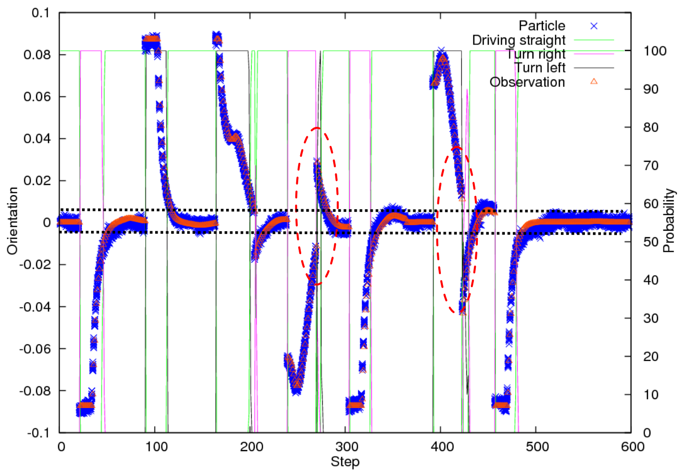

Fig. 9. Estimation results considering the specified deviation.

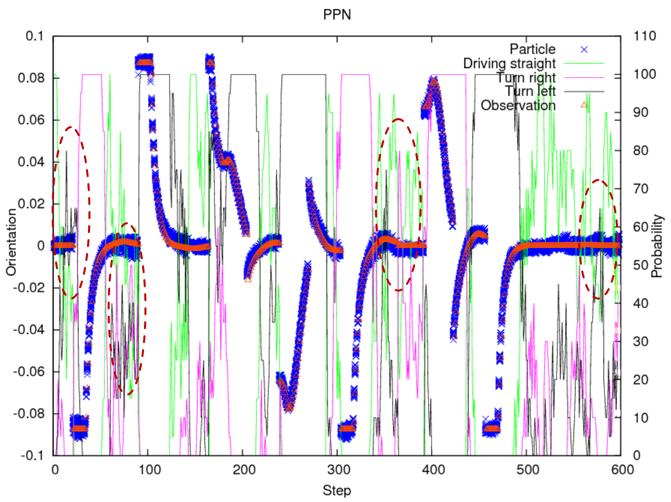

Fig. 10. Results by system deviation larger than specified.

the system state but with a higher probability for estimating wrong states (dashed circles).

5. CONCLUSIONS

This paper considers the hybrid monitoring approach for mobile robots based on Particle Petri-net in a behavioural framework. The approach enables (i) the modelling of the system's intended behaviour, which have to be specified correctly considering uncertainty and interactions between continuous and discrete dynamics (ii) the development of a generic description of Σ_{monit} from the behavioural specification of the system, which can decreases the effort to create a monitor during the system design.

ACKNOWLEDGMENT

This work is supported by the Ministry for Science, Research and Art, Baden-Württemberg in the framework of the ECOMODIS project. The authors would also like to thank M. Platho and A. Alexopoulos for the assistance in the implementation of the monitoring framework and Petri-net simulator.

REFERENCES

Arulampalam M., Maskell S., Gordon N., Clapp T.(2002), "*A Tutorial on Particle Filters for Online Nonlinear/Non- Gaussian Bayesian Tracking*" IEEE Trans. on Signal Processing, Volume 50, Number 2, pp 174-189

Badreddin E. (1989): "*Recursive Control Structure for Mobile Robots*", International Conf. on Intelligent Autonomous Systems (IAS.2), Amsterdam, pp.11-14.

Funiak, S., and Williams, B. 2003. *Multi-modal particle filtering for hybrid systems with autonomous mode transitions.* In Proceedings of SafeProcess.

Hofbaur M. and Williams B., "Mode estimation of probabilistic hybrid system," in Hybrid Systems: Computation and Control, edited by C. Tomlin and M. Greenstreet, vol. 2289 of Lecture Notes in Computer Science, Springer, Stanford, CA, March 2002.

Koutsoukos X.D., He K.X., Lemmon M.D., and Antsaklis P.J. (1998). *Timed Petri nets in hybrid systems: Stability and supervisory control*. Journal of Discrete Event Dynamic Systems: Theory and Applications, 8(2):137–173.

Lesire C., Tessier C. (2005). *Particle Petri nets for aircraft procedure monitoring under uncertainty.* In: ATPN, Miami, FL, USA.

McIlraith S. (2000). *Diagnosing hybrid systems: a bayesian model selection problem*. In Proceedings of the 11th International Workshop on Principles of Diagnosis (DX'2000).

Moor, T., Raisch, J., and O'Young, S. D. (1998). *Supervisory control of hybrid systems via l-complete approximations*. In Giua, A., Smedinga, R. and Spathopoulos, M., editors, Proc. WODES'98– International Workshop on Discrete Event Systems, IEEE, pp. 426-431.

Veeraraghavan H. and Papanikolopoulos N. (2004). *Combining multiple tracking modalities for vehicle tracking in traffic intersections.* In: ICRA. New Orleans, USA.

Willems, J. (1991). *Paradigms and puzzles in the theory of dynamical systems*. Automatic Control, IEEE Transactions on, 36(3):259–294.

Zouaghi L., Wagner A. and Badreddin E. (2010). *Hybrid, Recursive, Nested Monitoring of Control Systems Using Petri Nets and Particle Filters*. Proceedings of the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2010, Chicago, IL USA.