UNIVERSITÄT
HEIDELBERG

International Workshop on the
Design of Dependable Critical Systems

ECOMODIS

Proceedings

of the International Workshop on the Design of Dependable Critical Systems "Hardware, Software, and Human Factors in Dependable System Design"

DDCS 2009

September 15, 2009
Hamburg, Germany

In the framework of
The 28th International Conference on
Computer Safety, Reliability and Security
SAFECOMP 2009

Edited by

Achim Wagner[1], Meike Jipp[1], Colin Atkinson[2] and Essameddin Badreddin[1]

[1] Automation Laboratory, Institute of Computer Engineering,
University of Heidelberg
[2] Chair of Software Engineering, University of Mannheim

ziti

An Integrated Monitor-Diagnosis-Reconfiguration Scheme for (Semi-) Autonomous Mobile Systems

Yi Luo, Achim Wagner, Leila Zouaghi, Essameddin Badreddin

Automation Laboratory, University of Heidelberg, Germany
{yi.luo, achim.wagner, leila.zouaghi, badreddin}@ziti.uni-heidelberg.de

Abstract. A nested monitoring, diagnosis and reconfiguration (MDR) scheme is proposed for a Recursive Nested Behavior based Control structure (RNBC) constituting a generic system architecture for (semi-) autonomous mobile systems. Each behavior layer within the RNBC structure is associated with a MDR schema, which is responsible to ensure the dependability of every single layer. An online dependability measurement and diagnosis procedure is integrated into monitor and diagnosis blocks under consideration of performance and safety acceptability factors. The reconfiguration blocks within the MDR-scheme switch from components with unacceptable behavior to redundant components, which may have degraded performance but more robust and safe behavior. The MDR blocks at each layer are nested through unified interfaces in order to utilize the distributed modeling of system behavior and to facilitate the system design and implementation process. In a small case study the MDR scheme is demonstrated for an assistant wheelchair on the body velocity control and axis velocity control levels. Simulation results show the feasibility and effectiveness of the approach.

Keywords: Dependability, autonomous mobile systems, monitoring, diagnosis, reconfiguration.

1 Introduction

In (semi-) autonomous mobile applications, the primary objective to use fault detection and diagnosis (FDD) and fault tolerant control (FTC) techniques is to increase system dependability. A unified FDD/FTC framework that adapt to behavior-based architecture is required to assist system development. Some research projects [1][2] have developed layered fault tolerant control architecture for behavior-based mobile systems. However, finding novel control structures and design methods which are better applicable to engineering applications are still important research questions in the field of fault tolerance [3][4].

This work proposes a nested monitoring, diagnosis and reconfiguration scheme, named as MDR scheme, which is designed for the Recursive Nested Behavior-based Control (RNBC) structure [5]. Fault modeling and dependability concepts are adapted from [6] and [7]. In contrast to binary fault modeling the dependability concept is based on the behavior description of the system and its components. Dependability properties are related to the deviation of the actual system behavior from the desired behavior and to the distance of critical system states from safety boundaries. The desired behavior can be described in the form of a reference output signal (reference mission), which may be generated by a reference model in response of a system input trajectory. The deviation of the actual system output from the reference output is monitored by the corresponding monitoring component. If a state-space reference model is available, the monitoring component may be realized in form of a state observer, which estimates the internal system states besides the next predicted output value. The monitoring component outputs the deviation signals (residuals) and the distance of critical states from their limits. In case of black box modeling all critical states must be visible as external signals. The external signals will be fed to a diagnosis component, which assesses the acceptability of the retrieved value (s. example below). Depending on the result the system is reconfigured using a reconfiguration component. Here a hierarchical monitoring, diagnosis and reconfiguration (MDR) scheme is proposed.

2 Proposed MDR Concept

The MDR scheme is integrated in the Recursive Nested-Behavior-Based Control (RNBC) structure consisting of a number of layers, which are recursively connected to each other [5]. Each layer in the RNBC structure hosts a number of components and corresponding dynamic behaviors. The behaviors can be uniformly described as signals, which flow between the layers, regardless their type of implementations (e.g. hardware or software). A single MDR block ensemble is locally associated with a single behavior layer and responsible to keep the deviation from the specified behavior in an acceptable level.

Figure 1 shows two behavior layers of the RNBC structure, each of which containing a MDR scheme besides the functional components. Monitor, diagnosis and reconfiguration are the three components under consideration. The working principle of them will be explained in the following, using an exemplary modeling approach, i.e. all layers are described as linear time-invariant systems with time-continuous dynamics.

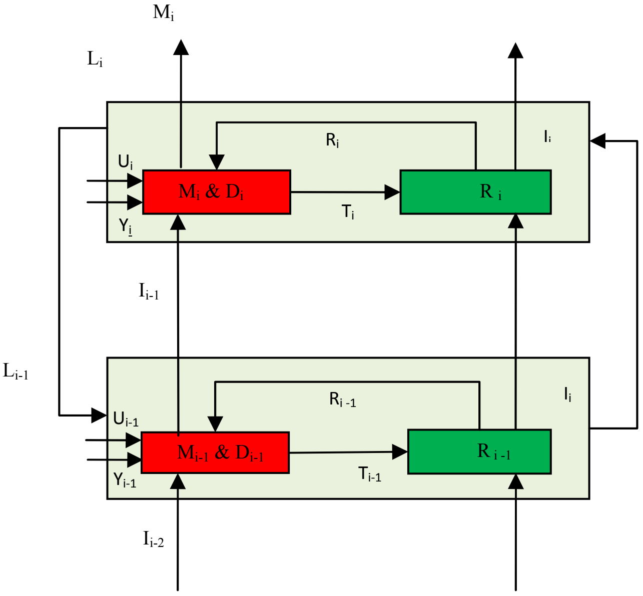

Fig. 1. Monitoring (M) – Diagnosis (D) – Reconfiguration (R) scheme integrated into the RNBC structure, exemplary shown for two behavior layers

2.1 Monitor Block and Diagnosis Block

The aim of the monitor block is to calculate the behavior deviation and the safety margin. Inputs for monitor M_i are: measured (u_i, y_i) of the ith layer, lower monitor status information I_{i-1}, and reconfiguration information R_i to indicate the status of the reconfiguration process and therefore to update the current reference model. A reference model, e.g. using a transfer function, which describes the nominal behavior of the considered layer, is required. The model is used to determine, for a given input, the reference output y_{ref}. The instantaneous deviation from the reference behavior is given by the residual (see also Fig. 2)

$$\boldsymbol{\varepsilon}_P(t) = \left|\mathbf{y}(t) - \mathbf{y}^{ref}(t)\right| \quad \textbf{(1)}$$

The residual is a basic ingredient for a normalized performance acceptability function

$$A_P(t) = 1 - \frac{\boldsymbol{\varepsilon}_P(t)}{E_p} \quad \textbf{(2)}$$

yielding a value range [0, 1] and indicating, how acceptable the system's (component's) behavior is in comparison to a maximum allowed output deviation E_P.

The definition of a safety acceptability function is also based on a behavioral description. Therefore, the concept a dynamic safety margin [6][8] has been adopted. In [8] safety boundaries for a state space model and a dynamic safety margin, which is the minimum distance from these boundaries, have been defined. In contrast to the original definition, here, the safety boundaries are related to the output signal, which

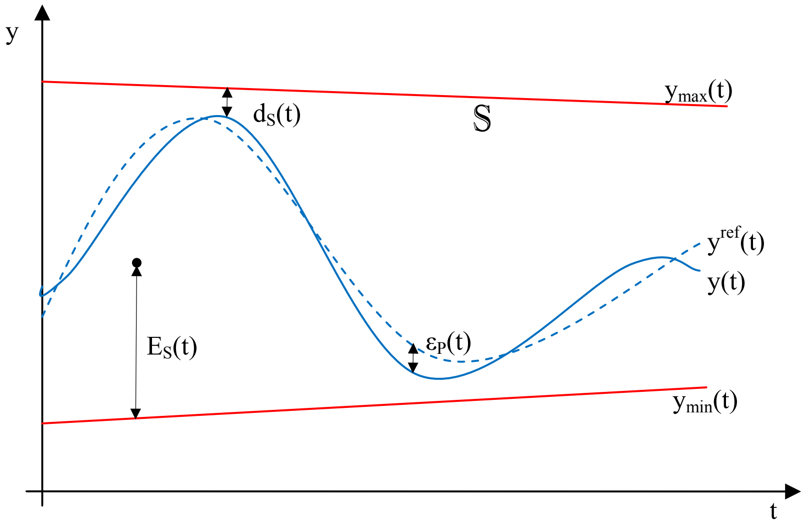

Fig. 2. Safety boundary and dynamic system trajectory.

is equivalent in the case of having all internal critical dynamic states available as system outputs.

For the given input u(t), there is a range [y_{min}, y_{max}] for the output y(t), where the system is considered to be in a safe condition. Now let

$$d_s(t) = \begin{cases} \min(((y(t) - y_{\min}), (y_{\max} - y(t))), y(t) \in [y_{\min}, y_{\max}] \\ 0, y(t) \notin [y_{\min}, y_{\max}] \end{cases} \tag{3}$$

be the distance to the safety boundary (fig. 2) and

$$E_S(t) = \frac{1}{2}(y_{\max} - y_{\min}) \tag{4}$$

the centre point of unsafe region at time t. Now, we can define the safety acceptability function

$$A_S(t) = \frac{d_s(t)}{E_S(t)} \tag{5}$$

with $A_S(t) \in [0,1]$ reflecting the system (component) safety level with respect to the maximum possible distance to the safety boundary $y(t) = E_S(t)$.

The total acceptability is the weighted sum of all acceptability terms

$$A_{TOT}(t) = a_P A_P(t) + a_S A_S(t), \tag{6}$$

which is a function of time and which reflects the coincidence of the actual system behavior with the specified behavior. According to [6], the integration of the acceptability values over the system's mission trajectory leads to a unique overall dependability measure. In this paper the instantaneous total acceptability function is used to decide, if the system yields an acceptance level A^* or not. If $A < A^*$, a system reconfiguration is enabled.

2.3 Reconfiguration Block

There are basically two questions, which must be answered before a system reconfiguration can be performed: 1. What configuration should the system have after the reconfiguration, 2. How can the system be brought to the new configuration (especially how does the system behave during the transient phase).
The question, what new configuration shall be used can be answered as follows:

Offline Design: Each behavior layer contains a nominal components and redundant components. Both are designed and tested offline. E.g. the nominal component is designed to deliver better performance while the redundant component is simple, well understood, and more robust against faults. Thus, for each component the (average) acceptability value for a set of predefined typical mission trajectories can be measured during system test. During operation of the system the best component (with the highest acceptability level) is selected. The component (or even a complete layer) under consideration is replaced by switching if the acceptability level drops under the level of the next best component. It is required that all possible combinations of components behave stable. The offline design method proposed is in contrast to online design methods, where the complete system (structure and parameters) is rebuild according to the instantaneous system constraints.

The second question cannot be answered so easily, if the system can be switched forward and switched back between different (at least two) configuration, since the system may behave unstable even in the case, when the single configuration themselves are stable. Therefore, we assume here one single transient from an undependable configuration to a new dependable configuration.

Online Switching: By default, all nominal components are supposed to be "normal". The switching is enabled only after the switching condition (enable signal from M&D blocks active) is fulfilled. When the reconfiguration is enabled, the reconfiguration block checks the configuration R_i from lower layers and it checks then the stability of the redundant component in the loop with the lower layers. When the stability condition is fulfilled the switching process will start. If the redundant component is already in operation and detected to be failed, the whole system will be brought in a fail-safe condition.

3 Application of the MDR Concept to an Intelligent Wheelchair System

In this section, a small application scenario is proposed illustrating the concept of the MDR scheme. Therefore, the three lower levels of a human-assisting "intelligent" wheelchair control system are considered.

Figure 3 illustrates the MDR scheme for the velocity controller in a wheelchair system. Layer L1 consists of the axes-level velocity controller, the actuators and some data processing blocks. Layer L2 contains the body velocity controller. It gets the reference velocity $[x', \theta']^{ref\ 2}$ from layer L3, compares it with the measured velocity $[x', \theta']^{m\ 2}$ from gyroscops and encoders and generates a control signal, which is the reference velocity $[x', \theta']^{ref\ 1}$ for next layer. In this example, the primary PID (proportional-integral-derivative) controller is used as nominal controller. The secondary PI controller has lower performance but is simpler and more reliable. The secondary component is running in parallel with the nominal component (hot standby). Thus an initialization period and a long term transient phase can be avoided.

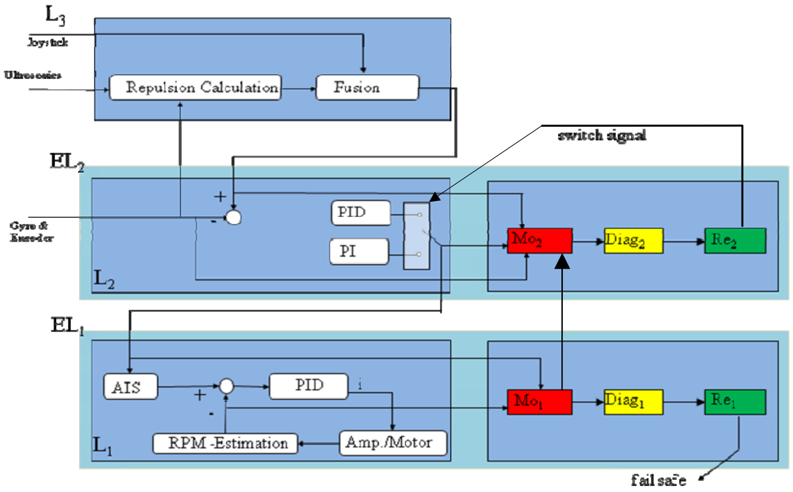

Fig. 3. Application example: Three lower layers of an intelligent wheelchair system

Parameters of PID/PI controllers and MDR are given in Table 1. These parameters comply with manufacturer and empirical data so that future implementation can be made based on them.

Table 1. System parameters for the body velocity control level.

Components	Parameters	Value
PID Controller	$K_{p,trans}, K_{i,trans}, K_{d,trans}$	1.33, 1.11, 0.37
	$K_{p,rot}, K_{i,rot}, K_{d,rot}$	1.6, 1.33, 0.53
PI Controller	$K_{p,trans}, K_{i,trans}$	1.0, 0.5
	$K_{p,rot}, K_{i,rot}$	1.2, 0.6
MDR	E_S, E_P	$[6, 6]^T$, $[5, 5]^T$
	a_S, a_P	0.5, 0.5
	A^*	0.75

Simulation results of the developed MDR mechanism using the model above are shown in Figure 4 a, b. A fault in layer L2 is emulated by injecting a 1 second output delay in the nominal PID controller. Figure 4.a shows the L2 behavior in 3 cases. The Blue line corresponds to the faultless case, the red line denotes the faulty case without MDR scheme and the green line denotes faulty case with MDR scheme. Figure 4.b shows the time-dependent acceptability level during a mission of 100 seconds. As the desired acceptability level is 0.75, the behavior switching happens at t = 0. It can be observed that the MDR mechanism has recovered the behavior to an acceptable level by switching to the redundant component upon failure detection.

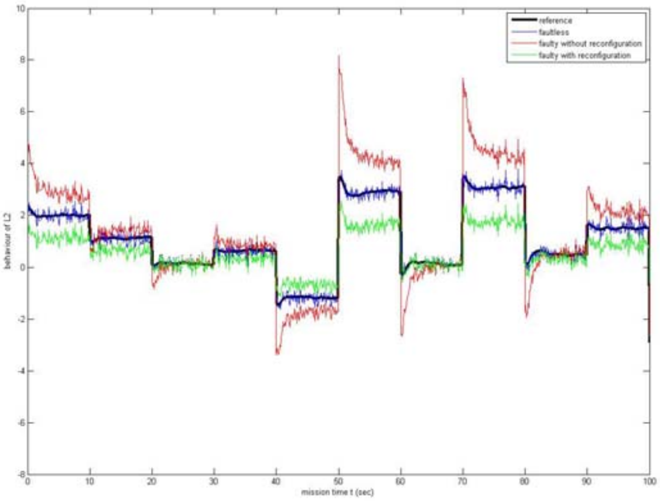

Fig. 4 a. L2 translative velocity behavior in faultless, faulty (no reconfiguration) and faulty (with reconfiguration) cases

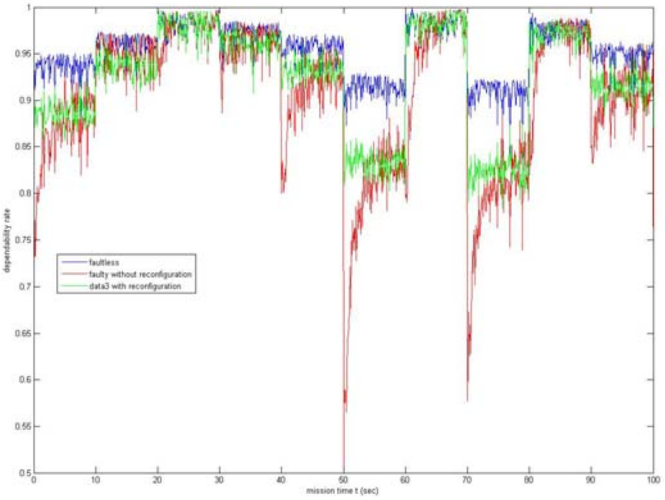

Fig. 4 b. L2 acceptability level in faultless, faulty (without MDR) and faulty (with MDR) cases

4 Conclusions and Future Research

In this paper, a monitor-diagnosis-reconfiguration scheme for autonomous and semi-autonomous systems is proposed. A Model-based monitoring and multiple-controllers online switching approach is realized and demonstrated within a realistic simulation example. Dynamic behavior acceptability improvement as reconfiguration goal is carried out. As a single behavior, the body velocity controller of a wheelchair system, was integrated together MDR within the proposed architecture. The simulation results show the feasibility of the proposed MDR scheme in terms of keeping the behavior of components and system layers within an acceptable performance and safety. In future research, the MDR scheme will be implemented into a real-time control system.

References

1. Ferrell, C.: Failure Recognition and Fault Tolerance of an Autonomous Robot, Adaptive Behavior, vol. 2, no. 4, pp. 375-398 (1994).
2. Visinsky, M. L., Cavallaro, J.R., and Walker, I.D.: A Dynamic Fault Tolerance Framework for Remote Robots, IEEE Transactions on Robotics and Automation, vol. 11, no. 4, pp. 477-490 (1995).
3. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems, Annual Reviews in Control Volume 32, Issue 2, Pages 229-252 (2008).
4. Duan, Z.H., Cai, Z., Yu, Z.: Fault Diagnosis and Fault Tolerant Control for Wheeled Mobile Robots under Unknown Environments: A Survey. in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp 3428 – 3433 (2005).
5. Badreddin, E.: Recursive Nested Behavior Control Structure for Mobile Robots, International Conference on Intelligent Autonomous Systems 2, (1989).
6. Wagner, A., Atkinson, C., Badreddin, E.: Towards a Practical, Unified Dependability Measure for Dynamic Systems, in Proc. of the International Workshop on the Design of Dependable Critical Systems, Hamburg, Germany, Sept. 15, (2009).
7. Rüdiger, J., Wagner A., Badreddin E.: Behavior Based Definition of Dependability for Autonomous Mobile Systems, in Proc. of the European Control Conference 2007, Kos, Greece, July 2-5, 2007, WeD11.4, (2007).
8. Abdel-Geliel, M., Badreddin, E., Gambier, A.: Application of Dynamic Safety Margin in Robust Fault Detection and Fault Tolerant Control, IEEE International conference on Control Applications, October 4-6, (2006).