Möbus, C., Eilers, M., Prototyping Smart Assistance with BAD Models, in: Mastrogiovanni, Chong (eds), Handbook of Research on Ambient Intelligence and Smart Environments, IGI Global, USA, 09/05/2010

Prototyping Smart Assistance with Bayesian Autonomous Driver Models

Claus Moebus[1, 2] and Mark Eilers[2]

Learning and Cognitive Systems / Transportation Systems
C.v.O University / OFFIS, Oldenburg, Germany
http://www.lks.uni-oldenburg.de/
claus.moebus@uni-oldenburg.de, mark.eilers@offis.de

Abstract. The Human or Cognitive Centered Design (HCD) of intelligent transport systems requires digital *Models of Human Behavior and Cognition (MHBC)* enabling *Ambient Intelligence* e.g. in a smart car. Currently MBHC are developed and used as *driver models* in traffic scenario simulations, in proving safety assertions and in supporting risk-based design. Furthermore, it is tempting to prototype assistance systems (AS) on the basis of a human driver model cloning an expert driver. To that end we propose the *Bayesian estimation* of *MHBC*s from human behavior traces generated in new kind of learning experiments: *Bayesian model learning under driver control*. The models learnt are called *Bayesian Autonomous Driver (BAD) models*. For the purpose of smart assistance in simulated or real world scenarios the obtained BAD models can be used as *Bayesian Assistance Systems (BAS)*. The *critical* question is, whether the driving competence of the BAD model is the same as the driving competence of the human driver when generating the training data for the BAD model. We believe that our approach is superior to the proposal to model the strategic and tactical skills of an AS with a *Markov Decision Process (MDP)*. The usage of the BAD model or BAS as a prototype for a smart Partial Autonomous Driving Assistant System (PADAS) is demonstrated within a racing game simulation.

Keywords: smart assistance, Bayesian autonomous driver models, driver assistance systems, Bayesian Assistance Systems, learning of human control strategies, human behavior learning and transfer, mixture-of-behaviors model, visual attention allocation, partial autonomous assistance system, shared space, probabilistic detection of anomalies, anticipatory planning, dynamic Bayesian networks, between-vehicle-cooperation, within-vehicle-cooperation.

---

[1] project Integrated Modeling for Safe Transportation (IMOST) sponsored by the Government of Lower Saxony, Germany under contracts ZN2245, ZN2253, ZN2366

[2] Project Integrated Human Modelling and Simulation to support Human Error Risk Analysis of Partially Autonomous Driver Assistance Systems (ISi-PADAS) funded by the European Commission in the 7th Framework Program, Theme 7 Transport FP7-218552

Möbus, C., Eilers, M., Prototyping Smart Assistance with BAD Models, in: Mastrogiovanni, Chong (eds), Handbook of Research on Ambient Intelligence and Smart Environments, IGI Global, USA, 09/05/2010

1 Introduction

The Human or Cognitive Centered Design (HCD) [1-3] of intelligent transport systems requires digital Models of Human Behavior and Cognition (MHBC) enabling Ambient Intelligence (AMI) e.g. in a smart car. The AMI paradigm is characterized by systems and technologies that are *embedded*, *context aware*, *personalized*, *adaptive*, *anticipatory* [4]. Models and prototypes we propose here are of that type.

Currently MBHC are developed and used as driver models in traffic scenario simulations [5], in proving safety assertions and in supporting risk-based design [6]. In all cases it is assumed that the conceptualization and development of MHBCs and ambient intelligent assistance systems are *parallel and independent* activities [7 - 9]. In the near future with the need for smarter and more intelligent assistance the *problem of transferring human skills* [10] into the envisioned technical systems becomes more and more apparent especially when there is no sound skill theory at hand.

The *conventional approach* to develop smart assistance is to develop control-theoretic or artificial-intelligence-based prototypes [5-9] first and then to evaluate their learnability, usability, and human likeness *ex post*. This makes revision-evaluation cycles necessary which further delay time-to-market and introduce extra costs. An *alternative approach* would be the *handcrafting* of MHBC [11-17] on the basis of human behavior traces and their *modification* to prototypes for smart assistance. An ex post evaluation of their human likeness or empirical validity and revision-evaluation cycles remains obligatory, too.

We propose a third *machine-learning alternative*. It is tempting to prototype assistance systems on the basis of a human driver model *cloning* an expert driver. To that end we propose the *Bayesian estimation of* MHBC*s from human behavior traces* generated in new kind of learning experiments: *Bayesian model learning under driver control*. The models learnt are called *Bayesian Autonomous Driver (BAD) models.*

Dynamic probabilistic models are appropriate for this challenge, especially when they are learnt online in *Bayesian model learning under driver control*. For the purpose of smart assistance in simulated or real world scenarios the obtained BAD models can be used as prototypical Bayesian Assistance Systems (BAS). The *critical* question is, whether the driving competence of the BAD model is the same as the driving competence of the human driver when generating the training data for the BAD model.

We believe that our approach is superior to a proposal to model the strategic skills of a PADAS with a *Markov Decision Process (MDP)* [18]. A MDP needs a reward function. This function has to be derived deductively from theoretical concepts or learnt inductively from car trajectories by solving the *inverse reinforcement learning problem* [19]. The deductive derivation of reward function often results in strange nonhuman overall behaviors. The inductive mining of the reward function from car trajectories or behavior traces seems to be a detour and more challenging than our approach.

The two new concepts *Bayesian learning of agent models under human control* and the *usage of a BAD model as a BAS or PADAS* are demonstrated when constructing a prototypical *smart assistance system* for driving stabilization within the racing game simulation TORCS [20].

BAD models [21-26] are developed in the tradition of Bayesian expert systems [27, 28], probabilistic robotics [29], and Bayesian (robot) programming (BP) [30-33]. For *Bayesian model learning under driver control* we need concepts from *parameter learning* in Bayesian networks [27]. We distinguish *descriptive* and *normative* BAD models. *Descriptive* models can be learnt from the behavior of individuals or groups of drivers. They can be used for simulating human agents in all kinds of traffic scenarios. *Normative* models are learnt from the behavior of *ideal* or *special instructed* human drivers (e.g. driving instructors, racing car drivers). They may be used for the conceptual new *BAS*. Due to their probabilistic nature BAD models or BAS can not only be used for *real-time control* but also for *real-time detection of anomalies* in driver behavior and *real-time generation* of supportive *interventions (countermeasures)*.

2 Distributed Cognition, (Partial) Cooperation, Smart Assistance, and Ambient Intelligence in Driving Scenarios

The concept of *distributed cognition* was originated by Edwin Hutchins in the mid 1980s [34]. He proposed that human knowledge and cognition is not confined to individuals but is also embedded in the objects and tools of the environment. Cognitive processes may be distributed across the members of a social group or the material or environmental structure. With anthropological and non-experimental methods Hutchins studied how crews of ships can function as a *distributed machine*. He preferred studying cognitive systems not as individual agents but which are composed of *multiple agents* and the material world. In later studies he generalized the domains and put an emphasis on airline cockpits crews and human-computer interaction scenarios. In a sense Hutchins anticipated the concepts of ambient intelligence with its *embedded*, *context aware*, *personalized*, *adaptive*, *anticipatory* systems.

Crews on navigation bridges or in aircraft cockpits work in agreement with a *single* principal. Such a scenario is called *cooperative* [35]. A crew forms a cohesive group whose members normally cooperate for longer periods in solving the problems arising from ship or aircraft control. This cooperation includes exchange of complex verbal messages which require a high dimensional state space for the agent models.

Public traffic scenarios are of a fundamentally different kind. Communication, cooperation and the action repertoire of agents is limited in amount and complexity. Agents are their own principals and do not belong to a formal cohesive group. Thus, a scenario is *partial* or *non-cooperative*, when goals are issued by *several different principals* [35]. Traffic agents form ad hoc groups by chance and try to maximize their personal utilities. Internal group norms are substituted by external traffic rules. The solution to a traffic coordination problem is a distributed but synchronized sequence of sets of actions (e.g. collision-free crossing an intersection) emitted by different autonomous agents. Successful problem solutions require (nonverbal) communication and distributed cognition across agents and artifacts.

2.1 Cooperative Scenarios: Crews and In-Vehicle-Dyads

Members of a public traffic scenario with Between-Vehicle Cooperation (BVC) do not form a stable social group but rather an ad hoc group with a limited life time and communication vocabulary. In contrast to that members in a nonpublic traffic scenario (Fig. 1) with In-Vehicle Cooperation (IVC) form for a short time period a stable social group similar to a crew.

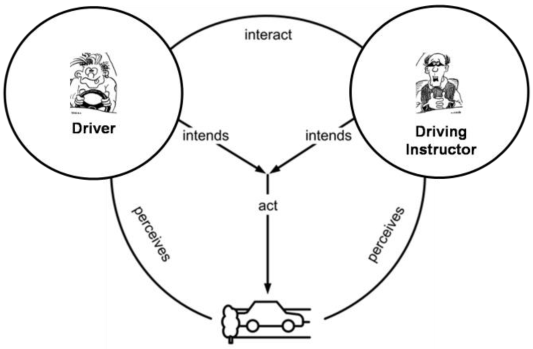


Fig. 1: Driving-school-scenario with in-vehicle-cooperation (graphics from [23] with kind permission of publisher of [7] and Springer Science and Business Media)

2.2 Partial Cooperative Scenarios: Ad-hoc groups and Shared Space

Shared space describes an approach to the design, management and maintenance of *public spaces* which reduces the adverse effects of conventional traffic engineering by stimulating the *situation awareness* of all traffic agents (Fig. 2).

Fig. 2: *Shared-space* with between-agent-, between-vehicle-, and in-vehicle-cooperation [36] (graphics with kind permission of Ben Hamilton-Baillie)

The shared space approach is based on the observation that individuals' behavior in traffic is more positively affected by the built environment of the public space than by conventional traffic control devices (signals, signs, road markings, etc.) or regulations. An explanation for the apparent *paradox* that a reduction in regulation leads to safer roads may be found by studying the *risk compensation effect*: "Shared Space is successful because the *perception of risk* may be a *means* or even a *prerequisite* for increasing public traffic safety. Because when a situation feels unsafe, people are more alert and there are fewer accidents." [37]

2.3 Smart Assistance in Traffic Scenarios

Traffic maneuvers can generate risk anytime. We call risky maneuvers *anomalies* when they have a low probability of occurrence in the behavior stream of experienced drivers and which only experienced drivers are able to prevent or to anticipate automatically. Other drivers probably cannot and therefore might need support generated by a BAS or PADAS. It is expected from assistance systems that they will enhance *situation awareness*, *cooperation*, and *driving competence* of unskilled or non-cooperative drivers. Thus the design challenge of smart assistance should aim at modeling human traffic agents with their (erroneous) beliefs, expectations, behavior, situation awareness, and their skills to recognize situations, to diagnose and prevent anomalies. These BAD models should then be adapted to BAS or PADAS to solve the *problem of transferring human skills*.

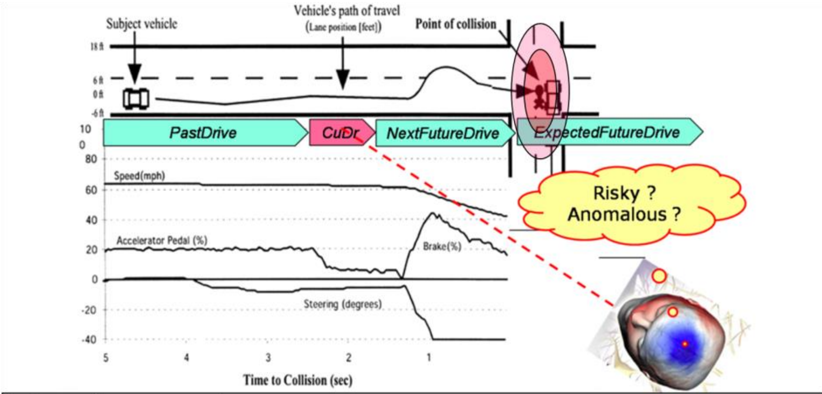
2.4 The Need for Bayesian Assistance in Vehicles with In-Vehicle-Cooperation

As an example for the concept of a BAS we present a scenario based on result of a study of Rizzo et al. [38]. The authors studied the behavior of drivers suffering from Alzheimer disease. At a lane crossing a car incurred from the right (Fig. 3). Many maneuvers of the Alzheimer patients ended in a collision, as they suffered from the *looking without seeing syndrome*. The modeling task should lead to a probabilistic BAS model, which is *diagnosing* and *correcting* the *anomalous* behavior of inexperienced or handicapped drivers. Fig. 3 demonstrates the *probabilistic prediction* of hazardous events, anomaly *detection* (1.) and the *anticipatory control* of the driver's behavior by the BAS (2.).

Pink ellipses denote contours of constant density. A driver's behavior is *risky or anomalous* if its behavior is unlikely under the assumption that the driver belongs to a group of normal error-free routine drivers. For *anticipatory planning* the conditional probability of the *NextFutureDrive* under the assumption of the *pastDrive*, the *currentDrive*, and the anticipated *expectedFutureDrive* has to be computed. The BAS gives an advice sampled from this conditional distribution (e.g. the expected value $E(NextFutureDrive \mid pastDrive, \ldots, exptectedFutureDrive)$).

Fig. 4 shows the replacement of the real driving inspector by the corresponding BAS model. Different BAS-types like an experienced Schumacher-racing-style BAS are possible (Fig. 5).

How can the BAS be derived by methods of Bayesian driver modeling? We explain this within an obstacle scenario which is known to generate driver *intention conflicts* (Fig. 6).



1. Anomaly assessment:

$$\left[1 - P(currentBehavior \mid CorrectBehavior)\right] > AnomalityThreshold$$

2. Anticipatory planning:

$$P(NextFutureDrive \mid expectedFutureDrive, currentDrive, pastDrive)$$
$$= \sum_{Context} P(NextFutureDrive, Context \mid expectedFutureDrive, currentDrive, pastDrive)$$

Fig. 3: Driving behavior of an Alzheimer patient in a simulated intersection incursion [38], (1.) *risk assessment* of a the current behavior or trajectory, and (2.) *anticipatory planning* of a BAS (graphics from [23] with kind permission of Springer Science and Business Media)

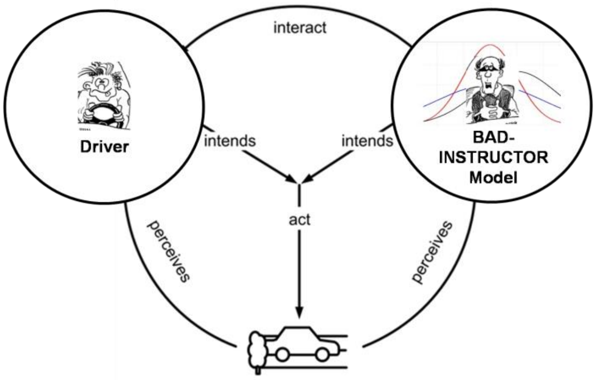


Fig. 4: Cooperative driving scenario with in-vehicle-cooperation between a non-expert driver and a BAS-prototype *Driving Instructor* (graphics from [23] with kind permission of publisher of [7] and Springer Science and Business Media)

When an obstacle (animal, car) is appearing unexpectedly people *autonomously* react with a maneuver $\mathbf{M}^-$ which is *not* recommended by experts. $\mathbf{M}^-$ drivers try to avoid collisions even at high velocities by steering to the left or right risking a fatal turnover. The *recommended* maneuver $\mathbf{M}^+$ includes the *hold and brake* sub-maneuvers though most times ending up in a collision.

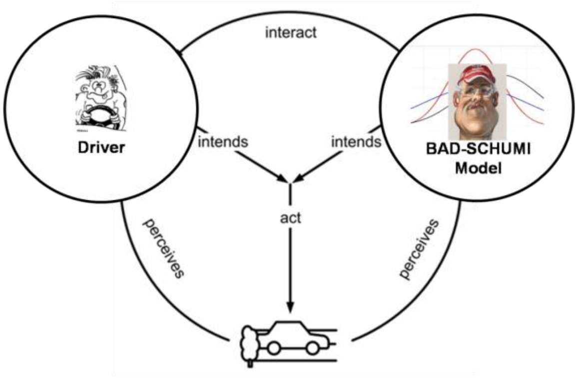


Fig. 5: Cooperative driving scenario with in-vehicle-cooperation between a non-expert driver and a BAS-prototype *Racing Driver*[3] (background graphics from [23] with kind permission of publisher of [7] and Springer Science and Business Media)

When drivers are instructed to drive $\mathbf{M}^{+}$ they generate data which are the training data for the BAS version of the PADAS according to the methods of chapters 4 and 5: *Bayesian learning of agent models under human control*.

With an existing BAS a worst-case scenario can be planned to test the services of the BAS. Drivers are instructed *not* to drive the recommended maneuver $\mathbf{M}^{+}$. Because of the probabilistic nature of the BAS it is possible to compute the conditional probability $P(currentDrive_t \mid \mathbf{M}^{+})$. This conditional probability is a measure of the *anomaly* of the driver behavior *under the hypothesis* that the observed actions are generated by a stochastic process which generated the trajectories or behaviors of the correct maneuver $\mathbf{M}^{+}$.

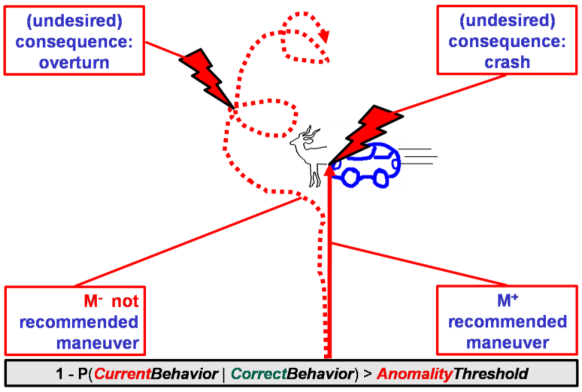


Fig. 6: Intention conflict scenario with conflicting behaviors $\mathbf{M}^{-}$ (incorrect or not recommended maneuver) and $\mathbf{M}^{+}$ (correct or recommended maneuver) (graphics from [23] with kind permission of Springer Science and Business Media)

[3] http://board.gulli.com/thread/573253-haderer-karikatur-von-michael-schumacher/ (25th,March 2010)

3 Probabilistic Models of Human Behavior and Cognition in Traffic Scenarios

Computational agent models have to represent perceptions, beliefs, goals, and actions of ego and alter agents. Agent models should

- predict and generate agent behavior sometimes in interaction with assistance systems
- identify situations or maneuvers and classify behavior (e.g. anomalous vs. normal) of ego and alter agents
- provide a robust and valid mapping from human sensory data to human control actions
- be learnt from time series of raw data or empirical frequency distributions with statistical sound (machine-learning) procedures making only a few non-testable ad hoc or axiomatic assumptions
- be able to learn new patterns of behavior without forgetting already learnt skills (stability-plasticity dilemma [39]).

A driver is a human agent whose skills and skill acquisition processes can be described by a well-known three-stage model with the *cognitive*, *associative*, and *autonomous* stages or layers [40, 41]. Accordingly various modeling approaches are adequate: (1) *production-system* models for the cognitive and associative stage (e.g. models in a cognitive architecture [13, 16, 17, 42-46]), *control-theoretic* [11, 12, 15, 16, 47, 48], or *probabilistic* models [21-26, 49-52] for the autonomous stage.
The great advantage of probabilistic models is that they avoid brittleness and provide *robustness*. This is a great advantage due to the *irreducible incompleteness* of knowledge about the environment and the underlying psychological mechanisms [33]. Furthermore probabilistic models of the *Bayesian* type are suited to implement MHBCs which are *embedded*, *context aware*, *personalized*, *adaptive*, *anticipatory* systems (Fig. 3, 6).

3.1 Bayesian Autonomous Driver Models

Due to the variability of human cognition and behavior and the *irreducible* lack of knowledge about latent cognitive mechanisms it seems rational to conceptualizes, estimate and implement *probabilistic* models when modeling human traffic agents. In contrast to other models probabilistic models are not *idiosyncratically handcrafted* but could be *learnt objectively* from human behavior traces. Model validity is either included in the modeling process by *model-driven data-analysis* without any ex-post validation or by our new machine-learning experiments: *Bayesian learning of agent models under human control*.

BAD models describe phenomena on the basis of variables and conditional probability distributions (CPDs). This is in contrast to models in cognitive architectures (e.g. ACT-R) which try to simulate cognitive algorithms and processes on a *granular* and *latent* basis which are difficult to identify even with technical sophisticated methods such as *functional magnetic resonance imaging (FMRI)* methods [43, 44].

According to the BP approach [30-33] BAD models [21-26] are instances of Bayesian Networks (BN) [53-57] using concepts from probabilistic robotics [29]. BP is a simple and generic framework suitable for the description of human sensory-motor models in the presence of incompleteness and uncertainty. It provides integrated *model-driven data analysis* and *model construction*. In contrast to conventional Bayesian network models BP-models put emphasis on a *recursive structure* and infer concrete motor actions for *real-time control* on the basis of sensory evidence. Actions are sampled from CPDs according various strategies after propagating sensor or task goal evidence. BAD models describe phenomena on the basis of the variables of interest and the decomposition of their joint probability distribution (JPD) into CPD-factors according to the *special chain rule for Bayesian networks* [56, p.36]. The underlying CIHs between sets of variables can be tested by standard statistical methods (e.g. the conditional mutual information index [56, p.237]). The parameters of BAD models can be learnt objectively with statistical sound methods by batch learning from multivariate behavior traces or by learning from single cases [27]. The latter approach is known as Bayesian estimation [56]. We use it for *Bayesian (online) learning of MHBC*s. The learning process runs in a new kind of learning experiments: *Bayesian learning of agent models under human control*. BAD models could be learnt solely by Bayesian adaption of the model to the real-time behavior of the human driver correcting the BAD model when necessary.

In [21] we described first steps to model lateral and longitudinal control behavior of single and groups of drivers with *reactive Bayesian sensory-motor models*. Then we included the time domain and reported work with *dynamic Bayesian sensory-motor models* [22, 23]. Now we work on the idea of behavior hierarchies and mixing behaviors [24-26, 58, 59]. The goal is a *dynamic* BAD model which is able to decompose complex situations into basic situations and to compose complex behavior from basic motor schemas (*behaviors*, *experts*). This Mixture-of-Behaviors (MoB) model facilitates the management of sensory-motor schemas in a library [24-26]. Context dependent driver behavior could be generated by mixing pure behavior from different schemas.

3.2 Basic Concepts of Bayesian Programs

A BP [30-33] is defined as a mean of specifying a family of probability distributions. By using such a specification it is possible to construct a BAD model, which can effectively control a (virtual) vehicle. The components of a BP are presented in Fig. 7 where the analogy to a logic program is helpful.

An *application* consists of a (competence or task model) *description* and a *question*. A *description* is constructed from *preliminary* knowledge and a *data set*. *Preliminary knowledge* is constructed from a set of *pertinent variables*, a *decomposition* of the JPD and a set of *forms*. *Forms* are either *parametric forms* or *questions* to other BPs.

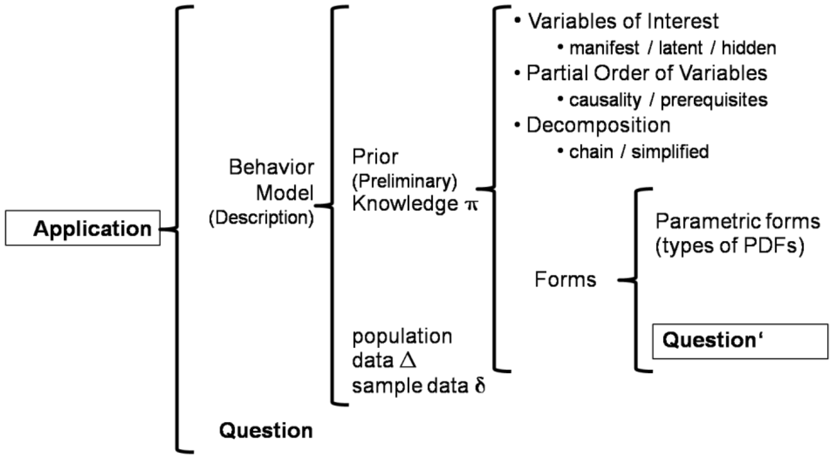


Fig. 7: Structure of a Bayesian Program (adapted from [30-33])

The purpose of a *description* is to specify an effective method to compute a JPD on a set of variables given a set of (experimental) *data* and *preliminary* knowledge. To specify *preliminary knowledge* the modeler must *define the set of relevant variables* on which the JPD is defined, *decompose the JPD* into factors of CPDs according to CIHs, and *define the forms*. Each CPD in the decomposition is a form. Either this is a *parametric form* which parameter are estimated from batch data (behavior traces) or another *question*. Parameter estimation from batch data is the conventional way of estimating the parameters in a BAD model. The *Bayesian estimation* procedure uses only a small fraction of the data (cases) for updating the model parameters. This procedure is described below.

Given a description a *question* is obtained by partitioning the variables into *searched*, *known*, and *unknown* variables. We define a question as the CPD $P(Searched \mid Known, preliminary\ knowledge, data)$. The selection of an appropriate action can be treated as the inference problem: $P(Action \mid Percepts, Goals, preliminary\ knowledge, data)$. Various *policies* (*Draw, Best,* and *Expectation*) are possible whether the concrete *action* is *drawn* at random, chosen as the *best* action with highest probability, or as the *expected* action. The last two strategies are necessary if the BAS should behave deterministically as demanded by industry.

3.3 Classes of Probabilistic Models for Human Behavior and Cognition

Currently we are evaluating the suitability of static and dynamic Probabilistic Graphical Models [57].

With the *static* type it is possible to generate reactive [21] and inverse (naïve) [22] models (Fig. A1.1 - A1.3). In practice, naïve Bayesian models can work surprisingly well, even when the independence assumption is not true [60, p.499]. Our research [21-26] has shown that static models generate behavior which is too erratic to be similar to human behavior. As a consequence we focus ourselves on the dynamic type of real-time control for simulated cars.

Dynamic models evolve over time. If the model contains discrete time-stamps one can have a model for each unit of time. These local models are called *time-slices* [56]. The time slices are connected through *temporal links* to give a full model.

A special category of time-stamped model is that of a *Hidden Markov Model* (HMM). They are repetitive temporal models in which the state of the process is described by a *single discrete* random variable. Because of the Markov assumption only temporarily adjacent time slices are linked by a *single* link between the state nodes.

In the case of identical time-slices and *several* identical temporal links we have a *repetitive temporal model* which is called *Dynamic Bayesian Network model* (DBN). The description of the DBN in the BF framework is shown in Fig. 8.

Application
- **Program**
  - **Specification**
    - **Variables of Interest:** $X_1^0,\dots,X_M^0,\dots,X_1^T,\dots,X_M^T$
    - **Partial Order of Variables: causality, prerequisites**
    - **Decomposition:**

      $$P(X_1^0,\dots,X_M^T)=P(X_1^0,\dots,X_M^0)\cdot\prod_{t=0}^{T}\prod_{j=1}^{M}P(X_j^t \mid parents(X_j^t))$$
    - **Forms**
      - **Parametric Forms (types of PDFs):**
      - ***Any***
  - **Identification**
- **Question** $P(X_j^T \mid Known)$

Fig. 8: Structure of a Dynamic Bayesian Network (DBN) as a Bayesian Program (adapted from [30]; graphics from [22] with kind permission of Springer Science and Business Media)

In 3.3.1 we present Markov, Hidden Markov Models (HMMs), and their generalization Dynamic Bayesian Networks (DBNs) and then in 3.3.2 we develop and evaluate in a proof of concept a sequence of models culminating in a *psychological motivated sensor-motor model with attention allocation* which could be the basis for a BAS.

3.3.1 Markov, Hidden Markov Models, and Dynamic Bayesian Networks

The *dynamic* type of graphical models [56, 57] enables the creation of Markov Models (MMs) [21 - 26], Hidden Markov Models (HMMs) (Fig. A1.4 – A1.6) [61 - 63], Input-Output-HMMs (IOHMMs) [64], Reactive IOHMMs (RIOHMMs, Fig. 9, Fig. A1.7), Discrete Bayesian Filters (Fig. A1.8 [29, 65]), Coupled HMMs (CHMMs; Fig. A1.9) [66], and Coupled Reactive HMMS (CRHMMS, Fig. A1.10). HMMs are sequence classifiers [67, 68] and allow the efficient *recognition* of situations, goals and intentions; e.g. diagnosing driver's intention to stop at a crossroad [63, 69]. Their suitability for the *generation* of behavior of Belief-Desire-Intention (BDI-) Agents will be evaluated in 3.3.2.

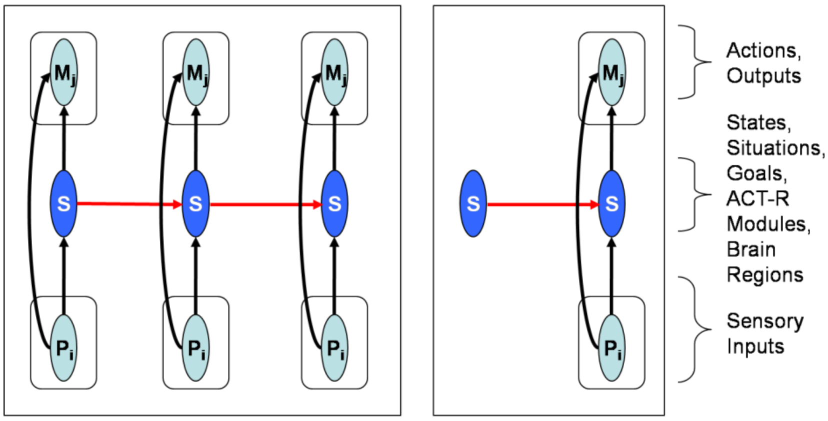


Fig. 9: (Reactive) Input-Output HMM (RIOHMM) as Probabilistic Abstraction of Anderson's cognitive ACT-R architecture: 2-time-slices *template* model (right), 3-time-slices *rolled-out* model (left) (graphics from [23] with kind permission of Springer Science and Business Media)

For instance, RIOHMMs (Fig. 9, A1.7) could in principle implement reactive driver models (e.g. with ACT-R module activations [42-46]). The two arrows into the random variable nodes M_j denote the combined dependence of actions on sensory inputs and activations of hidden ACT-R modules or brain regions. Even if module activations were known sensory inputs are still necessary to propose specific actions.

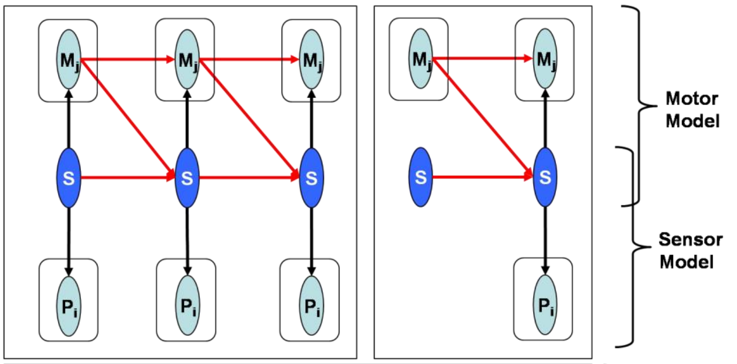
CHMMS and CRHMMS permit the modeling of several agents within the HMM formalism. The belief state of each agent depends only on his own history and on the belief state of his partner. Whether this is plausible has to be tested by conditional independence hypotheses [56]. Within each agent the model is of the HMM-type.
Whereas [58] rely on Hidden Markov Models (HMMs) for learning fine manipulation tasks like grasping and assembly by Markov mixtures of experts we strive for more general dynamic Bayesian Network (DBN) model in learning multi-maneuver driving behavior [23].

HMMs and DBN are mathematically equivalent. Though, there is a trade-off between estimation efficiency and descriptive expressiveness in HMMs and DBNs. Estimation in HMMs is more efficient than in DBNs due to algorithms (Viterbi, Baum-Welch [60, 68]) whereas descriptive flexibility is greater in DBNs. At the same time the state-space grows more rapidly in HMMs than in corresponding DBNs. Therefore we focus ourselves on DBNs and try to avoid the *latent* state assumption of HMMs, though it seems to be important to model the *state* of a driver/vehicle with the variables of *position, velocity*, *lateral and longitudinal (de|ac)-celeration.* This is implemented in the commercial product IPG-Driver [70]. The *state* of the driver/vehicle is important for the definition and description of *undesired events*, the *planning of countermeasures* and *intelligent anticipatory assistance* [25].

Especially two DBN models influenced our work. The first is the *Switching Linear Dynamic System* (Fig. A1.11) [63] and the second is the *Bayesian Filter and Action*

Model (Fig.10, Fig. A1.12) [65, p.180]. In both models actions are not only dependent on the current process state but also on direct antecedent actions. Thus the generation of erratic behavior is suppressed. Furthermore, the Bayesian Filter and Action Model includes *direct* action effects on the next future process state. This is important when the influence of action effects should be modeled *directly* into the state not making a detour via the environment and the perception of the agent. If the model should make predictions without an embedding (simulation) environment, we have to include a state and an action model. Only when the model can be treated as a self-contained mathematical object the properties of the model need not be evaluated in a simulation within the embedding simulation environment.

Also, including an action effect model is meaningful when some action effects are not perceivable by the agent. E.g. when a night-watchman closes and locks a door, the locking action has a *direct* effect on the state of the door: the door is closed *and* locked. But this *lockedness* is not visible. This could only checked only by further actions.



$P(S^t \mid ..., x_j^{0:t-1}, ..., z_j^{0:t-1}, ..., \pi_f)$; prediction of state

$P(S^t \mid ..., x_j^{0:t}, ..., z_j^{0:t-1}, ..., \pi_f)$; estimation of state

$P(M^t \mid ..., x_j^{0:t}, ..., z_j^{0:t-1}, ..., \pi_f)$; activation of motor commands

Questions

Fig. 10: Bayesian Filter and Action Model (adapted from [65, p.180]): 2-time-slices *template* model (right), 3-time-slices *rolled-out* model (left)

In our research [21-26] we strive for the realization of a dynamic Bayesian Autonomous Driver with Mixture-of-Behaviors (BAD-MoB) model. The model is suited to represent the sensor-motor system of individuals or groups of human or artificial agents in the functional *autonomous* layer or stage of Anderson [41]. It is a psychological motivated *mixture-of-experts* (= mixture-of-schema) model with *autonomous and goal-based attention allocation processes*. The template or class model is distributed across two time slices, and tries to avoid the *latent* state assumptions of HMMs. Learning data are time series or case data of relevant variables: percepts, goals, and actions. Goals are the only latent variables which could be set by commands issued by the higher *associative* layer.

The model propagates information in various directions. When working *top-down*, goals emitted by the associative layer select a corresponding expert (schema), which propagates actions, relevance of areas of interest (AoIs) and perceptions. When working *bottom-up*, percepts trigger AoIs, actions, experts and goals. When the task or goal is defined and the model receives percepts evidence can be propagated *simultaneously* top-down and bottom-up. As a consequence the appropriate *expert* (*schema*) and its *behavior* can be activated.

Thus, the model can be extended (Fig. 11) to implement psychological models, e.g. a modified version of the SEEV visual scanning or attention allocation model of Horrey, Wickens, and Consalus (2006). In contrast to Horrey et al. [71] the model is able to predict the probability of attending a certain AoI on the basis of single, mixed, and even incomplete evidence (goal priorities, percepts, effort to switch between AoIs). In 3.3.2 we show that this architecture is feasible.

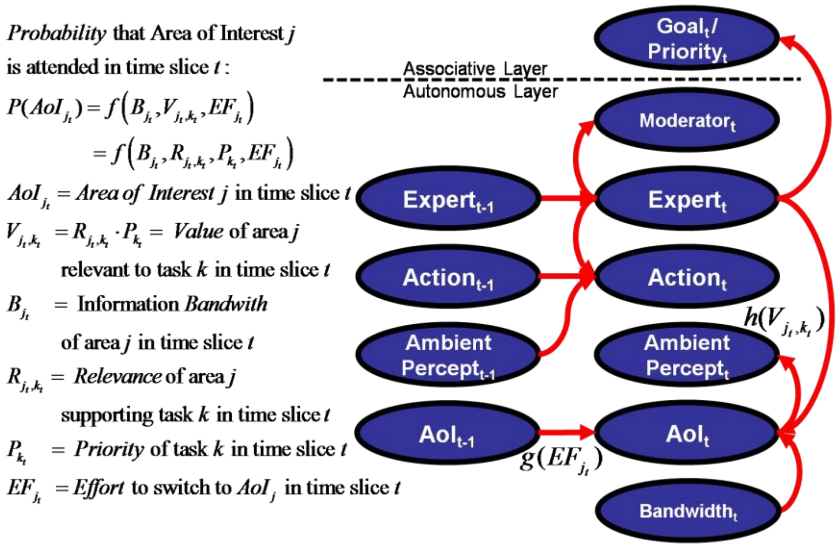


Fig. 11: Mixture-of-Behaviors (= Mixture-of-Experts) Model with Visual Attention Allocation Extension mapping ideas of Horrey et al. [71] into the Dynamic Bayesian Network modeling framework (graphics from [23] with kind permission of Springer Science and Business Media)

There are various scientific challenges designing and implementing BAD-MoB Models. The *first* main challenge is to describe driver-generated behavior by a *mixture-of-behaviors* architecture. While mixture-of-*experts* approaches are known from pattern classification [30, 59, 72] it is the first time that this approach is used in modeling human driver behavior [24-26]. In a MoB model it is assumed that the behavior can be context-dependent generated as a mixture of ideal schematic *behaviors* (= experts). Thus the stability/plasticity dilemma [39] of neural network models is avoided. A new behavior will only be learnt in special phases and by adding this new *behavior* to the library of behaviors. Behaviors do not influence each other directly. *Pure* behavior without any additional mixture component is shown only in typical pure situations (e.g. the perception of a hair pin triggers the hair-pin-model *behavior*).

The *second* main challenge is that we want to integrate various perceptional invariants known as tau-measures [73, 74] from psychological action control theory into a computational human model. In conventional models [5, 13, 16, 17] variables with different dimensions (distances, angles, times, changes, etc.) are input to the models. Tau measures transform all non-time measures into the time domain. Some measures are already used in standard engineering: time-to-collision (TTC) or time-to-line-crossing (TTLC).

3.3.2 From Discrete Bayesian Filters to Sensor-Motor Models with Attention Allocation

Now we give a proof of concept. We choose certain model classes and a set of constructed but plausible data and demonstrate that the models show the intended behavior.

In our research [21-26] we used partial inverted Markov models for modeling the sensory-motor system of the human driver (ch. 4; Fig. 24, 25, 31). We discuss what types of DBNs have to be considered when driver state variables (e.g. *lateral and longitudinal (de|ac)-celeration*) are included and when a psychological motivated *mixture-of-behaviors* model with *autonomous and goal-based attention allocation processes* is the ultimate goal (Fig. 11).

3.3.2.1 Discrete Bayesian Filter (DBF) and HMMs

We start with the Discrete Bayesian Filter (DBF) (Fig. A1.8). This is the most fundamental algorithm in probabilistic robotics for estimating state from sensor data [29]. The DBF is a HMM with state, percept and motor variables. The general algorithm consists of two steps in each iteration or recursive call [29, p.27]:

1. Prediction step: from the most recent *apriori* belief(state) and the current control (= action) compute a provisional belief(state)
2. Correction step: from the current provisional belief(state) and the current measurements (= percepts) compute *the posteriori* belief(state).

We extended a tutorial example from Thrun [29, ch.2.4.2] and implemented this DBF in NETICA [75], to show that *state identification* in a DBF works satisfactorily (Fig. A2.01, A2.02). Our model represents a night watchman approaching a door in the dark. Before he sees the door his *belief(state)* is uninformed. So the *apriori* belief distribution about the state is flat (Fig. A2.01). His beliefs are revised when he *pushes* the door (*prediction step*) (Fig A2.01). Now he believes with p_open = 0.633 that the door *is open* and with p_closed = (1-p_open) = 0.333 that the door is *closed and locked*. When turning on his flashlight he perceives that the door *is closed* (Fig A2.02). This leads in the *correction step* to the posterior *belief(state)* (Fig. A2.02): $P(State = is\_open \mid Action = push, \dots) = 0.161$, $P(State = is\_closed\_and\_locked \mid Action = push, \dots) = 0.763$, and $P(State = is\_closed\_and\_unlocked \mid Action = push, \dots) = 0.0763$.

Now we want to show that the DBF is *not* the right model class for the *implementation of a reactive agent*, because the steps in the iteration cycles for the *reactive* agent are different from those of the DBF:

1. Perception step: from the most recent *apriori belief(state)* and the current percept compute a provisional *belief(state)*
2. Action step: from the current provisional *belief(state)* and the current action compute *the posteriori belief(state).*

In the *perception step* the night watchman sees that the door *is closed* (Fig. A2.03). He revises his uninformed *apriori* beliefs. He is rather certain that the door *is closed*, but rather uncertain whether the door *is locked*: $P(State = is\_open \mid Percept = sense\_closed, \ldots) = 0.0526$, $P(State = is\_closed\_and\_locked \mid Percept = sense\_closed, \ldots) = 0.474$, and $P(State = is\_closed\_and\_unlocked \mid Percept = sense\_closed, \ldots) = 0.474$. Now the agent *pushes* the door (Fig. A2.04). The result is a bit surprising. The door is not opened in the current or next *state* $P(State = is\_open \mid Action = push, \ldots) = 0.161$ but the belief is that is the door *is closed and locked* $P(State = is\_closed\_and\_locked \mid Action = push, \ldots) = 0.763$!

The reason for this puzzling result is, that the belief is consistent with the perception *within* the same time slice, but that the *effect of the action* on the *next* state is *not* modeled by a *direct link* from the action node to the next future state node. Instead time slices are linked only between state nodes. So the action effect on future states is *not directly* included in the model. Action effects enter the model only via the (simulation) environment and the perception of the model. So the effect of actions could not be seen in the model even when the model contains a state variable. This criticism is true for all variants of HMM (Fig. A1.4-A1.10). It is irrelevant for DBNs *with* action effect models (Fig. 10, Fig. A1.12).

3.3.2.2 DBN-Models with Action Model and Action Effect Prediction

As we discussed in 3.3.1 an *action effect model* is necessary when the properties of the model have to be decoupled from the embedding environment. This is the reason why we discuss these kinds of models here. As an example we implemented the task of the night watchman with a DBN including an action effect model (Fig. A2.05-A2.08). The *apriori* beliefs are modeled in Fig. A2.05. The door is perceived as *is closed* (Fig. A2.06). Then the agent selects the action *push* (Fig. A2.07). The belief for the next future state is predominantly that the door *is open* then. Despite that belief the night watchman tries a second glance and sees that the door *is* still *closed* (Fig. A2.08). Now he revises his belief about the state again. He believes that the door *is closed and locked.* He should check then that belief by a *push action.*

3.3.2.3 Expert-Role, Mixture-of-Behavior, or Schema DBN Model

To the model in 3.3.2.2 we added the possibility that the agent is able to show role-specific or schematic behavior. We call these models Expert-Role, Mixture-of-Behaviors, or Schema Models (Fig. A2.09). *Top-down* generation of goal-based behavior is possible, when the role node gets evidence by selecting the role or the goal to generate role or schema-specific *behavior*. Furthermore, the model can be used *bottom-up* to infer the role, the *behavior*, or the goal from the percepts and/or the actions. For instance, when the agent *pushes* and *unlocks* the door despite his perception that the door *is closed*, we infer that he is either a *technician* or a *detective* but *not* a *night watchman* (Fig. A2.10).

3.3.2.4 AoI and *Ambient Vision-Role-Model*

Next according to Horrey et al. [71] we separated the visual perception into two components: (1) *foveal areas of interest* (AoIs) and (2) *ambient vision* (Fig. A2.11). If the agent is only interested in the *keyhole* of the door, we infer that he is active in the *detective* role (Fig. A2.12). The *plausible* actions in the *actual* time slice are *push, explore*, and *unlock*, the *expected* actions in the *next future* time slice are only *push* and *explore*. If we know that the model by its ambient vision component perceives that the door *is open* (Fig. A2.13), we expect that the agent is still in the *detective* role but with the different action *explore*. If we know for sure that the agent has the same perception as before but is in the role *night watchman* we expect his role-specific behavior is *shut* for the *actual* time slice and *lock* and *go on* in the *next two* future time slices (Fig. A2.14). When he *shuts* the door but is in the next time slice interested in the *door hinges*, we infer a *role, behavior,* or *intention conflict* because he might be also in the *role* of a *technician* (Fig. A2.15).

3.3.2.5 Reactive State-Based BAD-MoB-Model with Attention Allocation

Now we return to the driving domain. We developed a NETICA [75] model for a simple scenario with 3 maneuvers and 3 areas of interest (AoIs) (Fig. 12-13).

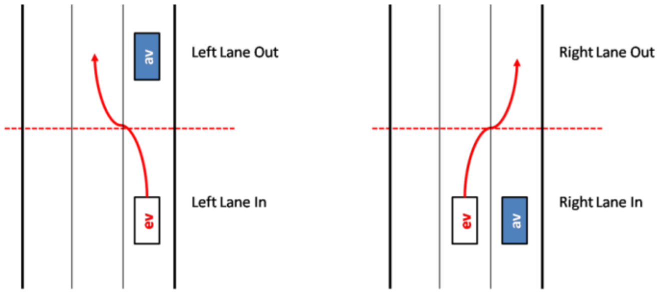


Fig. 12: Left and Right Lane Change Maneuvers

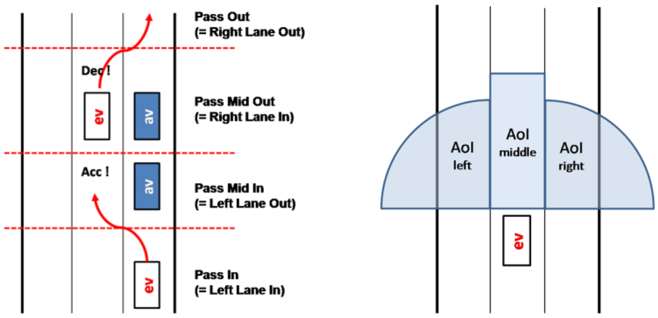


Fig. 13: Pass Vehicle or Overtake Maneuver (left) and AoIs viewed from ego vehicle (right)

A table describing the model representing levels of competence can be found in Fig. 14. The driver and the BAD model are sitting in the *ego* vehicle (ev). Sometimes the driver's perception signals evidence that the AoIs *is_occupied* depending on the *position* of an *alter* vehicle (av) or the *roadside*.

Levels of Expertise	**Model Component**	**Hierarchy of Skills, Levels of Expertise**
Skills		**Skills** = {…, drivingScenarioSkills, …}
Scenario Skills		**DrivingScenarioSkills** = { highway, countryRoad, city }
Driving **Maneuver** Skills	Driving **Maneuver** *Sequence* (horizontally distributed)	**highway.Maneuvers** = { leftLaneChange (lLC), rightLaneChange (rLC), passVehicle (pV), newManeuver }
Driving **Behavior** Skills	Driving **Behavior** *Layer*	**Behaviors** = { leftLaneIn (lLI), leftLaneOut (lLO), passIn (pI), passMidIn (pMI), passMidOut (pMO), passOut (pO), rightLaneIn (rLI), rightLaneOut (rO), newBehavior } e.g.: **leftLaneChange.Behaviors** = {leftLaneIn, leftLaneOut }
Driving **Action** Skills	Driving **Actions** *Layer*	**Actions** = { leftCheckLane (lCL), leftSignal (lS), leftTurn (lT), middleAcceleration (mA), middleDeceleration (mD), middleLookForward (mLF), rightCheckLane (rCL), rightSignal (rS), rightTurn (rT) } e.g.: **leftLaneIn.Actions** = {lCL, mD, mLF, lS, lT, mA}

Fig. 14: Hierarchy of Driving Skills, Levels of Expertise, and Model Components

The 2-time-slices template of the Dynamic *Reactive* BAD-MoB-Model is shown in Fig. 15 and a 3-time-slices rolled-out instance of that template model in Fig. 16. We call the model *reactive* because the AoIs *directly* influence actions. The model embeds two naïve Bayesian classifiers: One for the *Behaviors* and one for the *States.* This simplifies the structure of the architecture. Time slices are selected so that in each new time slice a new *behavior* is active. A *sequence* of behaviors implements a single *maneuver*. When we replace the reactive submodel for the *Action* variable in Fig. 15 by a *third* classifier we can simplify the model and reduce the number of parameters by 78%.

Behaviors are placed in the top layer of nodes (Fig. 15, 16). We have *behaviors* for each main part of a maneuver (Fig. 12-13): *left_lane_in, left_lane_out, pass_in, pass_mid_in, pass_mid_out, pass_out, right_lane_in, right_lane_out*. The next layer of nodes describes the *actions* the model is able to generate: *left_check_lane, left_signal, left_turn, middle_straight_accelerate, middle_straight_decelerate, middle_straight_look, right_check_lane, right_signal, right_turn*.

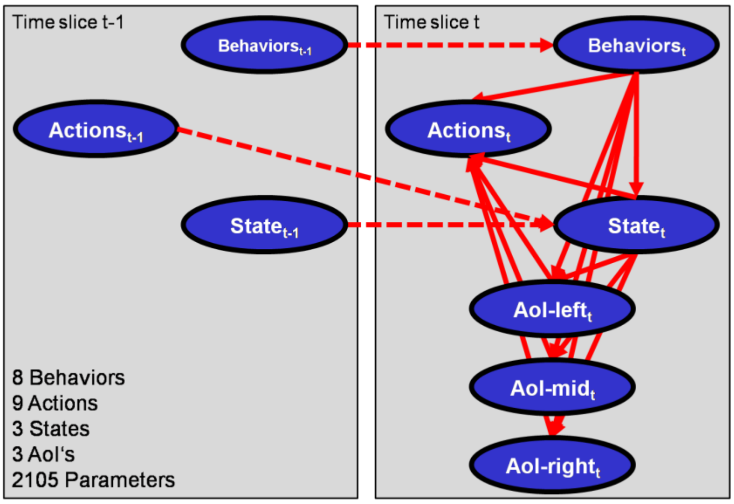


Fig. 15.1: Dynamic *Reactive* BAD-MoB-Model with *Behavior* and *State* Classifiers

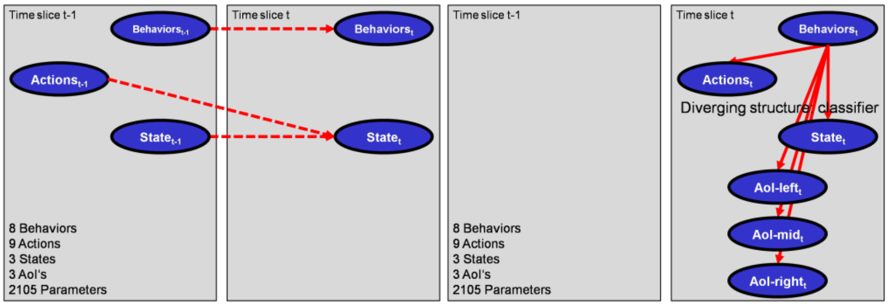


Fig. 15.2, 15.3: *Dynamic* Submodel (left) and *Behavior* Classifier (right)

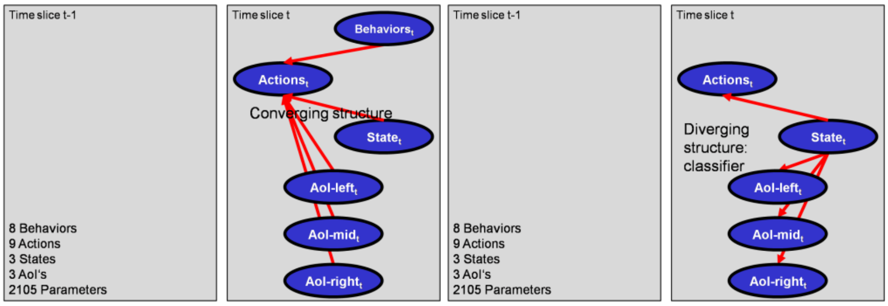


Fig. 15.4, 15.5: *Reactive Action* Submodel (left) and *State* Classifier (right)

Below that layer a layer of nodes is describing the *state* (*is_in_left_lane, is_in_middle_lane, is_in_right_lane*) of the vehicle. In the future these state nodes should be augmented by *tau*-, and *tau-dot*-variables describing the driver's state [73, 74]. The three bottom layers contain nodes describing the activation of the three AoIs *AoI_Left, AoI_Middle,* and *AoI_Right*.

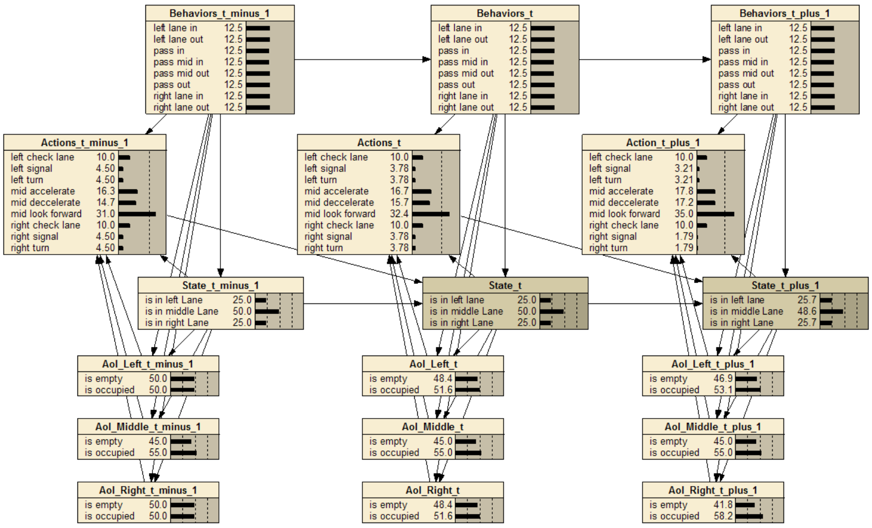


Fig. 16.1: *Reactive* State-Based BAD-MoB-Model with 2 Classifiers and 2 Levels-of-Expertise

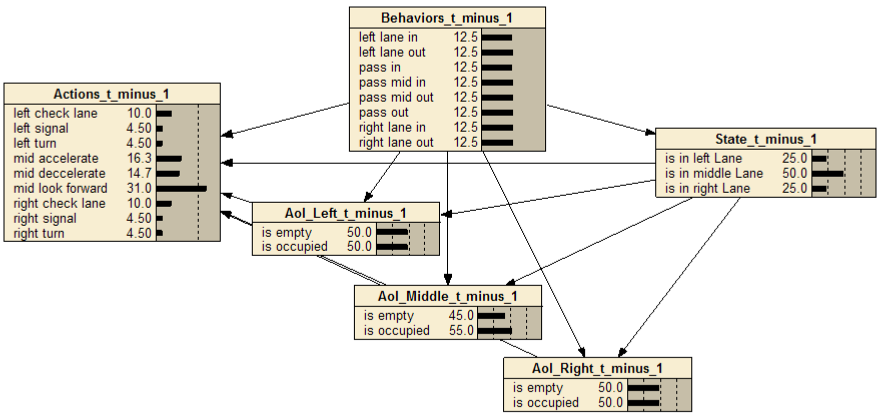


Fig. 16.2**:** Blown-up nodes of time-slice t-1 in NETICA-Model of Fig. 16.1

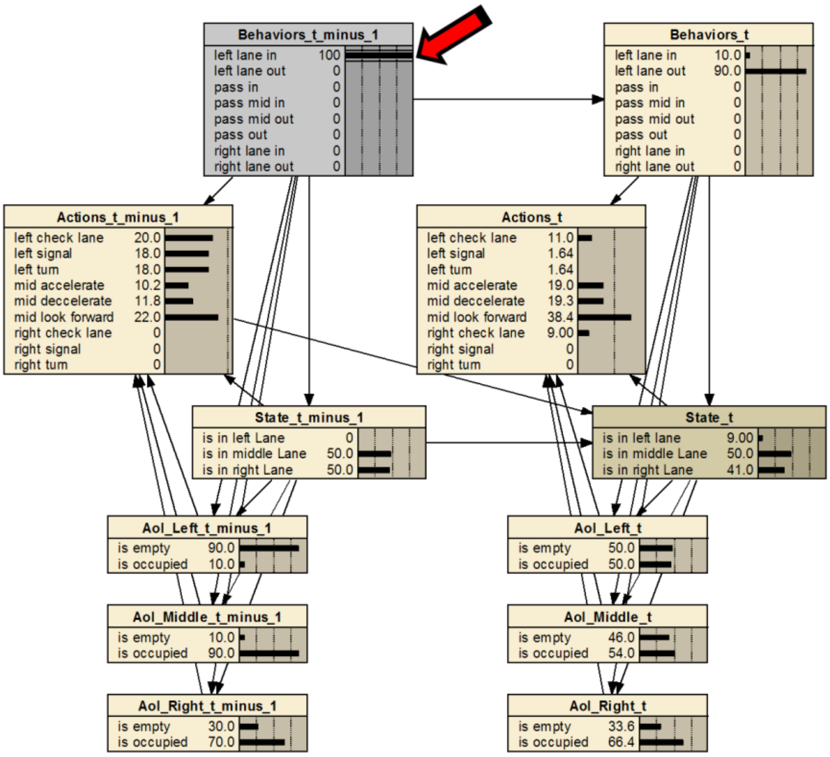


Fig. 17: *Expectations* when BAD-MoB model is *in left_lane_in behavior*

When the model is urged to be in the *left_lane_in behavior* by e.g. goal setting from the associative layer (Fig. 17, red arrow), we expect in the *same* time-slice that the driver though sometimes *looking forward* his behavior is focused towards the *left* lane. For the AoIs we expect that the middle and right AoI *are occupied* and the left AoI *is empty*. For the *next* time slice we expect the vehicle is *in the left* or *middle lane* and the driver will act according *left_lane_out* behavior. *Left_lane_out* activated actions in time slice t are a bit different than those before. We expect more forward orientated activities like (de-/*ac-)celebration* and *forward* directed attention.

When the *state* is known (e.g. S = *is_in_middle*_lane) we infer the appropriate expectations (e.g. *left and right lane check*, *looking forward*, and both *(ac|de)celerations*) (Fig. 18).

When the model perceives a combination of AoI evidence, we can infer the *behaviors*. For instance, in Fig. 19 the left AoI *is empty* and the middle AoI *is occupied*. We expect that the vehicle *is in the middle* or *right lane* and that the *behaviors* are ambiguous *left_lane_in* or *pass_in*. Their appropriate *actions* are activated as a *mixture of behaviors*. The most probable actions are *mid look forward*, *left check lane* and *middle deceleration*.

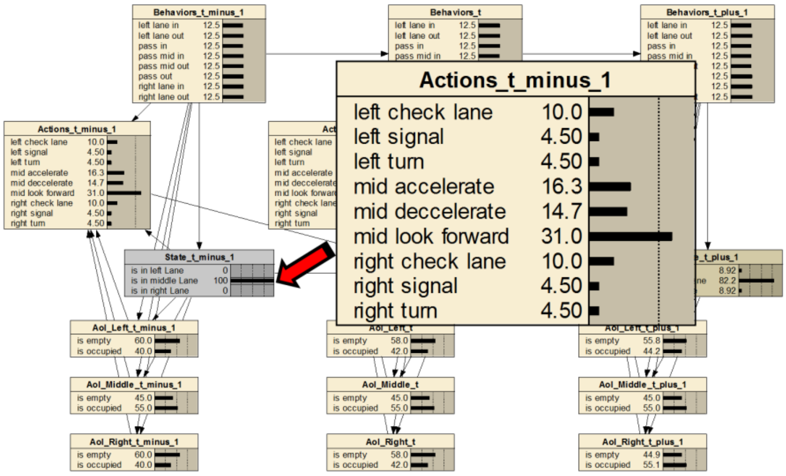


Fig. 18: *Expectations* when BAD-MoB model *is_in_middle_lane State*

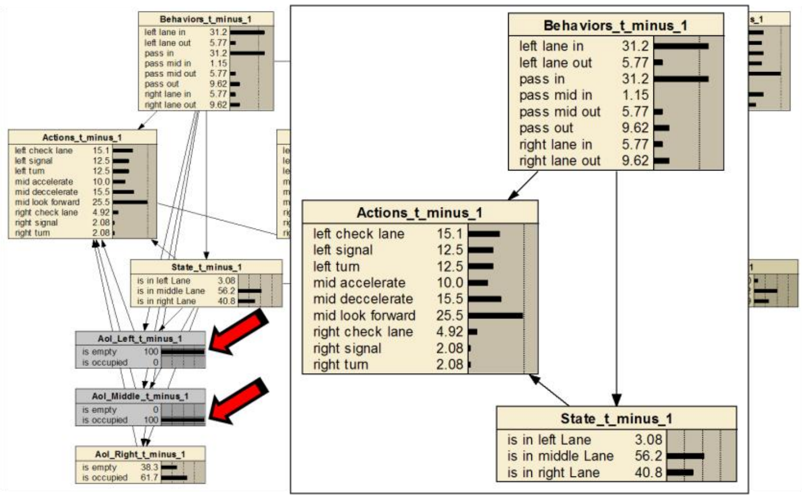


Fig. 19: *Expectations* when BAD-MoB model perceives a combination of AoI evidence

In the case, when all AoIs are occupied (Fig. 20) the model *is decelerating* with main attention to the middle AoI (*middle_look_forward*). We call this focusing of attention and narrowing of the attended vision field (sometimes under stress) *Tunnelblick* (tunnel view or tunnel vision[4]).

What will happen, if a *goal* is blocked? In Fig. 21 this situation is modeled by the appropriate evidence.

[4] In medical terms, *tunnel vision* is the loss of peripheral vision with retention of central vision, resulting in a constricted circular tunnel-like field of vision (http:// en.wikipedia.org/wiki/Tunnel vision, visited 1st March, 2010)

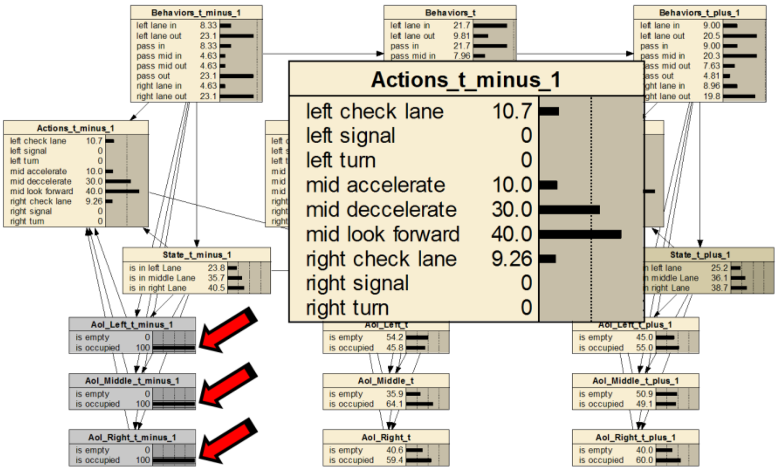


Fig. 20: Expectations when BAD-MoB model perceives that all AoIs are *occupied*: Tunnelview

The *left_lane_in behavior* is provided with evidence because we assume that a corresponding *goal* in a higher cognitive layer is activated. At the same time the perception all AoIs is set to *is_occupied*. The expected behavior is *looking forward, deceleration, and left_check_lane*, which are indicators for the *Tunnelview* and (helplessly?) looking to the left.

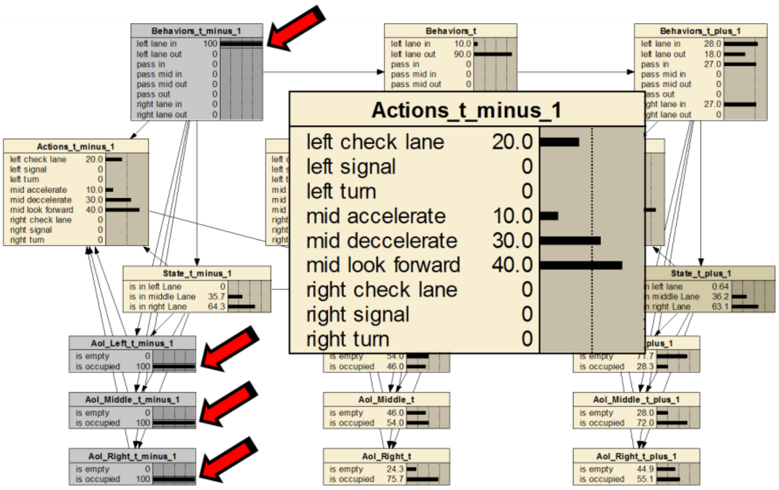


Fig. 21: *Expectations* when BAD-MoB model realizes a *blocking of goals or behaviors* by a combination of *occupied* AoIs: *Tunnelview*

With the rolled-out version of the BAD-MoB-Model it is possible to anticipate hazards (Fig. 22). The anticipated hazard is included as percept evidence in time-slice (t+1). *Conditional* to the current state (t-1), the *anticipated* percept evidence (t+1) of the hazard, the *proactively* selected goal-behavior *left_lane_in* (t-1, t), and the *proactively* selected action *left_turn* (t-1, t), we predict that we are able to avoid the hazard in time-slice (t+1) (Fig. 23).

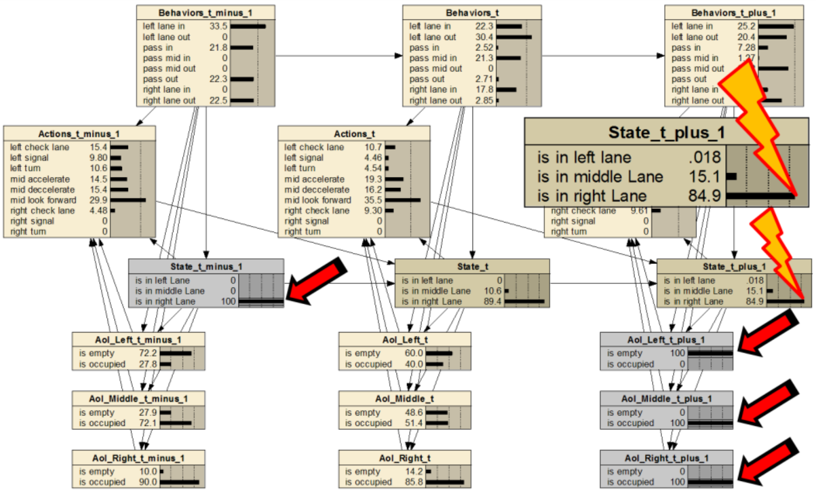


Fig. 22: *Anticipation of hazards*: the BAD-MoB model anticipates in time-slice (t-1) a hazard (relative to the current *state*) for time-slice (t+1)

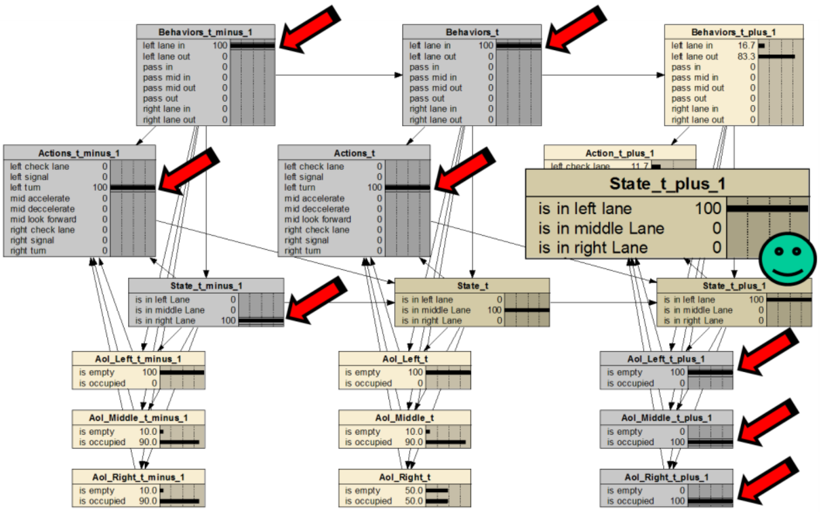


Fig. 23: *Anticipatory Plan*: the BAD-MoB model sets as goals the behavior *left_lane_in* and selects the *left_turn* action for time-slices (t-1) and t to avoid the hazard in time-slice (t+1)

We believe that the proof of concept is convincing: *state-based* BAD-MoB Models are expressive enough to describe and predict a wide range of phenomena including *prediction of hazards*, *anticipatory planning*, and planning of *minimal invasive countermeasures*.

4 Experimental Results

4.1 Use Case for *Autonomous* Driving: A Simple BAD Model

Static reactive or *static inverse* models (Fig. A1.01-A1.03) have not been satisfactory because they generate behavior which is more erratic and nervous than human behavior is [21]. Better results can be obtained by introducing a memory component and using DBNs. In a first step we estimated two DBNs separately for the lateral and longitudinal control. Our experience is that *partially inverse* models are technically well suited for modelling in the driving domain. (Fig. 24, 25).

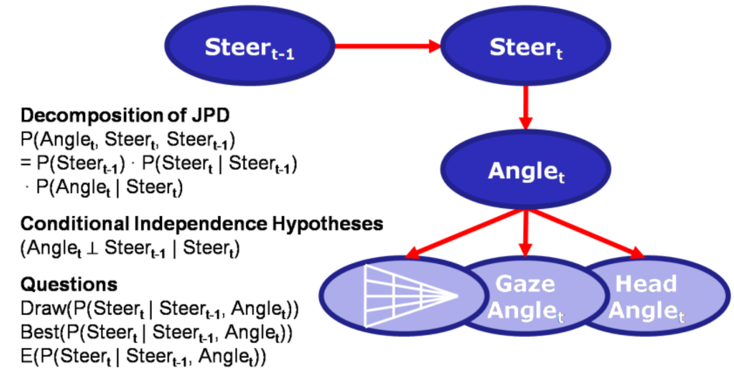


Fig. 24: *Partially inverse* classifier-*based* DBN of Lateral Control (graphics from [22] with kind permission of Springer Science and Business Media)

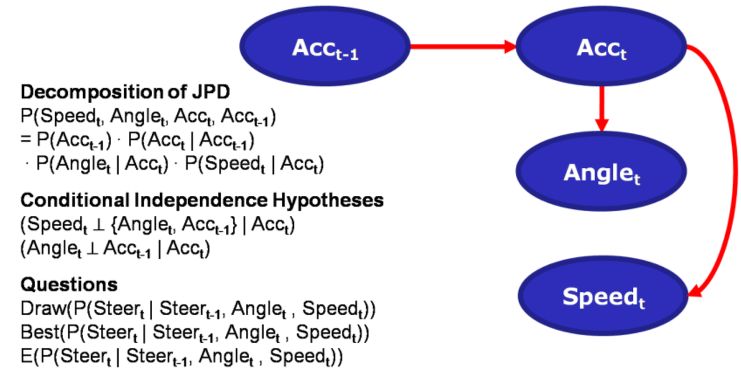


Fig. 25: *Partially inverse classifier-based* DBN of Longitudinal Control (graphics from [22] with kind permission of Springer Science and Business Media)

In an *inverse* model arcs in the directed acyclic graph (DAG) of the graphical model are directed from the *consequence* to the *prerequisites*. The semantics of these arcs are denoted by the conditional probabilities $P(Prerequisites \mid Consequence)$.

The reasons to use *inverted* $P(Prerequisites \mid Consequence)$ instead of *reactive* conditional probabilities $P(Consequence \mid Prerequisites)$ are the possible large number of prerequisites in a reactive model.

By using *inverted* conditional probability distributions, we significantly reduce the amount of parent nodes for *Consequence*. Furthermore, a conditional probability $P(Prerequisites \mid Consequence)$ is more robust to possible *unknown* evidence of *Prerequisites.* This occurs almost inevitably as missing sample data because it is rather unlikely to obtain all possible values of the joint probability distribution $P(Consequence, Prerequisites)$. Our models are *partially* inverse because most arcs are inverted but the arcs between time slice t-1 and t are in *causal order* from prerequisites to consequences.

The variables of interest are partitioned into *sensor*-variables and *action*-variables. The variables for the partially inverse DBN of *lateral* control are defined as follows: $Steer_t$ and $Steer_{t-1}$ can take 21 different values between -10 (hard right) and +10 (hard left). Variable $Angle_t$ represents the angle between heading vector of the car and the course of the racing track to be reached in 1 second by current speed and can take 21 values between -10 (large positive angle) and +10 (large negative angle). According to Fig. 24, the decomposition of their JPD is specified as:

$$P(Steer_{t-1}, Steer_t, Angle_t) =$$
$$P(Steer_{t-1}) \cdot \; P(Steer_t|Steer_{t-1}) \cdot \; P(Angle_t|Steer_t).$$

According to the visual attention allocation theory of Horrey et al. [51] the perception of the heading angle is influenced by areas in the visual field (*ambient* channel), the head angle and the gaze angle relative to the head. At the present moment light colored nodes in Fig. 24 are not included into the driver model. Instead we assumed that drivers are able to compute the aggregate sensory variables *heading angle* and *vehicle speed*. Compared to the lateral control in Salvucci & Gray's model [15, 52] our BAD model is more robust, makes less assumptions about the vision field, and *no* assumptions about gaze-control.

Variables Acc_t and Acc_{t-1} of DBN of *longitudinal* control take 21 different values between -10 (fully depress braking pedal) and +10 (fully depress acceleration pedal). Variable $Angle_t$ represents the angle between heading vector of the car and the course of the racing track to be reached in 2 second by current speed and can take 21 values between -10 (large positive angle) and +10 (large negative angle).

Variable $Speed_t$ represents the perceived longitudinal velocity and takes 10 values between 0 (low speed) and 10 (high speed). The decomposition of their JPD is specified as:

$$P(Acc_{t-1}, Acc_t, Angle_t, Speed_t) =$$
$$P(Acc_{t-1}) \cdot \; P(Acc_t|Acc_{t-1}) \cdot \; P(Angle_t|Acc_t) \cdot \; P(Speed_t|Acc_t).$$

All terms of the two decompositions are assumed to have a Gaussian form, whose parameters mean μ and standard deviation σ need to be obtained from experimental data.

4.1.1 Experimental Settings

To demonstrate the functionality of our BAD models for autonomous lateral and longitudinal control we use the open source racing simulation TORCS [20]. Though considered as a racing game, TORCS accurately simulates car physics and allows the user to implement personal driver models. A driver model controls a vehicle within the TORCS world by action parameters (steering, accelerating, braking etc.) and has access to the current world state of the TORCS simulation. We developed a driver model, referred as *TORCS driver model*, which is capable to derive action parameters by values read from external controllers, read/write experimental data from/into files and perceive its environment according to the perception component of the BAD model. The BAD model itself is embedded in the TORCS driver model (Fig. 26). For implementation and inference of the BAD model we use ProBT©, a Bayesian inference engine and an API for building Bayesian models. ProBT© is published by the ProBAYES company and free available for academic purposes.

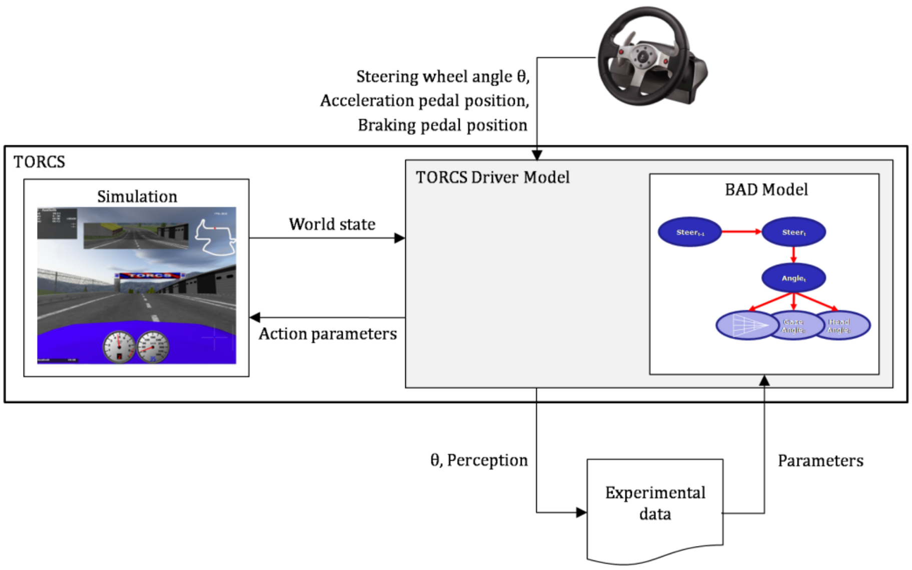


Fig. 26: Overview embedding the BAD model in the TORCS driver model

As external controller we use the Logitech G25, a controlling device consisting of a force-feedback steering wheel, pedals and a gear box. A human driver can manually control the TORCS vehicle via the steering wheel angle θ_t and the positions of acceleration- and braking-pedal. To achieve a usable drivability, operative steering wheel angles are limited to thirty percent of the possible steering wheel angles, leading to effective vehicle steering angles between $-13.5°$ and $+13.5°$. Greater steering wheel movements were possible but would not affect the actual vehicle steering angle.

4.1.2 Recording of Experimental Data

Data were obtained in experimental drives of a single driver on the TORCS racing track "Aalborg". The map of the drive and curve specific measurements are presented in Fig. 27. For that purpose, time series of values of *sensory*- and *action*-variables were recorded at an interval of 50ms. The experimental data were used to obtain parameters (means, standard deviations) for the Gaussian parametric forms for each of probability distribution of the BAD model. A single driven lap was sufficient to obtain the data for estimating parameters stable enough for autonomous control.

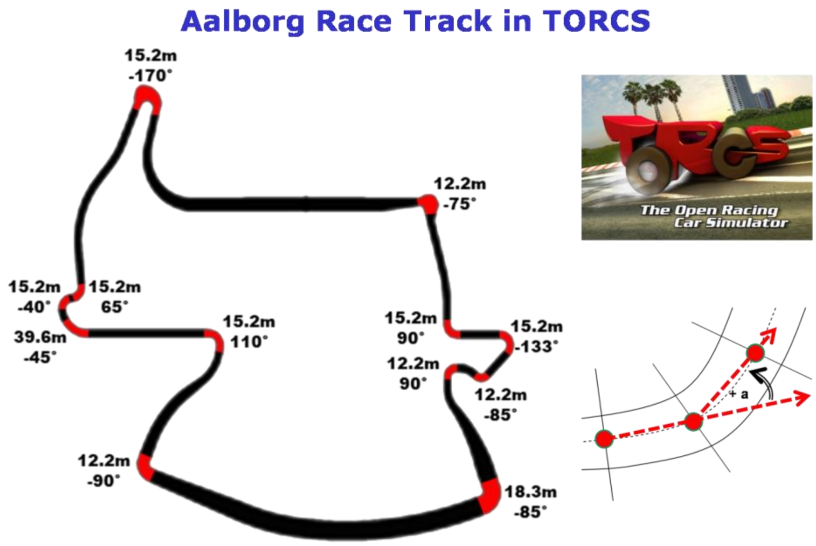


Fig. 27: Bird's eye view of race track with curve radii and rotation angles (graphics from [22] with kind permission of Springer Science and Business Media)

4.1.3 Autonomous Driving of BAD model

Under BAD-Model-control of the vehicle current values for *sensory*-variables of lateral and longitudinal control are sampled every time step t. After inferring the conditional probability distributions $P(Steer_t|steer_{t-1}, angle_t)$ of the lateral control DBN and $P(Acc_t|acc_{t-1}, angle_t, speed_t)$ of the longitudinal control DBN, concrete values for $Steer_t$ and Acc_t were randomly drawn from these distributions and used to control the TORCS vehicle. In principle we could choose the conditional expected actions to make the model react deterministically $E(Action_t|steer_{t-1}, angle_t)$.

4.1.4 Results

A snapshot of a BAD model drive is shown in Fig. 28 and a comparison of the driver's speed data with model generated speed data in Fig. 29. The comparison demonstrates the quality of the simple BAD model but due to collisions with the roadside it is apparent that the capabilities of the BAD model have to be improved.

Improvements are expected by combining the two controllers, by including cognitive constructs like goals and latent states of the driver, and above all segmenting *maneuvers* into context dependent schemas (= *behaviors*).

Using goals (e.g. driving a hairpin or an S-curve) makes it possible to adapt the model to different road segments and situations. We try to use the same model for situation recognition or to situation-adapted control. The modeling idea of a HMM was abandoned because the state variable has to be too fine grained to obtain a high quality vehicle control [23]. Instead we are guided by the idea of state-based *mixture-of-behaviors* models in 3.3.2.5.

Fig. 28: Snapshot of BAD model drive on TORCS race track (graphics from [22] with kind permission of Springer Science and Business Media)

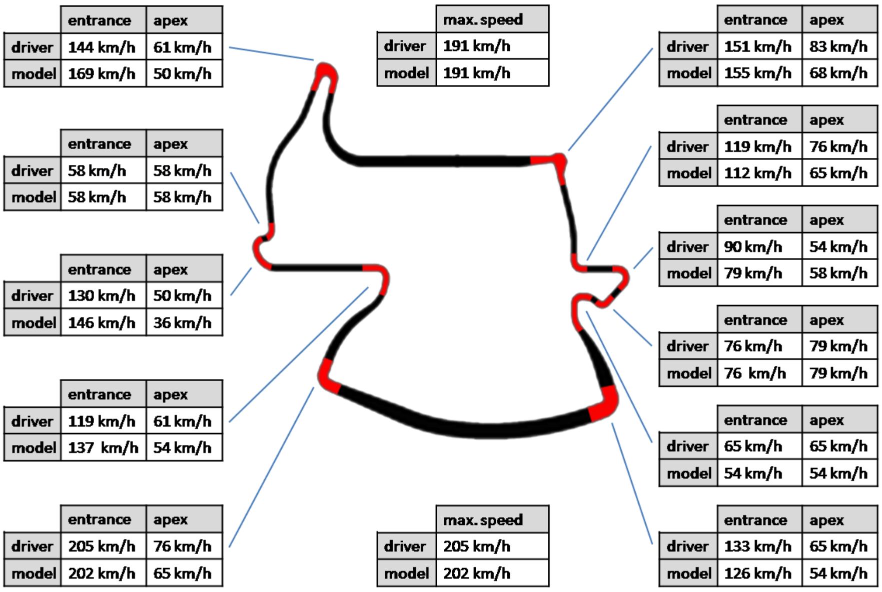


Fig. 29: Comparison of Human and BAD-Model Drives (graphics from [22] with kind permission of Springer Science and Business Media)

4.2 Use Case for In-Vehicle Cooperation: Driving Under the Lateral Assistance of a BAD Model

As an example for smart assistance, we present the use of a BAD model to assist a human driver's *lateral control*. We decided to change the former perception component of the BAD model from heading angle, represented by variable $Angle_t$ in Fig. 24 and 25, into twenty distance sensors, represented by variables S_t^0 to S_t^{19}. Positioned at the headlights of a car the distance sensors simulate according to the vision theory of Horrey's et al. [60] an *ambient vision field* with radius 105° (Fig. 30).

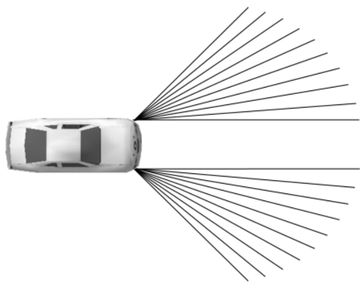


Fig. 30: Ambient Vision Field - Position and direction of distance sensors.

The variables of this BAD model are defined as follows: $Steer_t$ and $Steer_{t-1}$ can take 21 different values between -10 (hard left) and +10 (hard right). Each of the variables S_t^0 to S_t^{19} can take 20 different values between 0 (short distance) and 19 (long distance). The JPD is decomposed to

$$P(Steer_{t-1}, Steer_t, S_t^0, \ldots, S_t^{19}) = P(Steer_{t-1}) \cdot P(Steer_t | Steer_{t-1}) \cdot \prod_{i=0}^{19} P(S_t^i | Steer_t).$$

The graphical representation of this decomposition is shown in Fig. 31.

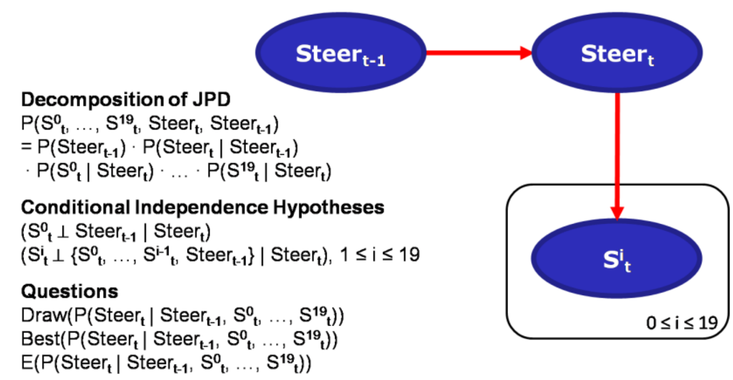


Fig. 31: *Partially inverse* DBN of Lateral Control for assisted driving

4.2.1 Experimental Setting

To demonstrate the functionality of the BAD model for assisted driving the open source racing simulation TORCS [20] and the Logitech G25 as external controller are used once again.

For lateral assistance an extension of the TORCS driver model was necessary (see Fig. 32). To assist the human driver, the BAD model not only must be able to control the simulated TORCS vehicle but also to influence the steering wheel angle θ_t. While the human driver can influence θ_t simply by turning the steering wheel in the ordinary manner, the BAD model has to control the steering wheel in a different way by applying *force-feedback* commands. These commands are realized by a force-feedback *spring effect* that pushes the steering wheel back toward a certain position φ after it has been moved from that position. The strength of the reset force is determined by a function $f(\varphi_t - \theta_t)$. The variables influencing the effect can be adjusted and therefore can be used to parameterize the strictness and amount of BAD assistance. An overview of the resulting structure is given in Fig. 32.

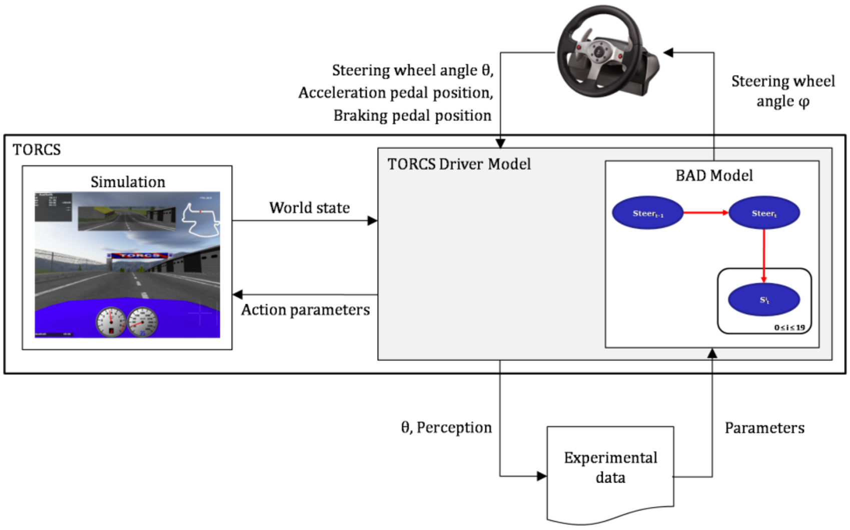


Fig. 32: TORCS driver model with BAD model for lateral assistance

4.2.2 Recording of Experimental Data

To collect the experimental data needed to determine the parameters to define the BAD model, three laps were driven by the second author once again on the racing track "Aalborg" (Fig. 27). The experimental data was then used to construct (conditional) probability tables for each of probability distribution of the BAD model. If the probability tables matched the shape of a Gaussian they were discretized with mean μ and standard deviation σ.

4.2.3 Driving under Smart Assistance of the BAD Model

While running, at an interval of 50 ms, the TORCS driver model calculates the values for each of the distance sensors to derive current values in the BAD model for $S_{0,t}$ to $S_{19,t}$. Knowing $Steer_{t-1} = steer_{t-1}$and $S_t^0 = s_t^0, \dots, S_t^{19} = s_t^{19}$, the conditional probability distribution $P(Steer_t | steer_{t-1}, s_t^0, \dots, s_t^{19})$ will be inferred by the BAD model. By now, the inferred conditional distribution is used for continuous (1.) and temporarily (2.) driving assistance:

1. We created a *highly-automated* approach to assisted driving, letting the BAD model automatically control the steering wheel while a human driver can choose to intervene. To achieve this, a value $steer_t$ was, according to the BP draw strategy, randomly drawn from the distribution $P(Steer_t | steer_{t-1}, s_t^0, \dots, s_t^{19})$ and used to calculate a new center angle φ for the force-feedback spring effect (steering wheel reset force).
2. As a *first* approach to *semi-automated* assisted driving we let a human driver control the vehicle while the BAD model only intervened when the steering movements had a significant low probability in the current situation. This was achieved by inferring $P(steer_t = \theta \, | steer_{t-1}, s_t^0, \dots, s_t^{19})$. Once this probability

falls below a certain threshold an *anomaly* is detected, therefore a new value $steer_t$ was randomly drawn from the distribution $P(Steer_t \mid steer_{t-1}, s_t^0, \dots, s_t^{19})$ and used to calculate a new center angle φ for the force-feedback spring effect (steering wheel reset force).

4.2.4. Results

The parameters derived from three driven laps turned out to be sufficient to create a BAD model that was able to assist a human driver. Furthermore, the level of assistance intensity can easily be shifted from rather light to very strict by simply adjusting the parameters of the spring effect and/or the threshold.

5 Bayesian Learning of Bayesian Agent Models

In order to learn the parameters of the CPDs of the BAD model in an objective manner a set of experimental data is needed. Learning can be done *offline* or *online*.

In *offline* learning as described in chapter 4 the collection of the training data and the testing of the BAD model are temporarily separated activities. Collecting experimental data without real-time reviewing the behavior of the BAD model allows only delayed information about its performance. Furthermore, due to the fact that $P(Steer_t | steer_{t-1}, s_t^0, \dots, s_t^{19})$ has to be inferred *inversely* (Fig. 24, 25), an inspection of the probability distributions of the BAD model is not very informative how to obtain the intended behavior and how to improve the completeness and quality of the model. Offline adapting the BAD model remains a clumsy and subjective procedure similar to the handcrafting of production system models.

A more natural approach would be the *online* learning of the BAD models by Bayesian parameter learning. We propose a *new* methodology: *Bayesian learning of agent models under human control*. The performance of the BAD model is observed by the human driver while the BAD model is driving. New data are learned only when the model behavior is unsatisfying. By *observing and correcting* the actions of the BAD model *only when needed*, problems can be solved, which are nearly impossible to discover by just analyzing its probability distributions. According to Bayesian methodology the old unsatisfactory BAD model is contained in the apriori-hypothesis, which will be revised by new training data to the aposteriori-hypothesis which contains the improved model.

We extended the TORCS driver model to provide the human driver with a learning control in the case of unsatisfactory BAD model behavior. New human experimental data are recorded by pressing a *learning button* attached to the steering wheel. When *learning* at every time step t current percepts provided by TORCS and actions read from the Logitech G25 controller are written into the database, updating the behavioral data and the CPDs. Once the button is pressed again, the data acquisition process is stopped and the conditional probability distributions are modified according to the Bayesian learning methodology.

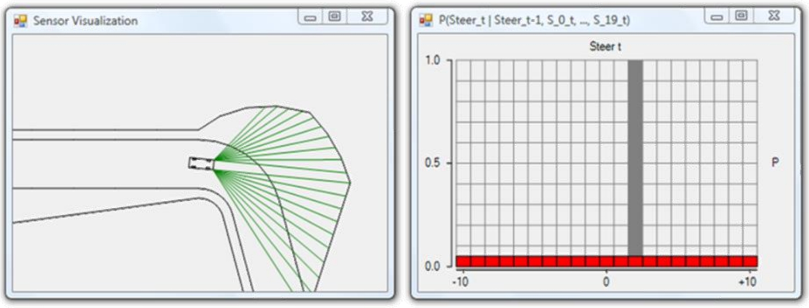


Fig. 33: Left: Runtime-visualization showing the driver model and its sensors while approaching a right curve from a bird's eye view. Right: Runtime-visualization of the corresponding apriori uniform conditional probability distribution (red squares) $P(Steer_t|steer_{t-1}, s_t^0, \dots, s_t^{19})$. Light gray bar shows the human chosen steering wheel angle $Steer_t = \theta_t$.

To test the functionality of this approach, we used an empty database to learn parameters for the BAD model. We started with uniform apriori distributions for each of the conditional probability distributions of the BAD model. At the beginning, the apriori CPD $P(Steer_t|steer_{t-1}, s_t^0, \dots, s_t^{19})$ was uniform and the driving behavior of the BAD model therefore completely random. Fig. 33 shows a screenshot of the TORCS driver model approaching a right curve and the corresponding apriori uniform CPD when starting with an empty database.

We then started collecting driving data while correcting the BAD model whenever its actions were not suitable to solve the current situation. The performance of the BAD model improved rapidly and it took only a few standard maneuvers to be able to let the BAD model drive the whole racing track successfully. As an example, Fig. 34 shows the driver model approaching the same right curve as showed in Fig. 33 after collecting experimental data of one driven right curve, resulting in a very peaked conditional probability distribution $P(Steer_t|steer_{t-1}, s_t^0, \dots, s_t^{19})$.

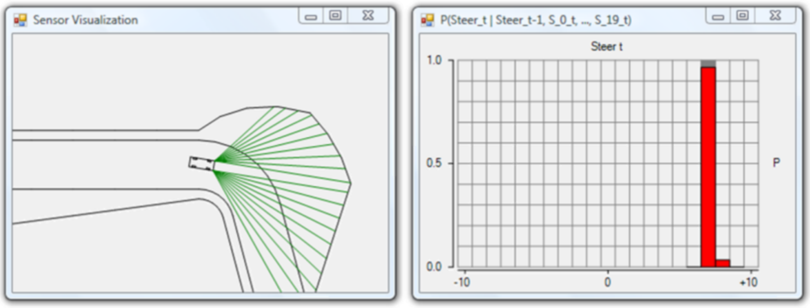


Fig. 34: Left: Runtime-visualization showing the driver model and its sensors while approaching a right curve from a bird's eye view. Right: Runtime-visualization of the corresponding Bayesian learned aposteriori conditional probability distribution (red squares) $P(Steer_t|steer_{t-1}, s_t^0, \ldots, s_t^{19})$.

6 The Conversion of BAD Models to Bayesian Assistance Systems

For the purpose of smart assistance in simulated or real world scenarios the obtained BAD models can be used as BAS in principle *as they are.* The only question is, whether the driving competence of the BAD model *is the same* as the driving competence of the human driver controlling the vehicle in the training session.

Our simulation world is so abstract that the sophisticated ambient human perception system can be simulated by a beam of sensors and sensor fusion. In more complicated scenarios we have to refine the model of the vision system [31].

We believe that our approach is superior to a proposal to model the strategic skills of a PADAS with a *Markov Decision Process (MDP)* [18]. A MDP needs a reward function. This function has to be derived deductively from theoretical concepts or learnt inductively from car trajectories by solving the *inverse reinforcement learning problem* [19]. The deductive derivation of reward function often results in strange nonhuman overall behaviors. The inductive mining of the reward function from car trajectories or behavior traces seems to be a detour and seem more challenging than our approach.

The two new concepts *Bayesian learning of agent models under human control* and the *usage of a BAD model as a BAS or PADAS* are demonstrated here and in [25, 26].

7 Conclusion and Outlook

We think that dynamic probabilistic models are sufficient expressive to describe and predict a wide range of phenomena. Their subtypes BAD and BAD-MoB models are appropriate for the challenges described in this paper, especially when they are learnt in experiments with *Bayesian learning of agent models under human control*. Next we have to implement further models creating a library of behaviors of various levels of expertise. To that end a careful selected taxonomy of scenarios, maneuvers, behaviors, and control actions without and with alter agents has to be defined and studied. We believe that our approach to use a BAD model as a BAS or PADAS is superior to a proposal to model the strategical and tactical skills of a PADAS with a *Markov Decision Process (MDP).*

8 References

1. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21197 (visited 27th February, 2010)
2. Sarter, N., Amalberti, R.R., and Amalberti, R., Cognitive Engineering in the Aviation Domain, Lawrence Erlbaum Assoc. Inc., 2000
3. Norman, D.A.; The Design of Future Things, Basic Books, 2007
4. Zelkha, Eli; Epstein, Brian, From Devices to Ambient Intelligence, Digital Living Room Conference, June 1998 (http://www.epstein.org/brian/ambient_intelligence/DLR%20Final%20Internal.ppt ; visited 27th February, 2010)
5. Cacciabue, P.C. (ed) Modelling Driver Behaviour in Automotive Environments, London: Springer, ISBN-10: 1-84628-617-4 (2007)
6. http://www.hmat-ws.eu/ (visited 27th February, 2010)
7. Löper, Ch., Kelsch, J., Flemisch, F.O., Kooperative, Manöverbasierte Automation und Arbitrierung als Bausteine für hochautomatisiertes Fahren, in: Gesamtzentrum für Verkehr Braunschweig (Hrsgb): Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel, GZVB, Braunschweig, 2008, S. 215-237
8. Flemisch, F., Schieben, A., Kelsch, J., Löper, Chr., and Schomerus, J., Cooperative Control and Active Interfaces for vehicle assistance and automation, FISITA 2008, http://elib.dlr.de/57618/01/FISITA2008_DLR_FlemischEtAl_CooperativeControl.pdf (visited 27th February, 2010)
9. http://www.kognimobil.net/index.php (visited 27th February, 2010)
10. Yangsheng Xu, Ka Keung Caramon Lee, and Ka Keung C. Lee, Human Behavior Learning and Transfer, CRC Press Inc., 2005
11. Jürgensohn, Th.: Control Theory Models of the Driver, in: Cacciabue (ed), 2007, p. 277 – 292
12. Weir, D.H. & Chao, K.C.: Review of Control Theory Models for Directional and Speed Control, in: Cacciabue, P.C., p. 293 – 311 (2007)
13. Salvucci, D.D., & Gray, R.: A Two-Point Visual Control Model of Steering, Perception, 33, 1233 – 1248 (2004)
14. Gluck, K.A.; & Pew, R.W.; Modeling Human Behavior with Integrated Cognitive Architectures, Mahwah, N.J.: Lawrence Erlbaum Associates, 2005
15. Möbus, C., Hübner, S., Garbe, H., *Driver Modelling: Two-Point- or Inverted Gaze-Beam-Steering,* in M. Rötting, G. Wozny, A. Klostermann und J. Huss (eds), Prospektive Gestaltung von Mensch-Technik-Interaktion, Fortschritt-Berichte VDI-Reihe 22, Nr. 25, 483 – 488, Düsseldorf: VDI Verlag, 2007, ISBN 978-3-18-302522-0
16. Salvucci, D.D.: Integrated Models of Driver Behavior, In: Gray, W. D. (Ed.) *Integrated models of cognitive systems*. New York: Oxford University Press, p. 356–367 (2007)
17. Baumann, M., Colonius, H., Hungar, H., Köster, F., Langner, M., Lüdtke, A., Möbus, C., Peinke, J., Puch, S., Schießl, C., Steenken, R., Weber, L., Integrated Modeling for Safe Transportation - Driver modeling and driver experiments, in Th. Jürgensohn, H. Kolrep (Hrsg.), Fahrermodellierung in Wissenschaft und Wirtschaft, 2. Berliner Fachtagung für

Fahrermodellierung, Fortschrittsbericht des VDI in der Reihe 22 (Mensch-Maschine-Systeme), Nr.28, 84-99, VDI-Verlag, 2009, ISBN 978-3-18-302822-1

18. Raghav Aras, Fabio Tango, and Olivier Pietquin, Learning An Optimal Warning And Intervention Strategy For A PADAS (paper submitted to HMAT2010, http://www.hmat-ws.eu/) (visited 27th February, 2010)
19. Abbeel, P. and Ng, A.Y., Apprenticeship learning via inverse reinforcement learning, ACM International Conference Proceeding Series; Vol. 69, Proceedings of the twenty-first international conference on Machine learning, Banff, Alberta, Canada, 2004
20. TORCS, http://torcs.sourceforge.net/ (visited 27th February, 2010)
21. Möbus**,** C., Eilers, M. (2008). First Steps Towards Driver Modeling According to the Bayesian Programming Approach, Symposium Cognitive Modeling, p.59, in: L. Urbas, Th. Goschke & B. Velichkovsky (eds) *KogWis 2008*. Christoph Hille, Dresden, ISBN 978-3-939025-14-6
22. Möbus, C., Eilers, M. (2009a). Further Steps Towards Driver Modeling according to the Bayesian Programming Approach, in: *Conference Proceedings, HCII 2009, Digital Human Modeling*, pp. 413-422, LNCS (LNAI), Springer, San Diego, ISBN 978-3-642-02808-3
23. Möbus, C., Eilers, M. Garbe, H., and Zilinski, M. (2009b). Probabilistic and Empirical Grounded Modeling of Agents in (Partial) Cooperative Traffic Scenarios, in: *Conference Proceedings, HCII 2009, Digital Human Modeling*, pp. 423-432, LNCS (LNAI), Springer, San Diego, ISBN 978-3-642-02808-3
24. Möbus, C., Eilers, M. Zilinski, M. Garbe, H. (2009c). *Mixture of Behaviors in a Bayesian Driver Model*, in: Lichtenstein, A. et al. (eds), Der Mensch im Mittelpunkt technischer Systeme, p.96 and p.221-226 (CD), Düsseldorf: VDI Verlag, ISBN 978-3-18-302922-8, ISSN 1439-958X
25. Möbus, C. Eilers, M. (2010). Mixture-of-Behaviors and Levels-of-Expertise in a Bayesian Driver Model, (paper accepted) 1st International Conference of Applied Digital Human Modelling, Miami, USA
26. Eilers, M., Möbus, C., Learning of a Bayesian Autonomous Driver Mixture-of-Behaviors (BAD-MoB-)Model, (paper accepted) 1st International Conference of Applied Digital Human Modelling, Miami, USA
27. Neapolitan, R.E., (2004), Learning Bayesian Networks, Upper Saddle River: Prentice Hall
28. Pearl, J. (2009). Causality – Models, Reasoning, and Inference, 2nd ed., Cambridge University Press, ISBN 978-0-521-89560-6
29. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics, Cambridge, Mass.: MIT Press (2005)
30. Bessiere, P., Survey: Probabilistic Methodology and Techniques for Artifact Conception and Development, Rapport de Recherche, No. 4730, INRIA, 2003
31. Lebeltel, O., Bessiere, P., Diard, J. & Mazer, E.: Bayesian Robot Programming**,** Autonomous Robots 16, 49 – 79 (2004)
32. Le Hy, R., Arrigoni, A., Bessière, P., Lebeltel, O.; (2004); Teaching Bayesian Behaviours to Video Game Characters; Robotics and Autonom. Systems (Elsevier), Vol. 47: 177-185
33. Bessiere, P., Laugier, Ch., & Siegwart, R., (eds.) Probabilistic Reasoning and Decision Making in Sensory-Motor Systems, Berlin: Springer, ISBN 978-3-540-79006-8 (2008)
34. Hutchins, E.: Cognition in the Wild, Cambridge, Mass.: MIT Press (1995)
35. Xiang, Y.: Probabilistic Reasoning in Multiagent Systems - A Graphical Models Approach, Cambridge University Press, Cambridge (2002)
36. http://www.hamilton-baillie.co.uk/ (visited 27th February, 2010)
37. http://www.sharedspace.eu/en/home/ (visited 27th February, 2010)
38. Rizzo, Matthew; McGehee, Daniel V.; Dawson, Jeffrey D.; and Anderson, Steven N.; Simulated Car Crashes at Intersections in Drivers With Alzheimer Disease, Alzheimer Disease and Associated Disorders, Vol. 15, No. 1, pp. 10–20, 2001

39. Hamker, F.H., RBF learning in a non-stationary environment: the stability-plasticity dilemma, in: R.J. Howlett and L.C. Jain (eds), Radial Basis Function networks 1: Recent Developments in Theory and Applications; Studies in fuzziness and soft computing: vol. 66, Heidelberg: Physica Verlag, Ch. 9, 219-251, 2001
40. Fitts, P.M. and Posner, M.I., Human Performance, Belmont, CA: Brooks/Cole, ISBN 0-13-445247-X, (1967)
41. Anderson, J.R.: Learning and Memory, John Wiley, 2002
42. Anderson, J.R., How Can the Human Mind Occur in the Physical Universe ?, Oxford: Oxford University Press, 2007
43. Quin, Y., Bothell, D., & Anderson, J. R. (2007): ACT-R meets fMRI. In Proceedings of LNAI 4845 (pp. 205-222). Berlin, Germany: Springer.
44. Anderson, J.R., Fincham, J. M., Qin, Y., & Stocco, A. (2008). A Central circuit of the mind. Trends in Cognitive Science. 12(4), 136-143.
45. Salvucci, D.D., Modeling Driver Behavior in a Cognitive Architecture, Human Factors, 48(2), 362-380, 2006
46. Salvucci, D.D., Integrated Models of Driver Behavior, p. 356-367, in: W.D. Gray (ed), Integrated Models of Cognitive Systems, Oxford University Press, 2007
47. Bischof, N., Struktur und Bedeutung: Einführung i. d. Systemtheorie, Bern: Huber, 1995
48. Jagacinski, R.J. & Flach, J.M., Control Theory for Humans: Quantitative Approaches to Modeling performance, Mahwah, N.J.: Lawrence Erlbaum Associates, 2003
49. Wickens, Th.D., Models for Behavior: Stochastic Processes in Psychology, San Francisco: Freeman, 1982
50. Gopnik, A. & Tenenbaum, J.B., Bayesian networks, Bayesian learning and cognitive development, Development Science, 10:3, 2007, 281 – 287
51. Griffiths, Th.L., Kemp, Ch., and Tenenbaum, J.B., Bayesian Models of Cognition, p.59-100, in: R. Sun (ed), The Cambridge Handbook of Computational Psychology, Cambridge University Press, 2008
52. Chater, N., and Oxford, M., (eds): The Probabilistic Mind: Prospects for Bayesian Cognitive Science, Oxford, England: Oxford University Press (2008)
53. Pearl, J., 1988, Probabilistic Reasoning in Intelligent Systems, San Mateo, CA: Morgan Kaufmann, 1988
54. Pearl, J., 2009 (2nd ed.), Causality: Models, Reasoning and Interference, Cambridge, UK: Cambridge University Press
55. Spirtes, P., Glymour, C., and Scheines, R., 2001 (2nd ed.), Causation, Prediction, and Search, Cambridge, Mass: MIT Press
56. Jensen, F.V. & Nielsen, Th.D.: Bayesian Networks and Decision Graphs (2nd edition), Springer, ISBN 0-387-68281-3 (2007)
57. Koller, D., Friedman, N. (2009). Probabilistic Graphical Models, Cambridge, Mass.: MIT Press, ISBN 978-0-262-01319-2
58. Meila, M., Jordan, M.I.: Learning Fine Motion by Markov Mixtures of Experts, MIT, AI Memo No. 1567, 1995
59. Bishop, C. M. and Svensén, M. (2003): Bayesian hierarchical mixtures of experts. In: Kjaerulff, U. and C. Meek (Ed.): Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 57-64.
60. Russell, St. and Norvig, P., Artificial Intelligence: A Modern Approach, Upper Saddle River, N.J.: Prentice Hall, 2010 (3rd ed.)
61. Oliver, N. & Pentland, A.P., Graphical Models for Driver Behavior Recognition in a SmartCar, IEEE Intl. Conf. Intelligent Vehicles, 7 – 12, 2000
62. Miyazaki, T.; Kodama, T.; Furushahi, T.; & Ohno, H.; Modeling of Human Behaviors in Real Driving Situations, 2001 IEEE Intelligent Transportation Systems Conference Proceedings, 2001, 643 - 645

63. Kumugai, T.; Sakaguchi, Y.; Okuwa, M.; & Akamatsu, M.; Prediction of Driving Behavior through Probabilistic Inference, Proceedings of the 8th International Conference on Engineering Applications of Neural Networks (EANN ’03), 117 – 123
64. Bengio, Y., Frasconi, P.: Input/output Hidden Markov Models for Sequence Processing, IEEE Transactions on Neural Networks, 1996, 7, 1231—1249
65. Koike, C.C.; Bessiere, P.; and Mazer, E.; Bayesian Approach to Action Selection and Attention Focusing, in P. Bessiere et al., (Eds.), Probabilistic Reasoning and Decision Making in Sensory-Motor Systems, Berlin: Springer, 2008, 177 – 201
66. Oliver, N.M., Towards Perceptual Intelligence: Statistical Modeling of Human Individual and Interactive Behaviors, MIT Ph.D. Thesis, 2000
67. Lawrence R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, Band 77, Nr. 2, S. 257–286, 1989
68. Jurafsky, D. and Martin, J.H., Speech and Language Processing, Pearson, 2009 (2nd ed)
69. Kumugai, T. & Akamatsu, M.; Prediction of Human Driving Behavior Using Dynamic Bayesian Networks, IEICE-Transactions on Info and Systems, Volume E89-D, Number 2, February 2006, Pp. 857-860
70. http://www.ipg.de/32.html (visited 28th February, 2010)
71. Horrey, W.J., Wickens, Ch.D., and Consalus, K.P.: Modeling Driver’s Visual Attention Allocation While Interacting With In-Vehicle Technologies, J.Exp. Psych.,2006,12, 67-78
72. Bishop, Ch. M.: Pattern Recognition and Machine Learning, Heidelberg: Springer, 2006
73. Lee, D. N.: How movement is guided (2006) http://www.perception-in-action.ed.ac.uk/publications.htm (visited 28th February, 2010)
74. Lee, D.N.: A theory of visual control of braking based on information about time-to-collision. Perception, 1976 , 5, 437-459.
75. http://www.norsys.com/ (visited 28th February, 2010)

9 Appendix A1: DAGs of Static and Dynamic Bayesian Models

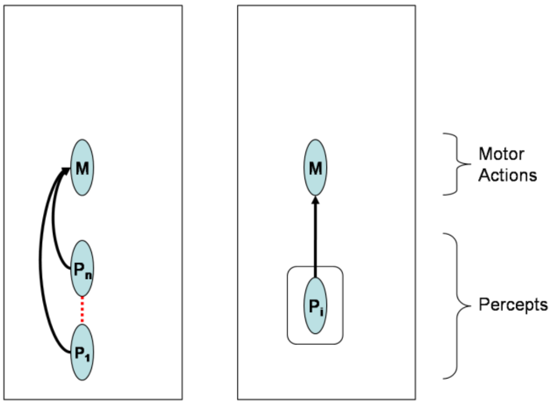


Fig. A1.01: Reactive Bayesian Network (BN) [21, 31]; ellipses in plates denote sets of random variables (plate notation [72])

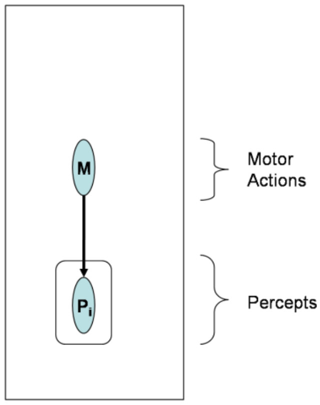


Fig. A1.02: Inverse (naïve) Classifier BN [32, 60]

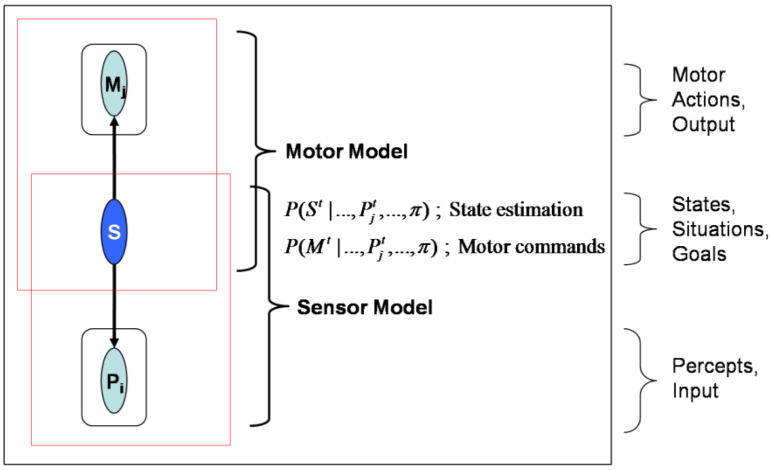


Fig. A1.03: Inverse BN-Model with State Variable [65]

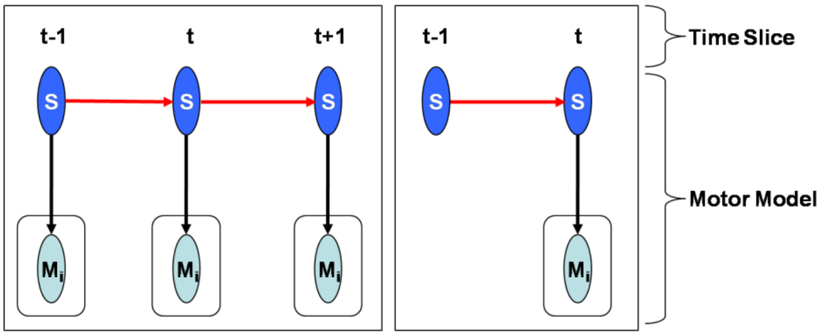


Fig. A1.04: Hidden Markov Model (HMM) [62, 63, 67]

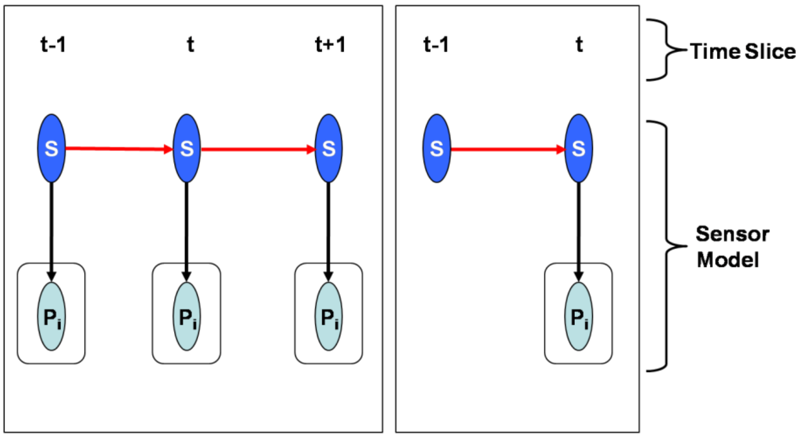


Fig. A1.05: Hidden Markov Model (HMM) with (Inverted) Sensor Model [32]

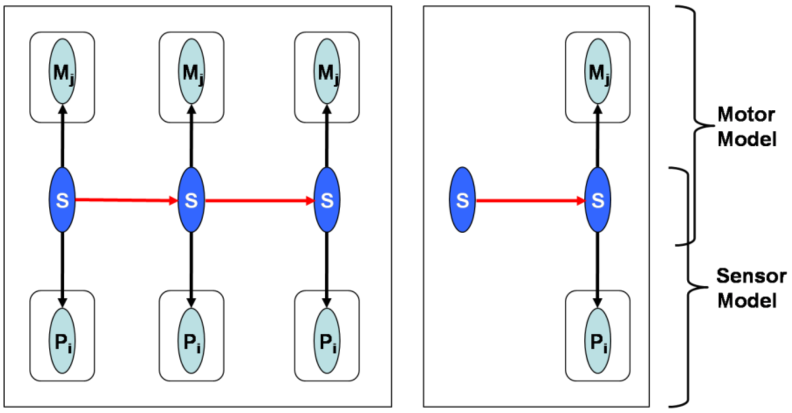


Fig. A1.06: Hidden Markov Model (HMM) with Motor and (Inverted) Sensor Model [61, 62, 65]

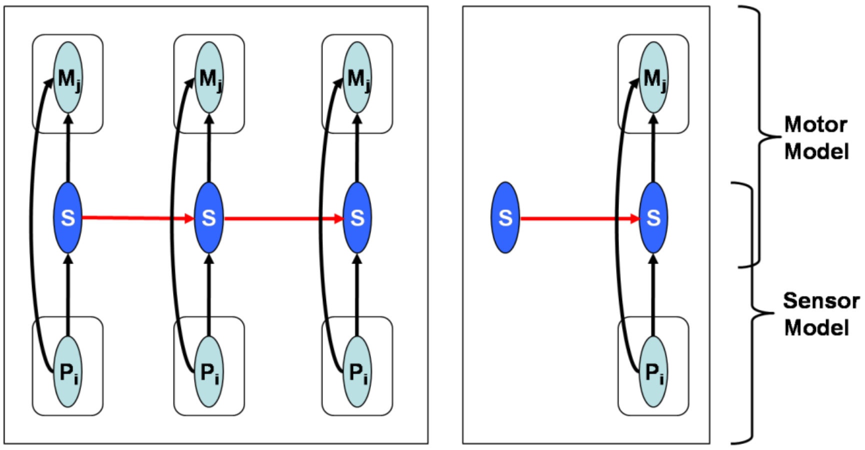


Fig. A1.07: (Reactive) Input-Output HMM (RIOHMM) – slight modification of [64]

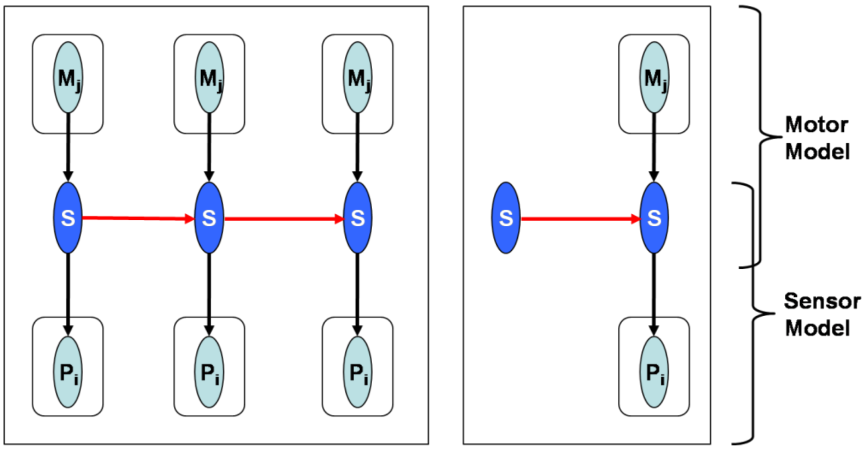


Fig. A1.08: Discrete Bayesian Filter (= HMM with Sensor and Inverted Motor Model) [29, 65]

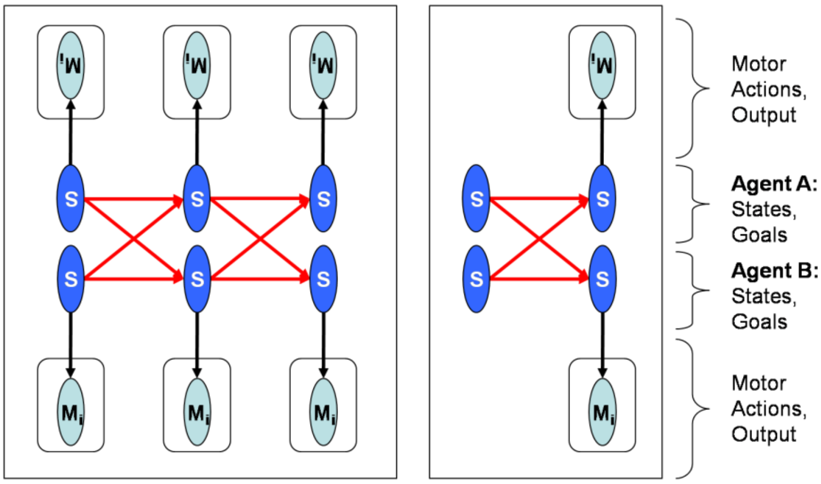


Fig. A1.09: Coupled HMM (CHMM) [66]

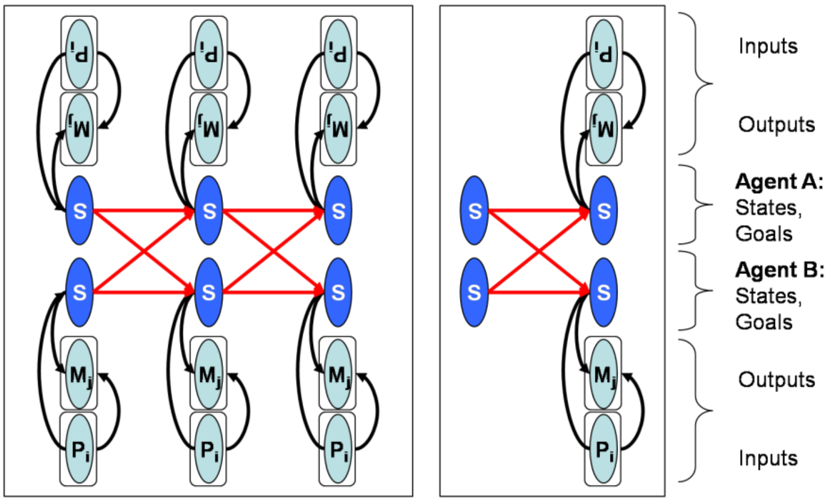


Fig. A1.10: Coupled Reactive HMM (CRHMM)

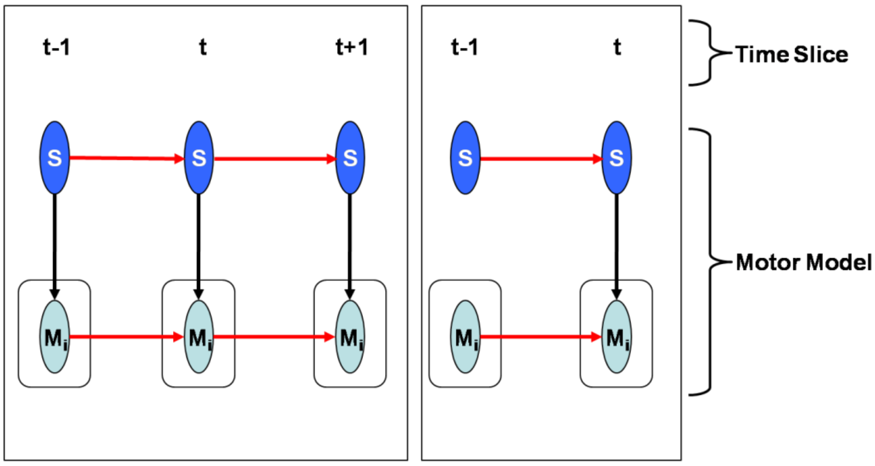


Fig. A1.11: Switching Linear Dynamic System (SLDS) [63]

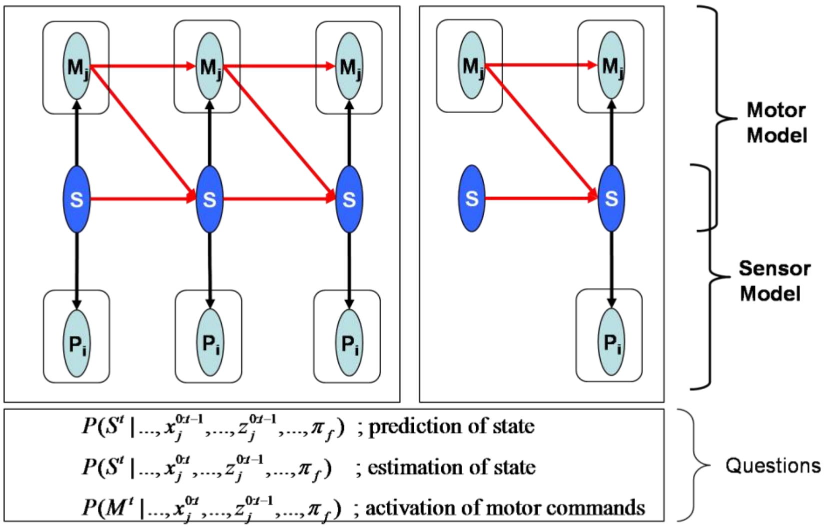


Fig. A1.12: Bayesian Filter and Action Model [65, p.180]

10 Appendix A2: Netica Implementations of Paradigmatic Dynamic Bayesian Agent Models

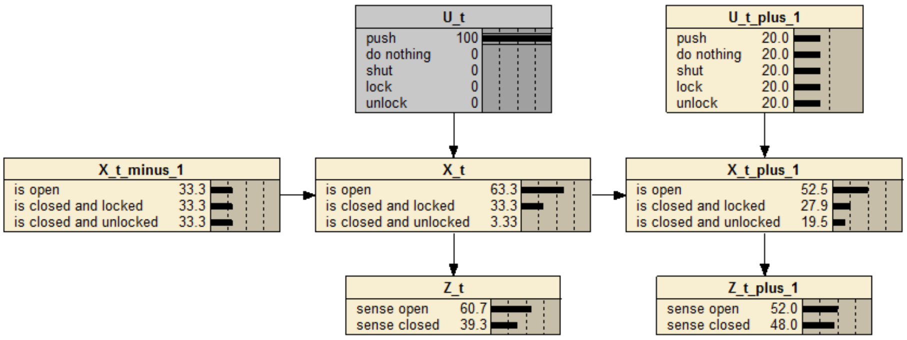


Fig. A2.01: *Prediction* Step in Night Watchman DBF

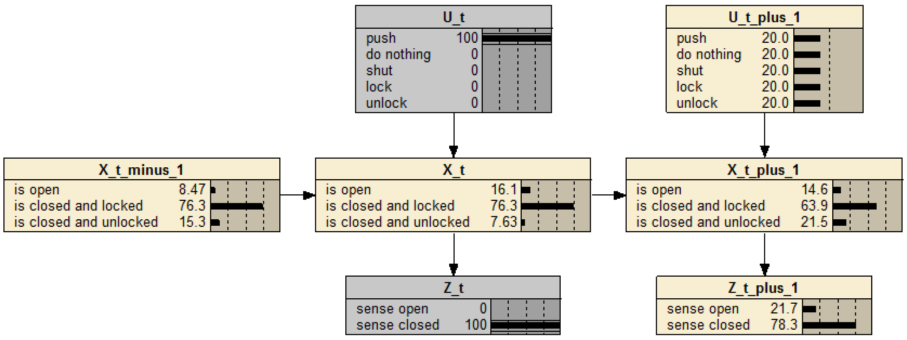


Fig. A2.02: *Correction Step* in Night Watchman DBF

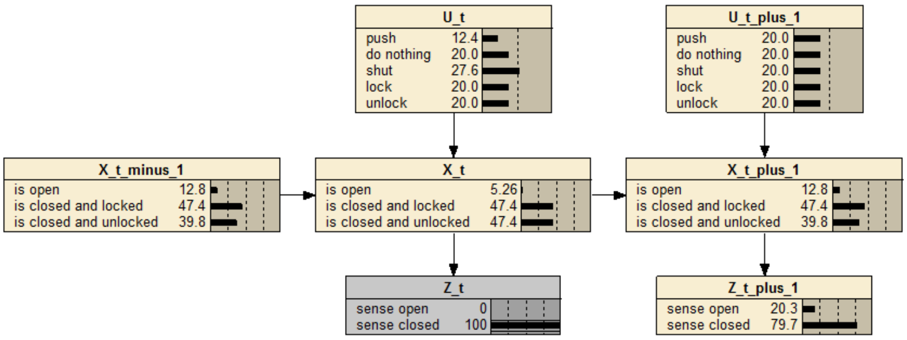


Fig. A2.03: *Perception Step* in Night Watchman DBF

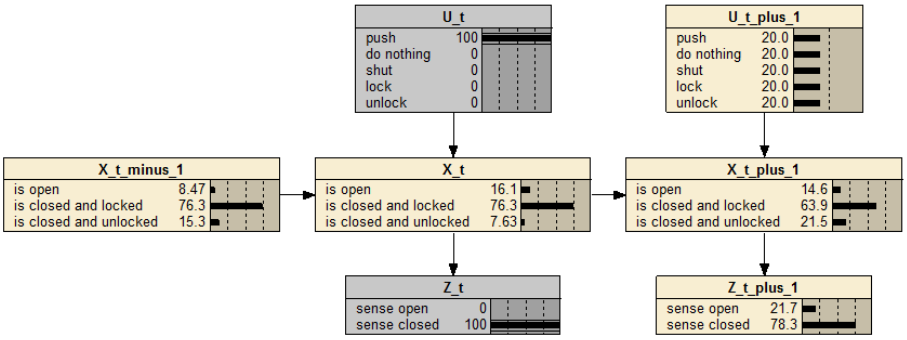


Fig. A2.04: *Action Step* in Night Watchman DBF

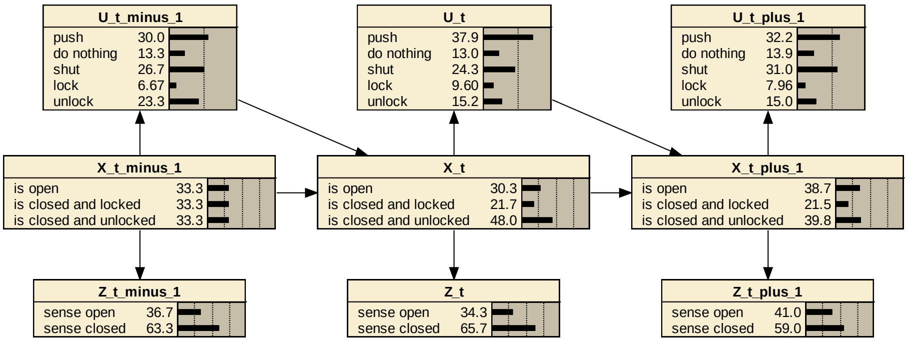


Fig. A2.05: *Apriori Beliefs* in Expert-Role, Mixed Experts, or Schema DBN Model with Action Effects

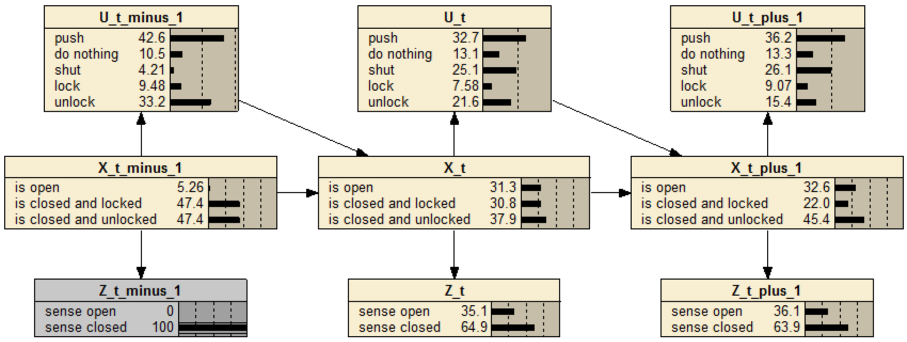


Fig. A2.06: *First Perception Step* in Night Watchman DBN with Action Effect Model

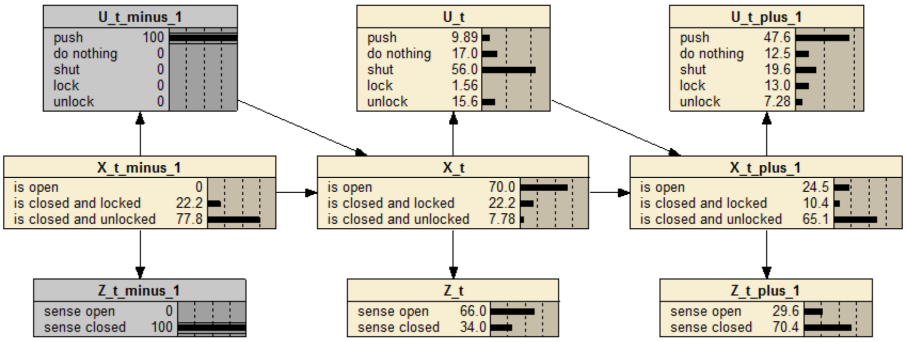


Fig. A2.07: *First Action Step* in Night Watchman DBN with Action Effect Model

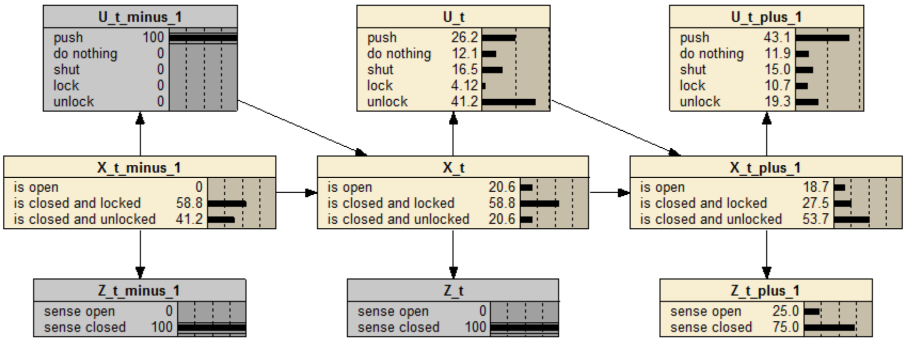


Fig. A2.08: *Second Perception Step* in Night Watchman DBN with Action Effect Model

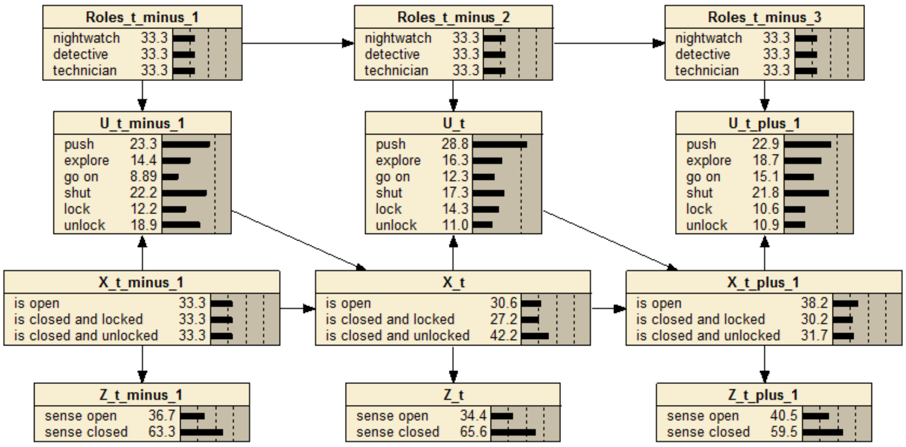


Fig. A2.09: *Apriori Beliefs* in Expert-Role, Mixed Experts, or Schema DBN Model with Action Effect Model

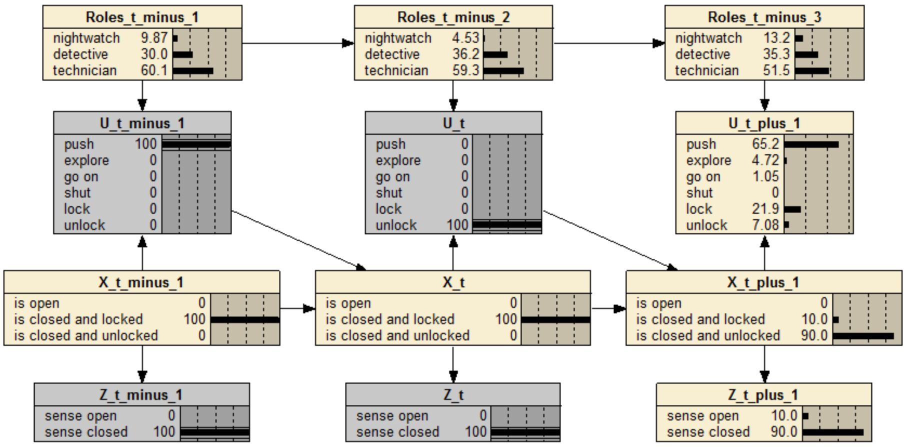


Fig. A2.10: *Role, Schema, or Intention Diagnostic* in Expert-Role, Mixed Experts, or Schema DBN Model with Action Effect Model

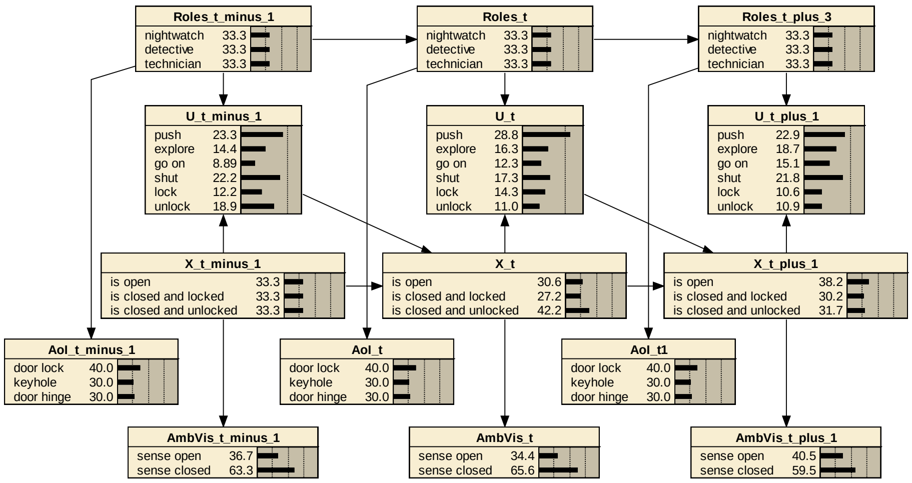


Fig. A2.11: *Apriori Beliefs* in AoI and Ambient Vision-Role-Model

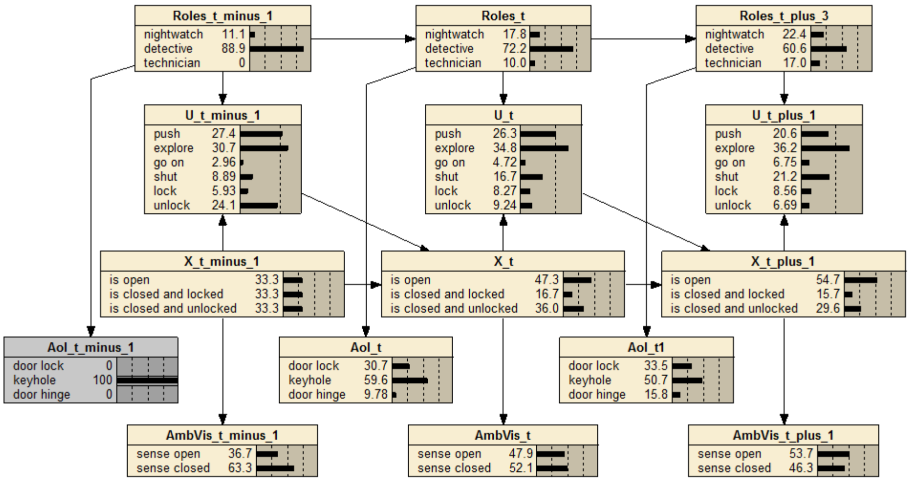


Fig. A2.12: *Inference of Intention, Role, and Action* in AoI and Ambient Vision-Role-Model

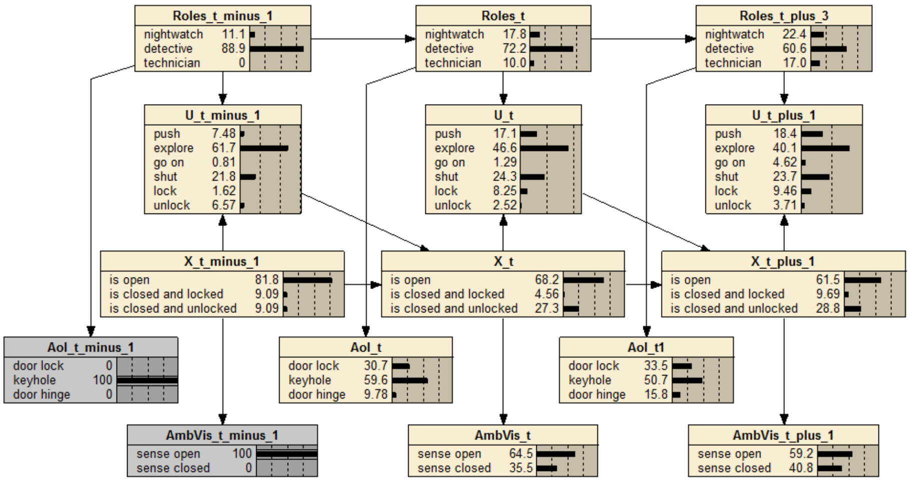


Fig. A2.13: *Inference of Intention, Role, and Action* in AoI and Ambient Vision-Role-Model

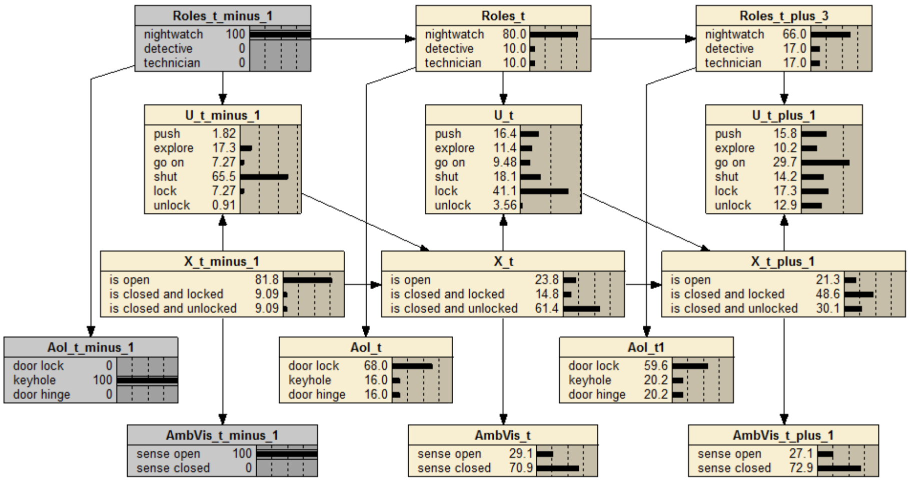


Fig. A2.14: *Inference of Role-specific Actions* in AoI and Ambient Vision-Role-Model

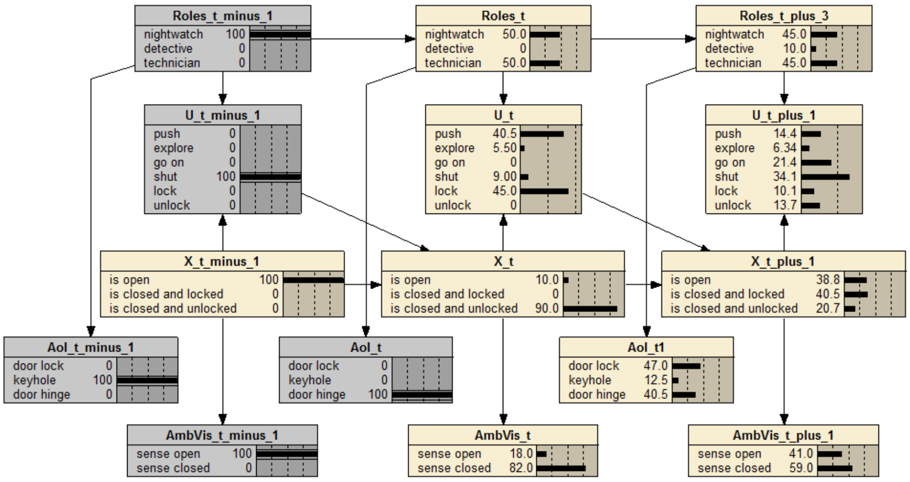


Fig. A2.15: *Role or Intention Conflict* in AoI and Ambient Vision-Role-Model

11 Glossary, key terms, definitions

anomalies

risky maneuvers are called anomalies when they have a low probability of occurrence in the behavior stream of experienced drivers and which only experienced drivers are able to prevent or to anticipate automatically. A measure of the anomaly of the driver's behavior is the conditional probability of his behavior under the hypothesis that the observed actions are generated by a stochastic process which generated the trajectories or behaviors of the correct maneuver M+.

anticipatory planning

For anticipatory planning the conditional probability of the NextFutureDrive under the assumption of the pastDrive, the currentDrive, and the anticipated expectedFutureDrive has to be computed.

Bayesian Assistance Systems (BAS)

For the purpose of smart assistance in simulated or real world scenarios the obtained Bayesian Autonomous Driver (BAD) models can be used as prototypical Bayesian Assistance Systems (BAS). Due to their probabilistic nature BAD models or BAS can not only be used for *real-time control* but also for *real-time detection of anomalies* in driver behavior and *real-time generation* of supportive *interventions (countermeasures)*.

Bayesian Autonomous Driver (BAD) model

BAD models describe phenomena on the basis of the variables of interest and the decomposition of their joint probability distribution (JPD) into conditional probability distributions (CPD-factors) according to the *special chain rule for Bayesian networks*. The underlying conditional independence hypotheses (CIHs) between sets of variables can be tested by standard statistical methods (e.g. the conditional mutual information index. The parameters of BAD models can be learnt objectively with statistical sound methods by batch from multivariate behavior traces or by learning from single cases.

Due to their probabilistic nature BAD models or BAS can not only be used for *real-time control* of vehicles but also for *real-time detection of anomalies* in driver behavior and *real-time generation* of supportive *interventions (countermeasures)*.

Bayesian Autonomous Driver with Mixture-of-Behaviors (BAD-MoB) model

The model is suited to represent the sensor-motor system of individuals or groups of human or artificial agents in the functional *autonomous* layer or stage of Anderson. In a MoB model it is assumed that the behavior can be context-dependent generated as a mixture of ideal schematic *behaviors* (= experts). The template or class model is distributed across two time slices, and tries to avoid the *latent* state assumptions of Hidden Markow Models. Learning data are time series or case data of relevant variables: percepts, goals, and actions. Goals are the only latent variables which could be set by commands issued by the higher *associative* layer.

Bayesian Filter and Action Model (BFAM)

In the Bayesian Filter and Action Model actions are not only dependent on the current process state but also on *direct* antecedent actions. Thus the generation of erratic behavior is suppressed. Furthermore the BFAM includes *direct* action effects on the next future process state. This is important when the influence of action effects should be modeled *directly* into the state not making a detour via the environment and the perception of the agent.

Bayesian learning of agent models under human control

The performance of the BAD model is observed by the human driver while the BAD model is driving. New data are learned only when the model behavior is unsatisfying. By *observing and correcting* the actions of the BAD model *only when needed*, problems can be solved, which are nearly impossible to discover by just analyzing its probability distributions.

Bayesian (Robot) Programs (BPs)

BP is a simple and generic framework suitable for the description of human sensory-motor models in the presence of incompleteness and uncertainty. It provides integrated *model-driven data analysis* and *model construction*. In contrast to conventional Bayesian network models BP-models put emphasis on a *recursive structure* and infer concrete motor actions for *real-time control* on the basis of sensory evidence. Actions are sampled from CPDs according various strategies after propagating sensor or task goal evidence.

Computational agent model

Computational agent models have to represent perceptions, beliefs, goals, and actions of ego and alter agents.

cooperative scenario

when goals are issued by one single principal.

cooperative driving scenario

is driving scenario with in-vehicle-cooperation between a human driver and a BAS

distributed cognition

was originated by Edwin Hutchins in the mid 1980s. He proposed that human knowledge and cognition is not confined to individuals but is also embedded in the objects and tools of the environment. Cognitive processes may be distributed across the members of a social group or the material or environmental structure.

Dynamic Bayesian Filter (DBF)

The DBF is a HMM with state, percept and motor variables. The general algorithm consists of two steps in each iteration or recursive call:

3. Prediction step: from the most recent *apriori* belief(state) and the current control (= action) compute a provisional belief(state)
4. Correction step: from the current provisional belief(state) and the current measurements (= percepts) compute *the posteriori* belief(state).

Dynamic Bayesian network (DBNs)

In the case of identical time-slices and *several* identical temporal links we have a *repetitive temporal model* which is called *Dynamic Bayesian Network model* (DBN). DBNs are dynamic probabilistic models. HMMs and DBN are mathematically equivalent. Though, there is a trade-off between estimation efficiency and descriptive expressiveness in HMMs and DBNs. Estimation in HMMs is more efficient than in DBNs due to algorithms (Viterbi, Baum-Welch) whereas descriptive flexibility is greater in DBNs. At the same time the state-space grows more rapidly in HMMs than in corresponding DBNs.

Dynamic probabilistic model

Dynamic probabilistic models evolve over time. If the model contains discrete time-stamps one can have a model for each unit of time. These local models are called *time-slices*. The time slices are connected through *temporal links* to give a full model.

Hidden Markow Models (HMMs)

A special category of time-stamped dynamic probabilistic models is that of a *Hidden* Markov *Model* (HMM). They are repetitive temporal models in which the state of the process is described by a *single discrete* random variable. Because of the Markov assumption only temporarily adjacent time slices are linked by a *single* link between the state nodes.

HMMs are sequence classifiers and allow the efficient *recognition* of situations, goals and intentions; e.g. diagnosing driver's intention to stop at a crossroad. HMMs and DBN are mathematically equivalent. Though, there is a trade-off between estimation efficiency and descriptive expressiveness in HMMs and DBNs. Estimation in HMMs is more efficient than in DBNs due to algorithms (Viterbi, Baum-Welch) whereas descriptive flexibility is greater in DBNs. At the same time the state-space grows more rapidly in HMMs than in corresponding DBNs.

partial or non-cooperative scenario

when goals are issued by several different principals.

shared space

approach is based on the observation that individuals' behavior in traffic is more positively affected by the built environment of the public space than by conventional traffic control devices (signals, signs, road markings, etc.) or regulations.