Söffker, D.: Modeling of the Knowledge-Based 'Intelligent System'-Environment Interaction: Part I: Description and Application to Human-Machine Interaction, Part II: Systemtheoretic Aspects leading to a New Type of Autonomous Systems. IEEE Int. Conf. on Systems, Man, and Cybernetics, Tucson AZ, USA, October 7-10, 2001, 10 pages.

MODELING OF THE KNOWLEDGE-BASED 'INTELLIGENT SYSTEM'-ENVIRONMENT INTERACTION: DESCRIPTION, APPLICATION TO HUMAN-MACHINE INTERACTION AND SYSTEM-THEORETIC ASPECTS LEADING TO A NEW TYPE OF AUTONOMOUS SYSTEMS

DIRK SOEFFKER
Safety Control Engineering
University of Wuppertal, Wuppertal, D-42097 Wuppertal

Abstract

The modeling of the Human-Machine-Interaction (HMI) gives ideas to transfer the understanding of human control and to apply the developed modeling technique to autonomous technical systems, like mobile robots. The task of this new kind of intelligent control is to respond autonomously and problem-equivalent to complex situations primarily specified in words. The paper describes the modeling technique. Modeling the HMI AI-like terms are used like situation, operator etc. to define the system to be considered. It will be shown in which way this modeling methodology is suitable to model human acting, planning, learning and also describing human errors and is able to design an advanced autonomous system.

Keywords

Human-Machine-Interaction, Modeling, Situation Calculus, AI.

1 Introduction

Whereas classic control schemes typically address issues relating to speed, accuracy and other (low-level) problems typically to physical oriented technical tasks, more complete theoretical models of system control or interaction behavior often are quite complex and unwieldy in unknown environments or situations.

In the sixties and seventies the human-control behavior was examinated for stimulus-response tasks, e.f. describing the time behavior of human driving etc., e.g. [8]. In the nineties the Human-Machine-Interaction itself has been focused more intensively. An actual overview to the developed approaches is given by Cacciabue [1]. Different research directions have been established, which are oriented between Artificial Intelligence (AI) approaches and phenomenological macro cognition oriented engineering approaches [1]. In [10,11] a modeling approach of human interaction with formalizable technical environments is developed. Core of the work is a specified Situation-Operator model (SOM). In contrast to known procedures [3,5,7] the developed approach is neither based on exact temporal logic nor assumes a mathematical perfect understanding of context structures. Gigerenzer and Goldstein [2] show that unsatisfying classical norms of rational inference fast and frugal algorithms can lead to effective rationality. This includes the possibility working with non-perfect algorithms to imitate aspects of human learning. The idea is to transfer the engineering oriented approach [11] to technical systems to give them some kind of memory and internal organizing features for autonomous interaction capabilities. In the sequel the developed SOM-technique is used to describe the 'human controler' (HC) as well as an autonomous system (AS) and will be summarized as intelligent system (IS).

2 From Scenes and Actions to Situations and Operators

Core of the approach is the assumption that changes of the considered parts of the real world (RW) are understood as a sequence of effects described by the items scenes and actions [10,11]. In the proposed approach the definition of the items scenes and actions are coordinated in a double win. They are related to each other and they relate the assumed structure of the RW to the structure of the database -called mental model - of IS. ISs are included in the real world (RW). While ISs are interacting with the RW, they can change the RW. Depending on their principal sensor inputs, their natural (HC) or technical (AS) perceptions, and on the related knowledge base, the ISs adapt and learn only parts or aspects of the RW. These parts are modeled using the developed situation and operator calculus. The describable part of RW is called a system.

The item situation, which is (in contrast to [4]) a time-fixed, system-, and problem equivalent one, is used describing the system structure (as a part of the RW). Here only the logical structure of the 3D-space, time and functional-oriented connections are of interest. The item operator is used to model effects / actions changing scenes (modeled as situations) in time. The situation *S* consists of characteristics *C* and a set of relations *R*. The characteristics are linguistic terms describing facts (as qualities). This will include physical, informational, functional, and logical connections. To describe the relations *ri* known problem related modeling techniques, like ODEs, DAEs, Algorithms or even graphical illustrations (e.g. Petri-nets) can be used.

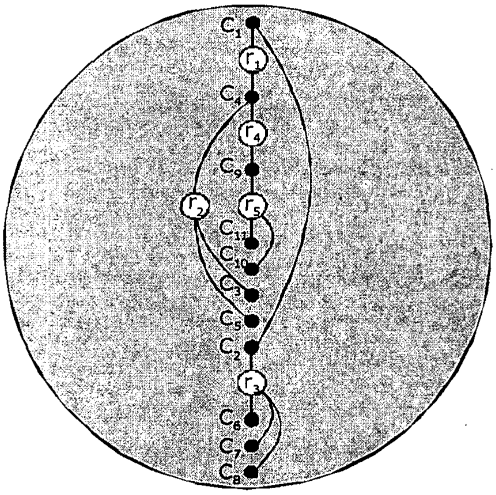


Fig. 1:Structure of the proposed item Situation *S* (Detailed example: Rail-wheel contact modeling)

The SOM-approach only gives the frame to model the structure of the relations, and therefore maps the 'reality' into a structural framework. This is useful describing problems, where the system structure is complex and can not be modeled using single approaches. This is especially useful to describe interactions between IS and its environment.

The introduced item characteristic *C* also includes the possibility of representing time-dependent parameters *P*. The set of relations *R* (of *C*s) fixes the structure of the considered scene of the world modeled as situation *S*. The introduced situation concept consequently allows the integration of different types of engineering-like descriptions. As an example in figure 1 the complex physical situation of the rail-wheel physics and the dynamic interaction and the structure of the related modeling is given. The illustration shows the dependencies of the considered mapping from the real world problem to the engineering oriented modeling using ODEs and case-depending algorithms into a qualitatively modeled and graphical illustrated network.

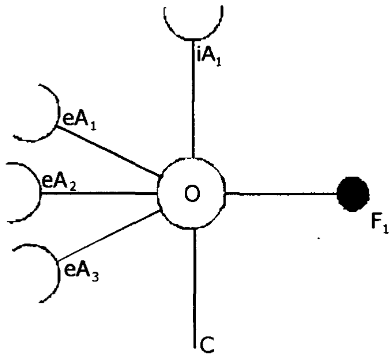
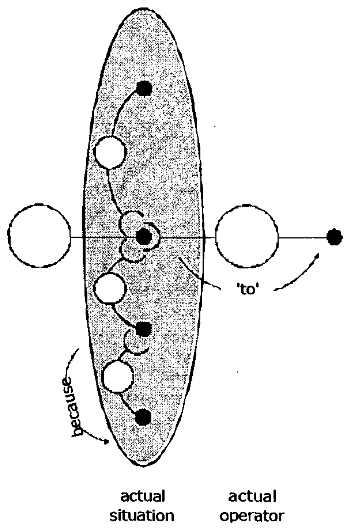


Fig. 2:Structure of the proposed item operator O

(principal example)



For this example shown in fig. 1 the relation *r1* connects *C1* and *C4*, which represents the tangential

Fig. 3: Conncection between situation and operator

contact force *(C1)* and the normal force *N* *(C4)*. The detailed modeling is given in [9]. For this purposes it should be noted, that the item situation is able to represent a qualitative modeling approach. Details of the mentioned problem can be found in [9]. The illustrated item operator is used for the modeling of a) internal (passive) connections of situations (cf. fig. 1) and b) changes between situations.

The operator O (cf. figure 2) is understood and modeled from a functional point of view: the operator is an information-theoretic term which is defined by his function F (as the output) and the related necessary assumptions. Here explicit and implicit assumptions eA, iA are distinguished. F will only be realized, if the explicit assumptions eA are fulfilled. The iA includes the constraints between eA and F of the operator. The eA are of the same quality as the characteristics C of S. For the internal structure of the operator other descriptions like textual, logical, mathematical or problem-related descriptions are allowed.

The description of systems using a Situation-Operator model allows

- the mixture of different types of variable quantities,
- the integration of logical and numerical quantities, and
- the description of real-world problems using a mixture of a complex set of descriptions (variables), which is in interaction with IS.

Operators are used to model the system changes (changes of situations). This also defines the discrete events of the change of situations. Operators and situations are strongly connected due to the identity (partly or complete) of the characteristics of the situations and the explicit assumptions of the operators. This includes that the situation consists of 'passive' operators (internal causal relation:

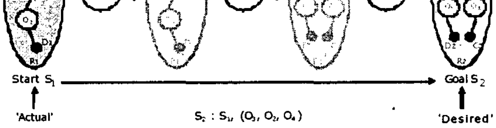


Fig. 4: Sequence of operator changing the situations from the originator to the desired goal (example)

'because'), whereby the change is done by 'active' operators (external causal relation: 'to'), shown in figure 3. The change of the considered world results as a sequence of actions modeled by operators as illustrated in figure 4.

Please note that operators correspond to situations. Both are not only used for structural organization of the system, but also for internal representation and storage of IS. They are the core/background of all higher organized internal operations and functions of the IS like learning, planning etc. [11].

3 Learning

The following assumptions have been made:

- The problem-dependent structures of the real world scenes can be clearly identified as situation dependent R's and C's.
- The resulting identified S describes the RW in the way, that the relevant structure of the scenes and those of the identified S is equal.
- Operators are defined as time-independent.

Based on the introduced assumptions, learning appears as the definition / redefinition of operators, driven by the interaction between IS and the considered system, whereby the interaction can be intended or not, useful or not, and planned or not. This includes several different cases, where Si denotes the i-th situation, Ri denotes the i-th set of relations *and* Ai,Bi,Di denote a set of characteristics Ci describing the system structure of the i-th situation.

A straight forward learning strategy includes the definition of $O1$ by his induced (: active learning) or observed (: passive learning) situation changes,

Oi: $Si(Ri(Ci)) \rightarrow S(i+1)(Ri(C(i+1))$

$Ri = R(i+1),\ Ci \neq C(i+1)$

Oi: $Si(Ri(Ci)) \rightarrow S(i+1)(R(i+1)(Ci))$

$Ri \neq R(i+1),\ Ci = C(i+1)$

Oi: $Si(Ri(Ci)) \rightarrow S(i+1)(R(i+1)(C(i+1))$

$Ri \neq R(i+1),\ Ci \neq C(i+1)$

which includes possible changes of situation structures $Ri \rightarrow R(i+1)$, characteristics $Ci \rightarrow C(i+1)$ or both.

This forward learning and definition procedure of operators may be the main mechanism to map the outer world of the considered environment to the inner (mental) world of IS and to add new experiences into the 'database' of IS.

This assumes that IS is able to identify Ri,Ci from the available sensor inputs in combination with the actual knowledge. This can not be assumed in general. To overcome the included problems of learning coincidental coherencies and learning non-concrete coherencies due to insufficient memory – mental model (MM) - capabilities, it is necessary to include backward oriented learning abilities: this includes the ability to distinguish Cs necessarily connected to R to those of coincidental presence and not directly connected to the problem structure.

Example 1:

The reality consists of $Si(Ri(Ai,Bi),Di)$ (Ri connects Ai and Bi, Di is unconnected present) and the learning mechanism of IS assumes / identifies Si as $Si(Ri(Ai,Bi,Di))$. After application of the $O1$-related action the system appears as $S2(R1(A2,B2,D1))$, so $O1$ can be defined by IS as

$O1: S1(R1(A1,B1,D1)) \rightarrow S2(R1(A2,B2,D1))$.

Due to the contingencies of the reality it may happen that

$S1(R1(A1,B1)),O1 \rightarrow S2(R1(A2,B2))$

can be observed, which is in opposition to definition of $O1$ mentioned before. So IS gets the chance to rebuild the $O1$ definition by 'replaying' to find the true $S1,O1 \rightarrow S2$ sequence redefining the operator $O1$. In the example this gives the opportunity to reject the coupling of $D1$ to $O1$.

In this way learning appears as a strictly nonlinear procedure due to the strong connection of the definition process of operators to the actual context, which includes the individual initial conditions of IS (the actual S and MM).

Example 2:

The task of IS should be the realization

$S1(R1(A1,B1,D1)) \rightarrow S2(R1(A2,B2,D1)$.

IS will take $O1$, as learned. Different results are possible:

- $D1$ appears as learned, so $O1$ seems to be confirmed.
- $D1$ changes unexpectedly to $D2$ or disappears.

As a result the reality may be in contradiction to the MM, so occurring differences give good reasons to reflect and change the definitions (previous learning procedures). It depends on internal features of IS to rebuild the MM immediately, after additional experiences or after extensive hypothesis oriented tests of the definition of $O1$ or not.

Please note that this definitions of learning are independent from external commendations, penalties or rewards. The learning capabilities are the key feature for successful acting in unknown situations.

From a system-theoretic point of view the key feature is the model-updating capability of IS.

4 Planning, Action, and Achievement of Planning

In this context planning is assumed as the internal preparation of the action or the series of actions to change actual $S(act)$ to desired ones $S(des.)$, cf. figure 4. Modeling of planning based on the SOM-technique includes a MM as a set of previously learned definitions / operators and the ability to identify the given goal $S(des.)$ and $S(act.)$. The goal elaboration is not considered here. To elaborate goals (or part goals) detailed procedures (as algorithms) have to be developed.

Planning includes the elaboration of a sequential ordered set of suitable Oi to solve the *task* $S(act.) \rightarrow S(des.)$. Due to the definition of S and O this can be done by comparison of Ci, eA, F applying a backward or forward inference strategy. The solvability strongly depends on the actual content of MM. If this can not be solved exactly (different reasons possible), practical planning procedures are possible which use operators which do not exactly fulfill the requirements (full set of Cs), but requirements close to the desired perfect ones. This will lead to testing strategies, associative combinations (where internal similarities between the relation eA,F of the supposed unknown, but perfect O and the C of known O exist). In reality conflicts may exist between goals, part goals, necessary actions, reachable situations and unexpected effects of 'known' operators. This may be typical for human interactions but also will appear for AS. The collection of possible human errors and the related SOM-oriented representation shows that there exists a large variety of practical problems [11]. In general solvable conflicts can be solved using decision making strategies with given goals. Therefore the solving strategy is to transform the problem to a higher level, where a solution may be given using an algorithm etc.

This includes the development and evaluation of alternative paths (operator sequences), the choice of weighting factors etc. and also strongly depends on the MM. In the (low level) case of scalar expressions (as characteristics and relations modeled by ODE) conflicts can be expressed by mathematical expressions, which can be solved perfectly or can be optimized using weighting functions to find compromises related to given goal functions. In the assumed general case of considering IS with formalizable and changing environments this problems are not considered up to now. Therefore game theory gives hints to the way the problem solution can be structured and solutions can be found.

The execution of the mentally prepared sequence of operators as actions realizes the interaction of IS with RW. The interaction itself gives a variety of learning sequences: the result of each action of IS can be compared with the predicted one to optimize the internal MM etc.

5 The resulting hiercharchy of system control

Here only a brief introduction into the system-theoretic consideration of different control approaches developed in [11] is given.

Feedback control is understood 'as the operation that, in the presence of disturbances, tends to reduce the difference between the output of the system and some reference input and that does so on the basis of this difference' [6]. The influence / effect of the values to be controlled to the input values is realized by the controller. Characteristics of automatic control are

- the closed loop by feeding back values to be controlled and
- the automatic response.

The design of controllers gives

- fix strategies / rules /relations / or just a set of gains as control coefficients which connect system output with system input and
- a system extension to optimize the system behavior.

Based on the SOM-structured view to system-IS-interaction a new and more general view to automatic control appears.

SOM-based definition:

Automatic control defines the autonomous transfer from actual to desired situations, whereby the inner structure or the modification of the inner structure of the considered and through the situation-space described system is used.

The proposed definition includes the old one, solves the problem of integrating algorithms / soft-computing algorithms, includes the interaction of IS, and - much more important - gives the view to new applications due to a homogeneous and uniform system approach.

5.1 Example: SOM-view to classical control

A controlled continuous SDF system, where the input (B) – output (A) behavior can be described using an ordinary differential equation, appears in the SOM-structured context as a fix situation. The connections are as follows:

Characteristic *Ci*:	*A*:	(Scalar) system output with time variant values
	B:	(Scalar) system input with time variant values
	C:	(Scalar) reference value
Relation *Ri*:	*r1*:ODE (B input, A output)	
	r2:Controller rule (A,C inputs, B output)	
Operator:	No operator exists, due to the fixed control law structure, this includes that nothing more will be changed by the controller	

The characteristics are scalar and fixed, the parameters of *A,B,C* are time variant, the *ri* are fixed, the implementation of control connects two situations (without and with control) by the application of *r2*, after the implementation of the control structure there is - from a SOM-theoretic point of view - only the fixed situation including the closed loop (as a closed chain of relations). This includes a fixed situation space, only depending on the dynamic behavior determined by the system behavior (described by *ri*) and the input, which can not be changed by this type of control.

5.2 Example: SOM-view to algorithms

An algorithm appears as a fixed sequence of situations, which may depend on external values. The algorithm steps directly correspond to operators. In contrast to IS-capabilities like cognitive based human behavior the complete algorithm is fixed before his execution.

Characteristic *Ci*:	Data of the algorithm
Relation *Ri*:	Internal connections between *Ci*, given by the problem modeling implemented in data-objects
Operator *Oi*:	Execution procedures change the objects of the algorithm: the data; the sequence of operators is defined before the operation itself, the situation space is previously defined.

The algorithm controls the changing of the objects of the algorithm: the data. The integrated feedback mechanism is not necessarily numerically defined, furthermore logical comparison is possible (repeat-until, while-do, for-do - algorithm). Additionally feedback elements are possible integrating external effects wh ich are combined with actual data outputs. Especially the dynamic sequences represent a higher quality of feedback, not only restricted to the comparison of (mostly scalar) numerical values and variables.

Without external effects there are not any feedback elements. With external effects (external given parameters, decisions etc.) algorithms also control the change and the flow of data goal-oriented. The reachable situation space is defined in advance, which includes that algorithms are not completely autonomous.

5.3 Example: SOM-view to IS

An intelligent system is able to interact goal-oriented with unknown or complex environments. In contrast to classical control laws and algorithms IS are able to learn relations between observed sets of characteristics and to modify them.

Characteristic *Ci*:		Facts of the environment partially corresponding to mental elements of ISs database
Relation *Ri*:		Internal connections between *Ci*, given by the structure of the considered system environment
Operator *Oi*:	Active:	Item which corresponds to the change of the system
	Passive:	Item which maps outer world relations to inner mental relations

The key features of IS are learning and database capabilities. The main differences are:

- The interaction with a given environment is neither restricted to known (simple) single characteristics (classical control) nor to a known system (classical control, algorithm).
- The IS-system behavior can be defined in the moment of interaction, which will give a large flexibility, whereby with classical control the structural system behavior is fixed in advance (by given rules (e.g. coefficients)), or predefined as with algorithms.
- IS are able to learn, this includes 'cognition' about the interaction and the observed behavior of the environment. It should be noted, that this strongly depends on the actual internal (mental) model and on the sensor input capabilities and will lead to a highly nonlinear process.

6 New criteria to distinguish different control and interaction mechanisms

To distinguish different levels of control new criteria are necessary, which are suggested here:

6.1 Quality of the closing feedback and reference

The comparison between the reference (which represents the goal) and the control value can be done in different ways and with different qualities, called characteristics here.

- Example 6.1.1: Classical Control

Physical differences (numerically expressed) between scalar characteristics are widely used, the comparison is directly forwarded to the controller.

- Example 6.1.2: Algorithm

Logical and / or numerical differences control the part of algorithms, which includes control structures (c.f. while-do, repeat-until, for-do constructs).

- Example 6.1.3: IS

Generalized logical characteristics or sets of characteristics are used defining reference (goal) and actual situation.

6.2 Connection between control criteria and control goal

The previously mentioned quality of the closing feedback is connected to the goal of the feedback itself.

- Example 6.2.1: Classical Control

The fixed control law is designed with a fixed reference situation as goal. The fixed goal situation defines the fixed operator which is applied to reach the goal. The goal is represented by the reference value.

- Example 6.2.2: Algorithm

The fixed algorithm also represents a fixed goal, resp. the steps (sequence) the goal.

- Example 6.2.3: IS

The goal of the interaction can be changed during the interaction, the goal is represented by an internal situation. The goal gives the reference to develop strategies, for planning purposes and to define actual acting.

6.3 Time behavior during control

- Example 6.3.1: Classical Control

Classical controllers typically work time continuously, time discretely, sometimes event discretely.

- Example 6.3.2 Algorithm

Algorithms (and mainly programs) are working time discretely.

- Example 6.3.3 IS

The interacting behavior of IS with their environment is connected strongly with the goal and the tasks and the resulting actions. It can be continuous, time discrete, or event discrete, but in general it is action discrete.

6.4 Variability of control (law)

The variability of control defines the possibility to change the feedback and therefore the control law.

- Example 6.4.1: Classical Control

No variability exists, may be that the fixed laws can be adapted using (fixed) adaptation rules.

- Example 6.4.2: Algorithm

Some kind of variability exist, but is related to a previously defined set of rules / steps.

- Example 6.4.3: IS

The IS is able to vary goals and rules. This is the key feature of IS. The resulting (unconsidered and also open) question is how to guarantee stability or success.

6.5 Anticipation of the situation trajectory

- Example 6.5.1: Classical Control

The situation trajectory usually consists of a single situation and a fixed set of characteristics, whereby the parameters are changing.

- Example 6.5.2: Algorithm

The situation trajectory can be complex, is usually not fixed, and is open in a given situation space.

- Example 6.5.3: IS

The situation trajectory may be unknown.

7 The architecture of a SOM-based autonomous system

The architecture of the proposed behavior- and memory- based open system is given in figure 5. Here additionally modules of hypothesis testing, of data based reconfiguration and, very important, of object- and scene recognition, scene and phenomenological interpretation are given for the example of a mobile robot.

It will be clear that the system is able to get an impression of his environment, depending on his sensor inputs, the observable change of the environment and on his interaction sequence. The system is designed for interaction with unknown environments, which includes that firstly interactions are needed to build the MM.

The basic modules

- electromechanical system of the mobile robot,
- basic control (drives, collision avoidance), and
- vision system
- object recognition

should not be declared here.

The function of the new modules for scene interpretation, phenomenological interpretation, planning, testing, and learning is declared briefly.

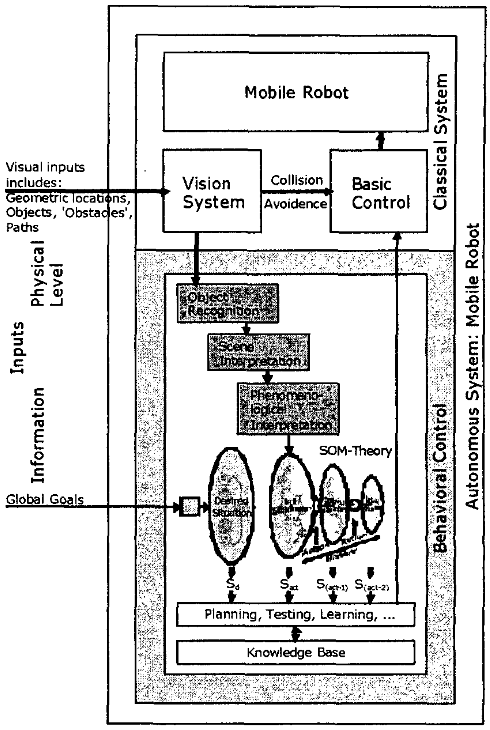


Fig. 5: Outline of the proposed intelligent control scheme based on the SOM-technique

7.1 Scene interpretation

The scene interpretation module connects the vision system / object recognition with the logical interpretation module. As input reference data sets of the knowledge base about known objects, situations, and strategies ('situation' environment) can be used. Therefore the physical oriented input information are clustered, referenced and summarized to a set of characteristics.

7.2 Logical interpretation

The logical interpretation module connects the set of characteristics to known sets of characteristics as known situations identifying the logical structure of the actual situation as new or as known. The result is related to given external goals or even internal part goals etc.

7.3 Planning

The planning module has two tasks: to develop strategies reaching external given goals and to observe actual situations and relate them to the previously defined strategy.

For developing strategies the knowledge database has to be used ordering operators, defining sequences, to prepare the acting of AS. The developed strategy has to be given to the execution level of AS. Observation of the actual situation is important, because of unexpected changes of the environments and of planning errors etc. The detection of difference between the planned reference and the actual situation gives the flexibility to change strategies, to react to unexpected situations and to learn.

7.4 Learning

In contrast to classical control schemes and to algorithms an autonomous intelligent system has not only the feature of changing strategies but also of adapting the knowledge background to new situations (and environments). The transformation of spatial (and visual) information into the structural logical information based on the SOM-technique may allow the strong reduction of new information to the relevant source. The algorithms of the learning module (as declared with different qualities) give data for the knowledge base, arranges and changes the knowledge base.

7.5 Testing

Besides the case that situations clearly can be identified, not ambiguous situations or relations can be checked using testing strategies. Furthermore it might be necessary to test the environment to learn something about it. This might be a worst case solution, if other solutions are not available. The other necessary aspect can be compared with the learning by playing aspect.

7.6 Knowledge Base

Using a SOM-discretized description of the sensor inputs, the developed strategies etc., allowing definition and redefinition of operators etc. the requirements for the technical realization of the data base system are high, this is mainly due to the fact that data sets are dynamically and open, furthermore the assignments are also dynamical.

8 Conclusions

The contribution briefly describes elements and architecture of a new kind of intelligent control to autonomous systems like mobile robots based on the description of the Human-Machine-Interaction.

References

1. Cacciabue, P.C., Modelling and Simulation of Human Behavior in System Control, Springer series: Advances in Industrial Control, Springer, Berlin, Heidelberg, New York, 1998.

2. Gigerenzer, G.; Goldstein, D.G., Reasoning the Fast and Frugal Way: Models of Bounded Rationality, Psychological Review, 103, 1996, No. 4, 650-669.

3. Gö rz, G. (Ed.), Einführung in die kü nstliche Intelligenz. Addison-Wesley, 1995.

4. McCarthy, J.; Hayes, P.J., Some Philosophical Problems from the Standpoint of Artificial Intelligence, Machine Intelligence, 4, 1969.

5. Mü ller, J. (Ed.), Verteilte Kü nstliche Intelligenz, BI-Wissenschaftsverlag, 1993.

6. Ogata, K.: Modern Control Engineering,. 2nd Edition, Prentice Hall, New Yersey, 1990.

7. Sandewall, E., Features and Fluents, Clarendon Press, Oxford, 1994.

8. Schweitzer, G., Probleme und Methoden zur Untersuchung des Regelverhaltens des Menschen, in: Oppelt, W.; Vossius, G., Der Mensch als Regler, VEB Verlag Technik, Berlin, 1970, 159-238.

9. Soeffker, D., Observer-based measurement of contact forces of the nonlinear rail-wheel contact as a base for advanced traction control, in: Wallaschek, J.; Lü ckel, J.; Littmann, W. (eds.): Mechatronics and Advanced Motion Control, Proc. of 3rd International Heinz Nixdorf Symposium, HNI-Verlagsschriftenreihe, Band 49 (1999), pp. 305-320.

10. Soeffker, D., Modeling the Human-Machine Interaction. in: Technical Report SS-99-04 Knowledge System Lab, Stanford University, California, 1999.

11. Soeffker, D., System-theoretic modeling of the knowledge guided human interaction, Habilitation Thesis (in german), University of Wuppertal, 2001, Text version, accepted.