The International Journal of
Robotics Research
30(5) 601–614

DOI: 10.1177/0278364910393040
ijr.sagepub.com

System interdependence analysis for autonomous robots

Georgios Lidoris[1], Florian Rohrmüller[1], Dirk Wollherr[1,2], and Martin Buss[1]

Abstract

With the increasing complexity of robotic systems, system robustness and efficiency are harder to achieve, since they are determined by the interplay of all of a system's components. In order to improve the robustness of such systems, it is essential to identify the system components that are crucial for each task and the extent to which they are affected by other components and the environment. Such knowledge will help developers to improve their systems, and can also be directly utilized by the systems themselves, for example, to detect failures and thereby correctly adjust the system's behavior.

In this article a method of system interdependence analysis is presented. The basic idea is to learn and quantitatively evaluate the coherence between performance indicators of different system components, as well as the influence of environmental parameters on the system. To validate the proposed approach, system interdependence analysis is applied to the navigation system of an autonomous mobile robot. Its navigational methods are presented and suitable indicators are derived. The results of using the method, based on experimental data from an extended field experiment, are given.

Keywords

AI reasoning methods, autonomous agents, cognitive robotics, learning and adaptive systems

1. Introduction

Autonomous robotic systems are used to carry out tasks in partially known or unknown environments where they constantly encounter situations that require decision-making capabilities under perceptual uncertainty. This uncertainty can lead to undesired system behavior. However, the consequential series of internal reactions that cause the observed behavior is often unclear, since it results from the interaction of various system components. As a consequence, in order to ensure robustness and reliability of such autonomous robots, it is important to identify the crucial environmental and system component indicators that reflect the system's overall behavior. The identification of the mutual interdependencies allows conclusions to be drawn about the influence of these indicators on the system's behavior. Such knowledge is, for example, valuable for design choices, since the factors that contribute most to the variability of the system's behavior can be identified and it can be determined if these require additional research to strengthen system robustness. Additionally, this information can be used to enable autonomous systems to avoid failures by predicting the effects of actions and correctly adjusting their behavior.

Most of the current autonomous robots are complex systems designed for specific applications. They usually consist of components that can be separated into three categories according to their purpose. Perceptual components are responsible for building an environment representation, for example, in the form of a map, and also for localizing the robot. This representation is then used by planning components to calculate a plan of actions such as the trajectory of the robot. Finally, the chosen plan is executed and progress is monitored by task execution components. It is obvious that sensing, planning and execution are interconnected. Although performance indicators have been proposed for each of these domains, it is still hard to assess the effect that environmental parameters, or variations of the performance of specific system components, have on the rest of the system.

[1]Institute of Automatic Control Engineering (LSR), Faculty of Electrical Engineering, Technische Universität München, Germany
[2]Institute for Advanced Study (IAS), Technische Universität München, Germany

Corresponding author:
Georgios Lidoris, Institute of Automatic Control Engineering (LSR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany.
Email: georgios.lidoris@tum.de

This article describes a generic method to tackle this problem. A probabilistic model of the interdependencies between system components, such as perception, planning and execution, is learned, which leads to a mathematical model of the system that enables the determination of the crucial components with respect to robustness within a system. In this paper, performance is expressed as system stability against external and internal influences. In principle, such a model can also be derived by examining the deterministic interdependencies within the system. However, with increasing system complexity – more components and higher interconnectivity between them – the derivation of a deterministic model is hard. In this regard, a learned model facilitates design choices without the need to fully examine the deterministic interrelations in the system. It also provides a way of verifying existing deterministic system models and to identify relations that have not been modeled. Furthermore, the knowledge gained can be integrated into the online reasoning process of the system itself to enhance its autonomy. The presented analysis is applicable to any robotic system for which the components of interest are observable. For these, it identifies the best interdependence model that explains the data retrieved from the system. The presented analysis is illustrated by the application of the method to the navigation system of the Autonomous City Explorer (ACE) robot.

This article is organized as follows. Section 2 summarizes existing work on robotic performance evaluation. In Section 3, the system interdependence analysis method is presented. Section 4 gives a sample case study for the ACE robot, including the description of the ACE project, the navigational methods used and the identification of suitable indicators for interdependence analysis. In Section 5, results from system analysis applied to the ACE robot are presented.

2. Related work

The existence of literature focusing on the performance evaluation of autonomous systems confirms the importance of such methods. Qualitative evaluation criteria of robotic systems have been proposed in Crandall and Goodrich (2003). These approaches focus on task objective and social measures to identify the efficiencies of both robot and human. In Huang et al. (2004) an evaluation framework for characterizing the autonomy of unmanned vehicles by considering mission complexity, environmental difficulty and Human Robot Interaction (HRI) is presented. However, in order to apply these concepts to embodied autonomous robots and compare their performance with other existing systems and different environments, benchmarks and quantitative performance evaluation criteria are required.

Benchmark scenarios, such as the DARPA Grand Challenge (DARPA, 2007) and RoboCupSoccer (RoboCup, 2009), are used to compare the performance of autonomous systems. A similar way to provide reproducibility of environmental conditions is to standardize test arenas for mobile robots (Jacoff et al., 2002). However, such benchmarks cannot provide a comparison of robotic systems applied to different scenarios. For example, it is not possible to compare a robot that was built to operate in a home environment (Srinivasa et al., 2008) with autonomous vehicles, which are supposed to navigate through an urban environment (Urmson et al., 2008). Scenario-dependence is so strong that the winning vehicle of the first DARPA Grand Challenge (Thrun et al., 2006) would not have been able to take part in the second challenge, since the scenario changed from the desert to an urban environment. Furthermore, standardized benchmark scenarios result in an intensified development of robotic systems for these specific situations. A problem in this respect might be the adaptation of algorithms to these specific situations to allow for the cost of generality.

Further approaches introduce quantitative metrics to evaluate robot performance and the influence of the environment during navigation missions. Several metrics are proposed in Munoz et al. (2007) to characterize path quality. The entropy and compressibility of the environmental information are used in Anderson and Gang (2007) to estimate the complexity of an environment. This method can also be used to identify attractor points. The relation between the environment and the performance of a robotic system is learned in Held et al. (2006), using a Dynamic Bayesian Network. This way the coherence among the metrics and also the environment is identified. To what extent the performance of one system component influences the quality of another is determined in Lampe and Chatila (2006), where the degree of autonomy of a robot is evaluated by combining task performance with world complexity. However, the evaluation is based only on simulated data assuming complete knowledge of the environment.

The method presented in the next section uses a Bayesian Network (BN) for coherence identification and extends it to a formal method, which allows the quantitative determination of the interdependence among all pairs within a set of arbitrary indicators.

3. System interdependence analysis

In this section a method for system interdependence analysis is proposed. It allows the determination of the degree of interaction between specific system components and its influence on the system's overall performance.

The proposed approach is illustrated in Figure 1. As the robot operates, system outputs are monitored and performance indicators for the system components are calculated. The indicator values are used offline to learn the structure of the BN that best reflects the system data gathered, and to train its parameters. BNs are a network-based framework for representing and analyzing models involving uncertainty. They find application in several fields ranging from intelligent decision support aids, to data fusion, 3D-feature

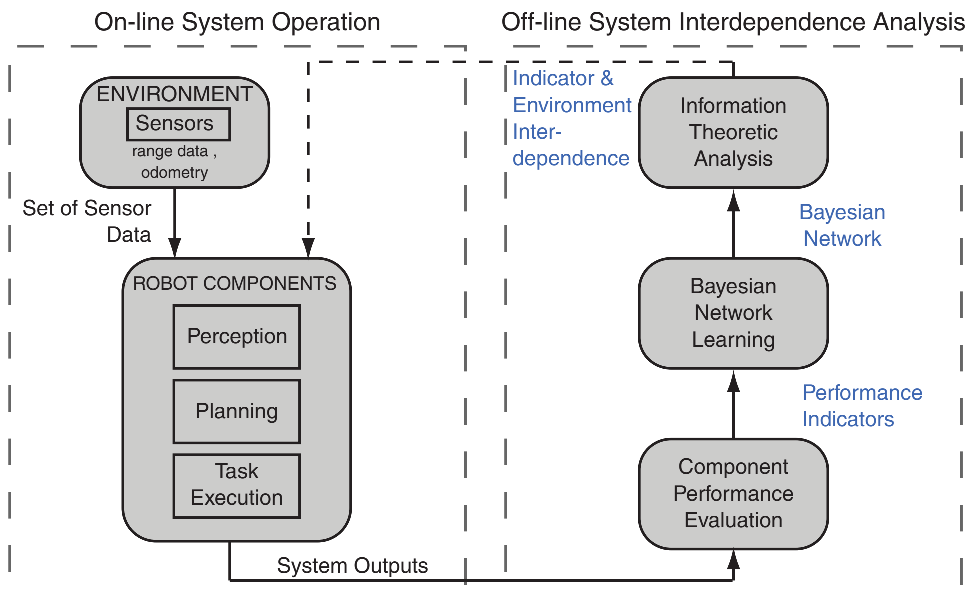

Fig. 1. Flow chart of the proposed system interdependence analysis.

recognition, intelligent diagnostic aids, automated free text understanding and data mining.

The learned BN structure identifies the coherence between the performance indicators computed from the system outputs, that is, the extent to which the indicators are associated with each other. In order to quantitatively evaluate this coherence between indicators from different system components, information-theoretic analysis is performed on the parameters of the learned BN. The calculated quantitative relation can then be used to adjust the online functionality of the robot to the situation. This is illustrated by the dashed line in Figure 1. Section 3.4 gives a brief description of how this can be achieved. However, a more detailed discussion is beyond the scope of this article but can be found in Rohrmüller et al. (2010).

In the following the determination of indicators, the calculation of relevant data, the structure learning and the information-theoretic analysis are described.

3.1. Component performance evaluation for autonomous robotic systems

Before a model can be learned a set of component indicators needs to be specified, which reflects the state of the components of the autonomous robot as well as its environment. Therefore, for each component of interest at least one representative indicator needs to be chosen. In principle, these indicators can be arbitrarily chosen by the system designer, for example, based on design knowledge or experience, thus there is no fixed set of universal indicators. Rather, the choice should be based on the properties of the system in question. In general, a good starting point is to select indicators that reflect the performance of the components. As already stated in Section 1, here performance is understood as system stability against external and internal influences.

In the literature, suitable performance measures have been established for various methods and problems. For example, planning algorithms are commonly measured with respect to solution quality or required time, such as in Ross et al. (2008), Van den Berg et al. (2006) and LaValle (2006). While perception algorithms typically use statistical criteria, for example, as in Christensen and Förstner (1997) and Roy et al. (1999).

Nevertheless, the indicator selection is not crucial for the performance of the analysis, since inappropriate indicators are identified because they have no or only weak interdependencies with the other indicators. This means that these indicators provide no information about the parts of the system represented by the other indicators, but, of course, they might be still relevant for the analysis of the relevant components.

When the set of indicators is defined, the system output data, which is required to compute the indicator values, needs to be gathered. This is simply done through various experimental runs under appropriate environmental conditions. During these runs, all system data of interest is recorded by sampling with a fixed rate. The latter is assumed to be the same for all data of interest and should be chosen with respect to the dynamics of the system and the environment. Thereafter the gathered data is processed offline and the indicator values, which provide the basis for the model learning, are derived.

Before the model can be learned, the indicator data needs to be discretized. The number of intervals used for the discretization of the indicator values should not be too low, in order to maintain the contained information. However, if there are too many intervals, the probability distributions are too flat, and this makes the determination of mutual interdependencies hard. This similarly applies to the rate of sampling, as it requires discretization in the temporal domain. A possible solution for selecting the size of the intervals is to use an entropy-based approach, such as in Clarke and Barton (2000).

3.2. Learning BN structures

As discussed previously, in order to find out whether and to what extent performance indicators of system components interact with each other, a BN is learned from system outputs and suitable performance indicator values. The topology of the network is unknown beforehand, but the system is fully observable by the data. In order to find the network structure that best models the data, a search through the space of possible structures is performed using a likelihood heuristic.

BNs are well-established tools for representing uncertain relations between several random variables (Russell and Norvig, 2002). They demonstrate several advantages over other knowledge representation and probabilistic analysis tools by providing a simple way to visualize the structure of a probabilistic model. This structure can be used to design and motivate new models. Also insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph. Uncertainty is handled in a mathematically rigorous yet efficient and simple way, by using Bayesian statistics. Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

A BN is an annotated Directed Acyclic Graph (DAG), which encodes a joint probability distribution over the set $X = \{X_1, \ldots, X_n\}$ of random variables. Formally, it is a tuple $B = \langle G, \Theta \rangle$, where G is a DAG whose vertices correspond to the random variables. A DAG implies conditional independence of each variable X_i and its non-descendants, given its set of parents $\mathrm{Pa}(X_i)$. Θ represents the set of parameters that define the transition between nodes. It contains a value $\theta_{i,j,k} = p(X_i = k_i | \mathrm{Pa}(X_i) = j_i)$ for each possible value k_i of X_i and each possible set of values j_i of $\mathrm{Pa}(X_i)$. The conditional probability distribution of each node is represented in a Conditional Probability Table (CPT).

In case there is no a priori transition information available, the space of possible DAGs is super-exponential in n, the number of variables described, and is given, according to Robinson (1977), by

$$g(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} g(n-k). \quad (1)$$

So the complexity of an exhaustive search is given by $O(cg(n))$, where c is a term expressing the operations required to evaluate a single structure. Figure 2 shows the

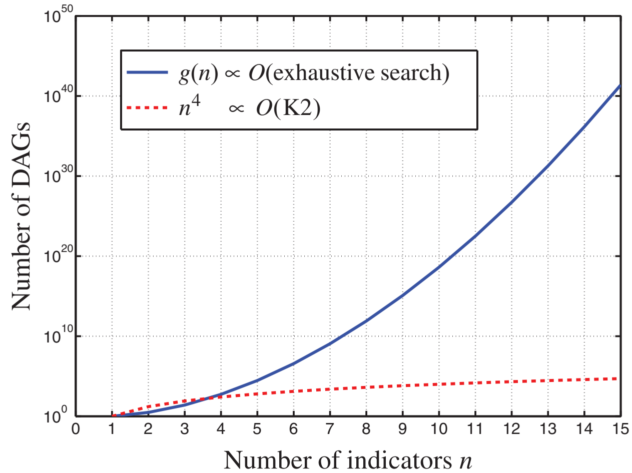

Fig. 2. Course of complexity with an increasing number n of indicators: for an exhaustive (e) and for the K2 (k) search. For simplicity $c = 1$.

rapid growth of $g(n)$ with increasing n, which illustrates that an exhaustive search is problematic for less than ten indicators. Therefore, to reduce the number of graph structures examined to a tractable number while still allowing for good solutions, various tools are used, for example, sampling-based methods, such as the Markov Chain Monte Carlo (MCMC) search (Murphy, 1999), are widely used. MCMC takes randomly sampled structures from the space of possible DAGs and evaluates them. The number of samples is chosen to be large enough that the search converges. The acceptance ratio is used as a convergence indicator for the search. This is the fraction of proposed samples with likelihood accepted by the approximation algorithm, divided by the number of samples that are rejected. Even though there is no guarantee that MCMC will find the optimal solution, its major benefits are controllable complexity and that it does not get stuck in local optima.

The Bayesian Information Criterion (BIC) (Schwarz, 1978), which is a function of the log likelihood of the structure according to the training data, penalized by the complexity of the structure, is used as a scoring function to evaluate the structures.

If a sequential ordering of the indicators is available, an alternative search algorithm is applicable: the well-established local greedy search algorithm K2 (Cooper and Herskovits, 1992). The ordering implies that nodes are dependent on preceding nodes and is used to initialize the K2 search. The search starts from an empty set of nodes. Parents are added incrementally – according to the ordering – and the node whose addition increases the score of the resulting structure the most, is kept. The maximum number of parents can be constrained by an upper bound. The algorithm stops adding parents to a node when it is no longer possible to increase the BIC score of the structure. In the worst case, without an upper bound, the complexity of K2 is $O(mn^4r) = O(cn^4)$, where m is the number of data samples and r the maximum number of values of any indicator.

K2 is beneficial in the sense of its comparably low complexity, as shown in Figure 2. The major drawback is the requirement for an initial node ordering, which is not available in general and also biases the solution. For a robotic system, the ordering could be retrieved from information about the sequential structure or causal relation of the indicators, for example, it could be deduced from the system design. Alternatively, the node ordering can be calculated from the solution of the MCMC search – by placing parents first and children subsequently – to further improve the result of the latter.

The presented search algorithms have been chosen due to their good tradeoff between complexity and solution quality. Furthermore, in combination they require no initial information. There are many other search methods in the literature and their possible benefits are still to be investigated.

From the search, the structure with the highest BIC score is used for further analysis. It provides a first qualitative view of the mutual interactions among the indicators.

In the following, information-theoretic criteria are used to evaluate the coherence between the indicators within the learned network.

3.3. Information-theoretic criteria

The BN structure itself does not provide a quantitative measure of indicator interdependence. In order to derive the latter, information-theoretic criteria (Cover and Thomas, 1991) are applied. Once the structure of the net is learned, the CPTs can be computed using the experimental data. For each pair of indicators X_i, X_j, the mutual information

$$I(X_i, X_j) = \sum_j p(j) \sum_{k_i} p(k_i|j) \log \frac{p(k_i|j)}{p(k_i)} \tag{2}$$

is derived. Intuitively, mutual information measures the information that X_i and X_j share, that is the extent that knowledge about one of them reduces the uncertainty about the other. For instance, if two variables are independent then knowledge about one of them does not give any information about the other. Consequently their mutual information is zero.

In order to make comparisons between different pairs of variables a distance metric is required. In this respect the joint entropy

$$H(X_i, X_j) = H(X_i|X_j) + H(X_j) \tag{3}$$

is calculated, where $H(X) = -\sum_{k \in X} p(x) \log p(x)$ is the entropy of the random variable X. The joint entropy measures the overall uncertainty of the two variables. The final distance metric is then derived using

$$0 \leq \eta(X_i, X_j) = \frac{I(X_i, X_j)}{H(X_i, X_j)} \leq 1, \tag{4}$$

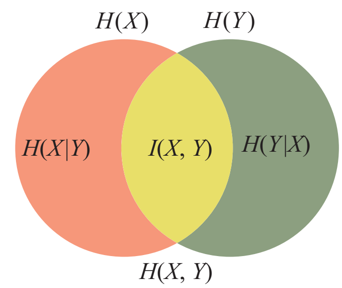

Fig. 3. The relation between mutual information $I(X, Y)$ and joint entropy $H(X, Y)$.

which corresponds to the ratio of mutual information and joint entropy. The relation of $I(X, Y)$ and $H(X, Y)$ is illustrated in Figure 3. It can be proven (Cover and Thomas, 1991) that η satisfies all properties of a metric such as the triangle inequality, non-negativity and symmetry. If two variables are independent then $\eta(X_i, X_j) = 0$, whereas when the variables are fully dependent and knowledge about the one completely reduces the uncertainty about the other $\eta(X_i, X_j) = 1$.

By computing η, the interdependence, within a set of indicators, between any pair is determinable, no matter whether there exists a direct connection in the BN or not.

3.4. Online applicability

The essential motivation of the presented method is to identify the system components that are crucial for the robustness and efficiency of a robot. The information gained can be utilized during system operation. This can be either achieved via a system redesign by the developer or by explicit incorporation into the online decision-making.

In this respect, the method enables also a reverse interpretation, that is, observations about the robot's internal state can be used to make predictions and estimations about the current environmental situation. This way, the robot is able to assess the situation and the effect that changes in performance indicators of specific system components have on the rest of the system. Such information can be traced back into the online system operation and used for behavior selection, any kind of online learning techniques, and the parameterization of adaptive controllers.

In Rohrmüller et al. (2010) the analysis is applied to a set composed of scenario-specific costs, task-specific costs and task-specific parameters. The identified metric interdependencies are utilized to estimate the scenario-specific cost of a task, which is required by planning components to reason about which action to perform next. The interdependencies are further used by execution components to adapt task-specific parameters in order to react to changes in a dynamic environment.

Since the scope of this article is system interdependence analysis, the online applicability of the information gained is not discussed any further. Instead, the method is

a b

Fig. 4. Two scenes from the field experiment in the *Deutsches Museum*.

demonstrated by using it for the ACE robot, for which a performance indicator set is determined in the next section.

4. Case study for the ACE robot

We demonstrate system interdependence analysis using the ACE mobile robot. In the following, the robot's navigation system is described and suitable indicators are determined. Hence, a system structure typical for most autonomous robots is used. System components are classified as perception, planning or execution components. Before deriving the appropriate indicators for the ACE robot an overview of the system is provided and the important system components are analyzed.

4.1. Overview of the ACE project

The ACE project aims to combine autonomous navigation with human–robot interaction. A robot has been developed that is capable of navigating in an unknown urban environment, based only on information extracted through interaction with passers-by and its local perception capabilities. A detailed description of the system is given in Bauer et al. (2009).

In 2008 two major field experiments were conducted with the robot. In an outdoor experiment the robot managed to find its way from the Technical University of Munich to the central square of Munich, without any prior map knowledge or GPS information. In the second experiment the robot was part of a museum exhibition, as shown in Figure 4. The demonstration was successfully performed without any algorithmic modifications to the system, which indicates the suitability of the navigation subsystem for both indoor and outdoor settings.

During these experiments the robot was faced with various difficult situations, where it had to navigate through crowded and narrow places. These environmental conditions had an observable influence on the robotic motion, such as more frequent turns or stops, for example. By applying the method of Section 3 their influence on the navigation system of the robot can be identified.

The navigational methods used and suitable indicators are presented next.

4.2. Indicators for perception module performance

In order to navigate safely to a defined goal, the ACE robot must be capable of localizing itself, generating a representation of the environment and finding a drivable path through it. This section describes the approaches used for Simultaneous Localization and Mapping (SLAM).

For the SLAM problem, the ACE project uses a grid-based approach with particle filters. Particle filters allow the approximation of arbitrary probability distributions, making them more robust to unpredicted events that cannot be modeled, such as small collisions, which often occur, especially in outdoor environments. Furthermore, grid-based SLAM does not rely on predefined feature extractors, which are dependent on the assumption that the environment exhibits a known structure. Therefore, in cluttered outdoor environments the grid-based approach provides a more robust and accurate mapping. More details of the SLAM implementation that was deployed on the robot can be found in Lidoris et al. (2009).

The most likely occupancy grid map m^b of the environment is calculated by the SLAM module. In order to integrate traversability information, such as the presence of curbs, the grid m^b is fused with the grid m^n retrieved from the traversability assessment (Bauer et al., 2009), to obtain the combined 2.5D grid m^c. The resulting 2.5D grid m^c is sent to the path-planning module, which uses it to assess obstacles from the SLAM module and non-traversable regions detected by the inclined laser range finder for path planning.

Perceptual indicators describe the uncertainty of the position and environment model of the robot. For a mobile robot, such a model is commonly represented by a map. Map uncertainty can be measured by the entropy H^m of the map. For an occupancy grid m this is given, as in Stachniss et al. (2005), by

$$H^m = -r^2 \sum_{l \in m} -p(l) \log p(l) + (1 - p(l)) \log p(1 - p(l)), \tag{5}$$

where l is a cell, $p(l)$ the occupancy probability of l and r the resolution of m.

Pose uncertainty

$$H^P = H(p(X_t|Z_t, U_t)) \approx \frac{1}{t} \sum_{j=1}^{t} H(p(x_t|z_t, u_t)), \tag{6}$$

is given as an average over the uncertainty of the different poses along the path as proposed in Roy et al. (1999).

Finally, map information $\mathrm{INFO}(m_{t-1} \| m_t)$ has been proposed in Held et al. (2006) as a measure of the local complexity of a map. It is defined as the relative entropy of m_{t-1} with respect to m_t, where m_t is the local map at time t. The local map m_t is extracted from the occupancy grid m, by taking an area $10\,\mathrm{m} \times 10\,\mathrm{m}$ around the robot. m_{t-1} is the corresponding spatial part of the map at the previous time step. The relative entropy

$$D_l(p_{t-1}(l) \| p_t(l)) = p_{t-1}(l) \times \log \frac{p_{t-1}(l)}{p_t(l)}, \tag{7}$$

for cell l is also known as Kullback–Leibler divergence. By taking the sum of the symmetric form

$$\mathrm{info}_l(m_{t-1} \| m_t) = \frac{D_l(p_{t-1}(l) \| p_t(l)) + D_l(p_t(l) \| p_{t-1}(l))}{2}, \tag{8}$$

the relative quantity of information around the robot

$$\mathrm{INFO}(m_{t-1} \| m_t) = \frac{1}{N} \sum_{l \in m_t} \mathrm{info}_l(m_{t-1} \| m_t) \tag{9}$$

is derived similar to Held et al. (2006), where N is a normalization factor.

Equations (5), (6) and (9) can be used to calculate three indicators for the perception modules of ACE.

The next part describes the method and indicators for the planning module.

4.3. *Indicators for planning module performance*

The planning components of a robotic system are responsible for reasoning about the appropriate actions to be taken next. For a mobile robot, a path planning module is needed, which generates safe paths to a specified goal. The path planner of the ACE robot is described in detail in Lidoris et al. (2009).

In order to assess the quality of a path planning module, its generated paths are examined with regard to several quantitative indicators. Below a set of indicators is proposed, which are applicable to most path planning approaches.

Probably the most intuitive indicator is the path length s_p, since it is usually minimized. Indicators that characterize the complexity of the planned path are the number n_w of waypoints w in the path, relative to the Euclidean distance to the goal, the variance $\mathrm{var}(\angle(w^1, \phi_r))$ of the angular deviation

$$\angle(w^1, \phi_r) = \left| \arctan(w^1_y, w^1_x) - \phi_r \right| \quad \in [0, \pi] \tag{10}$$

between the next waypoint w^1 and the robot's orientation ϕ_r, and the cumulative sum of the angular deviation

$$\mathrm{cad} = \sum_{i=1}^{n_w} \angle(w^i, w^{i-1}) \tag{11}$$

between consecutive w in the path, where $\arctan(w^0) = \phi_r$. Finally, the number of waypoints n_v that satisfy a maximum clearance constraint are considered. This can be calculated by using, for example, distance transformation algorithms (Cuisenaire, 1999).

These metrics can be applied to any global planner that generates paths consisting of a sequence of waypoints. The planning approach used by ACE performs an A* search on a hybrid graph composed of nodes extracted from a bounding box structure and a Voronoi graph. Since Voronoi graphs belong to the family of distance transformation algorithms, the corresponding waypoints satisfy the maximum clearance constraint.

Next the execution module is considered, which is responsible for a safe drive along the computed path.

4.4. *Indicators for execution module performance*

Once the next action of the robot has been chosen by the planning components, it needs to be transformed to motion commands that must be carried out by the robot actuators in a coordinated manner. This is the responsibility of the task execution components. For example, the planner of a mobile robot will choose an intermediate target in the form of a waypoint. The execution components of the robot are responsible for generating controls for the wheel motors such that the target is safely reached.

The execution components of the ACE robot obtain global waypoints from the planning module, described in the previous section. These are given as input to the

Table 1. Overview of proposed performance indicators.

Category	Indicators
Perception	H^m = map uncertainty (5) H^P = pose uncertainty (6) INFO($m_{t-1} \| m_t$) = rel. quantity of information (9)
Planning	s_p = path length n_w = #of waypoints var($\angle(w^1, \phi_r)$) = variance of angular deviation cad = cumulative sum of angular deviation (11) n_v = #of maximum clearance waypoints
Execution	v_r = robot's velocity var(ϕ_r) = variance of robot's orientation

obstacle avoidance module, which generates motor commands for the mobile platform. This module takes into account dynamic obstacles in the vicinity of the robot and ensures safe local navigation. A method similar to Philippsen and Siegwart (2003) is used to generate smooth and safe robot trajectories.

The execution efficiency of a performed navigation task can be evaluated by observing the execution time and the smoothness of the path, or, more specifically, the robot's speed v_r and the variance of the robot's orientation var(ϕ_r).

4.5. Performance indicators at a glance

All aforementioned indicators are summarized in Table 1. The indicators are grouped into the three categories according to the system component they characterize. Even though the indicators have been chosen to represent the internal system state of the navigational components of the ACE robot, some of them have been described in the literature and they all are directly applicable to any mobile robot that performs SLAM and plans the path that it drives along.

The performance indicators discussed in this section are now evaluated using experimental data gathered by the ACE robot.

5. Experimental results

In order to validate the proposed method, system interdependence analysis has been performed using the ACE robot, based on data gathered during the outdoor experiment that was described in Section 4.1. Next the behavior of the robot in two sample situations is described, followed by the results of system interdependence analysis.

5.1. Behavior of the ACE robot in two sample situations

The overall route of the robot during the experiment is illustrated in Figure 5.

Data chunks from two representative situations were used for system interdependence analysis. Figure 6 shows two typical scenes. The first situation shows navigation on a sidewalk in a less populated area. The occupancy grid map calculated by the robot is shown in Figure 5(a). The second situation is a typical example of navigation in a densely populated pedestrian zone. The calculated occupancy grid map is given in Figure 5(b). Both occupancy grids have been overlaid with satellite images taken from Google Earth to illustrate the accuracy of the maps, which have a resolution of 15 cm.

Figure 7 shows the output of the planning module for two sample scenes encountered during the experiment (left) and the corresponding outputs of the path planner (right). In scene (a), the robot is located in a narrow sidewalk passage and in scene (b) it is in a highly populated street, where the direction of travel may be blocked by people.

The right-hand side of Figure 7 shows details of the occupancy grid, which is transformed to C-Space and used for path planning. The transformation is indicated by the dark blue regions around obstacles (black). Replanning was performed at 2 Hz.

In Figure 7(a2), the Voronoi graph (blue line), which traverses the free space (white), and resulting path (three parallel red lines) can be seen. The path shown illustrates the advantage of the dual path planning approach. The first two waypoints (red dots) correspond to nodes retrieved from the Voronoi graph. All other waypoints correspond to nodes from the visibility graph, which is not shown here for clarity. While the path would proceed straight to the corner of the obstacle in the bottom (below the second waypoint) if only the bounding boxes were used, the extension with the Voronoi graph keeps the robot in the center of the narrow sidewalk leading to maximal clearance from obstacles. However, the passage in the area of the third and fourth waypoint is closer than the preferred obstacle distance used for the Voronoi graph. To drive through this area, the ACE robot needs to get as close to the obstacles as possible. This is enabled by using the nodes from the bounding boxes. The solid red center line indicates the actual path through free C-Space. The dashed red lines on the left and right indicate the path through free space (i.e. without C-Space transformation), sized appropriately for the robot's width.

Figure 7(b) shows the robot in a crowded street where its direct route is blocked. In this situation the computed path consists of nodes from the bounding boxes. People are passed as close as possible, instead of taking a huge detour around them as would be the case if utilizing the Voronoi nodes.

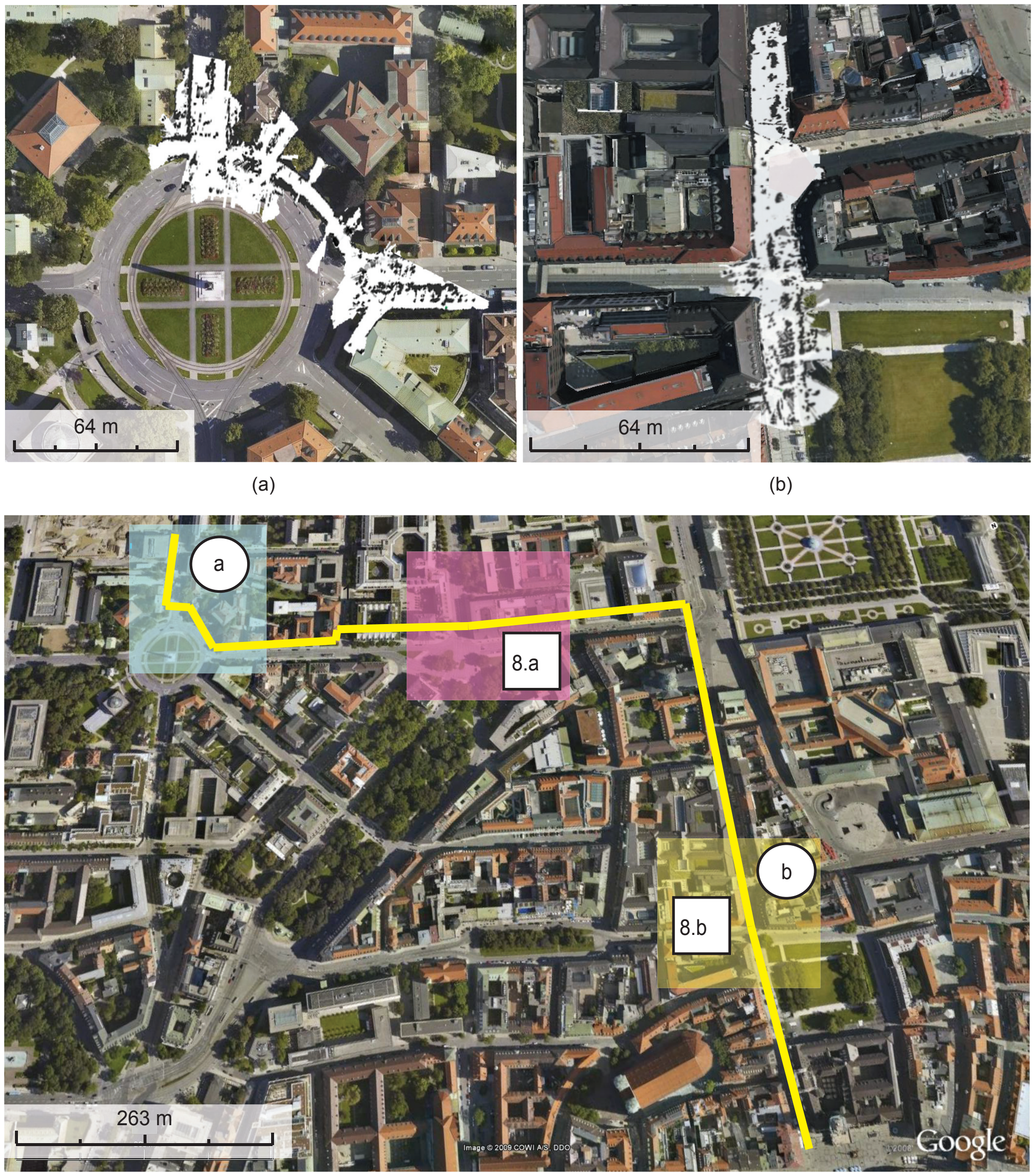

Fig. 5. Downtown area of Munich. The route ($\sim$ 1.5 km) of the robot from the Technical University of Munich to Marienplatz is indicated by the yellow line. (a)–(b): Parts of the map generated by the SLAM module during navigation that correspond to the two representative situations, which were chosen for interdependence analysis. The occupancy grids have been overlaid with satellite images taken from Google Earth to illustrate the accuracy of the maps. The squares mark the positions of the scenes shown in Figure 7.

These sample situations indicate the impact of a dynamic environment on the system's behavior. Its quantitative influence will be evaluated in the rest of this section.

5.2. System interdependencies of the ACE robot

Several considerable differences exist between the two scenes shown in Figure 6. In the first scene, which is referred to as *Sidewalk*, the robot is confined by the narrow sidewalk but the dynamic characteristics of the environment are low. In the second scene, referred to as *Pedestrian zone*, the environment is extensive but primarily characterized by high dynamics and local complexity. This can be noticed from the indicator values, introduced in Section 4, which have been sampled in both scenes at 2 Hz. Some of the values are shown in Figure 8. For example, in the *Pedestrian zone* the map uncertainty H^m and robot orientation variance var(ϕ_r) have mean values that are 43% and 45% higher, respectively. The same applies to their variance which is 6.3 and 6.5 times higher in the *Pedestrian zone*. Intuitively this can be explained by the lower dynamics in the *Sidewalk*. In

Fig. 6. Two representative situations, which were chosen for interdependence analysis. (a) Navigation on a sidewalk in a less populated district. (b) Navigation in a densely populated pedestrian zone

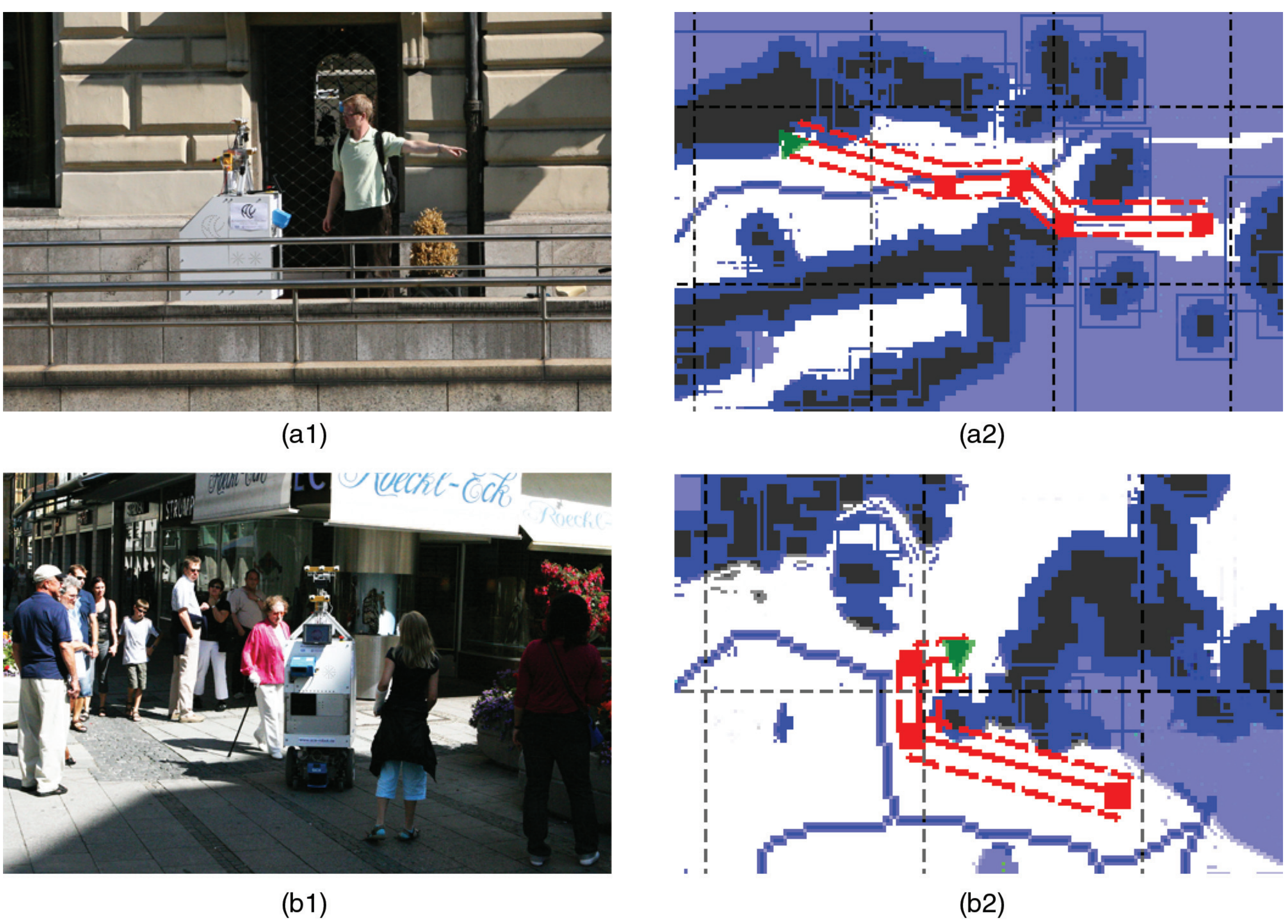

Fig. 7. Two different scenes encountered during the experiment (left) and the corresponding outputs of the path planner (right). (a) shows the robot in a narrow sidewalk and (b) in a crowded place. The coordinate raster (black lines) has a cell size of $5\,\text{m}\times5\,\text{m}$ and is used for illustrative purposes.

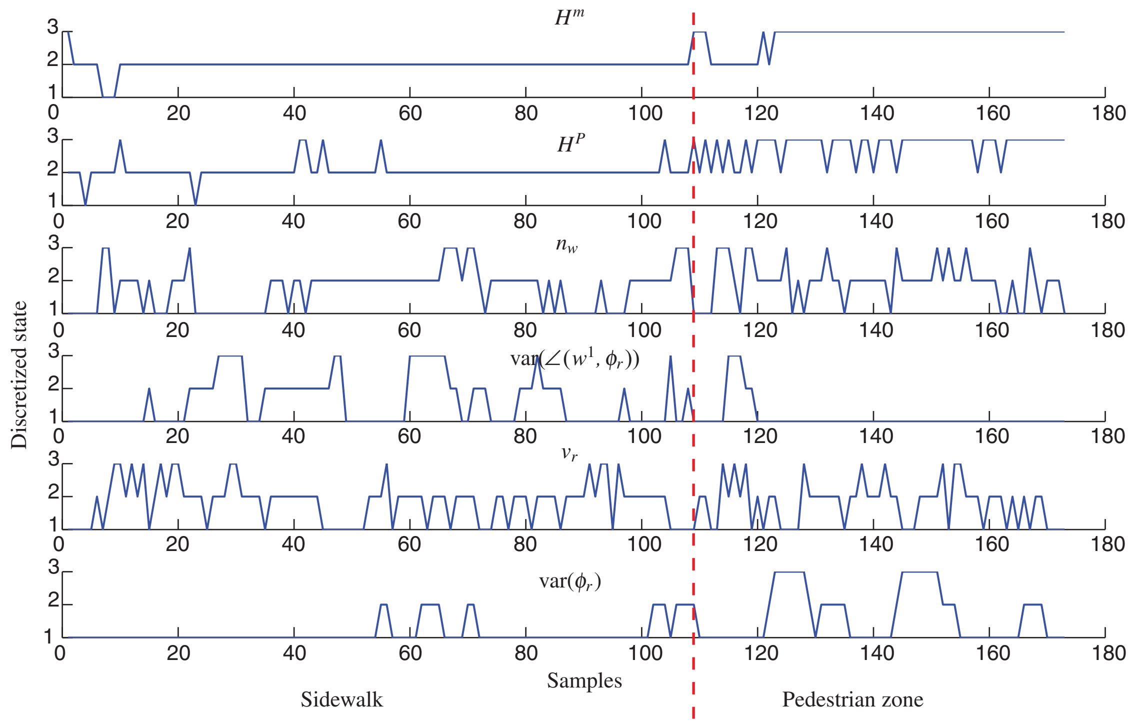

Fig. 8. Discretized indicator (vertical axis) values extracted from experimental data, for two different environments. The dashed line indicates the transition between the environments. The horizontal axis shows the consecutive sample number.

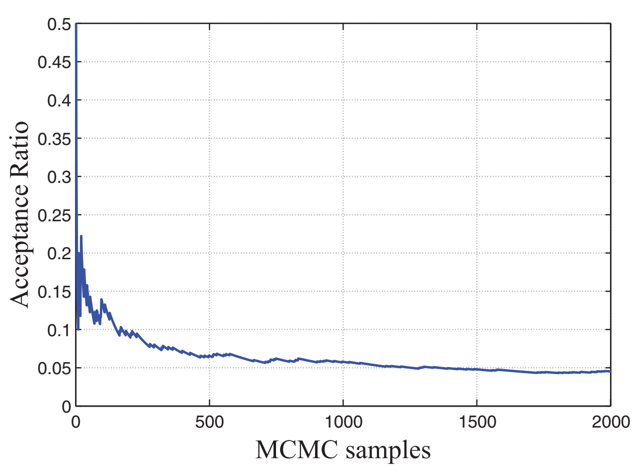

Fig. 9. Acceptance ratio versus the number of MCMC steps. The MCMC search converges since the acceptance ratio does not change after 2000 steps.

contrast, the speed of the robot v_r is on average the same. This is due to the fact that the robot's speed was limited by design for safety reasons.

Before the structure of the BN is learned, the data must be discretized and transformed into a predefined number of states. For the following results a discretization of three steps was used for all indicators.

As described in Section 3.2, in order to learn the structure of the BN that describes the interaction between indicators, the space of possible DAGs needs to be searched and the most likely structure identified using a scoring function. However, the space of possible DAGs is super-exponential with the number of variables described. In the presented case study ten indicators have been identified, leading to a search space with 4.2×10^{18} graphs, which cannot be searched exhaustively.

Therefore, an MCMC search was performed on the preprocessed data to calculate the node order for the BN. In Figure 9 the acceptance ratio versus the number of MCMC steps is illustrated. In order to converge to the most likely graph 2000 steps are needed. The resulting ordering is [H^P, n_w, H^m, s_p, var(ϕ_r), n_v, cad, var($\angle(w^1, \phi_r)$), INFO, v_r].

Using this ordering, the K2 algorithm generated the final BN, which has an improved BIC score of about approximately 6% and is shown in Figure 10. The resulting BN indicates a lot of interdependencies between the indicators but cannot express the intensity of these relations. For that reason information-theoretic criteria are applied, as described in Section 3.3.

The learned structure was utilized to train a BN with all the data. Sequential Bayesian parameter updating was performed and the respective CPTs were calculated for the network. An implementation based on the Bayes Net Toolbox for MATLAB (Murphy et al., 2001) was used. The distance metric given by Equation (4) is calculated for each possible pair of indicators. The results are illustrated in Figure 11 by the solid line.

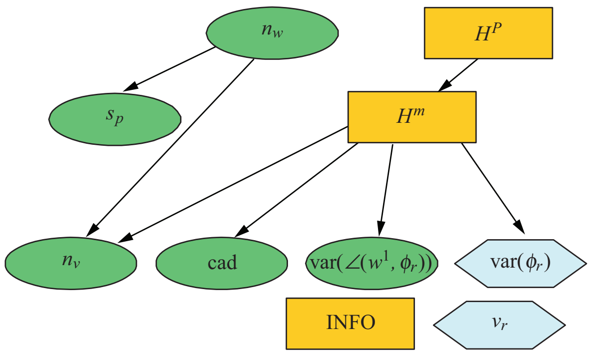

Fig. 10. Directed Acyclic Graph (DAG) learned with MCMC and K2, showing the relationships between the perceptual (rectangles), planning (ellipses) and execution (hexagons) indicators.

A strong interdependence of H^m on H^P, cad, var $(\angle(w^1, \phi_r))$ and var(ϕ_r) is observed. The relation between H^m and H^P is obvious, since without map knowledge it is impossible for the robot to localize itself. Also the influence of H^m on the planning indicators is intuitive, since the path quality is directly dependent on the map used. Map knowledge influences the planned path and therefore the motion of the robot, as reflected by the dependency between H^m and var(ϕ_r). Furthermore, n_w is strongly interconnected to n_v and s_p, which can be ascribed to the fact that all of them are indicators for the complexity of the calculated path.

The indicators INFO and v_r show no influence from and to other indicators. This means that these indicators cannot give any information about the internal system state or the influence of the environment on the system. The complexity of the system and the application domain cannot be captured by simple and purely local indicators. More specifically the velocity of the robot has been limited by design for safety reasons in most situations. It would be dangerous to allow sudden accelerations or fast speeds for the robot, in the proximity of people. Therefore, it is logically consistent that the influence from other indicators is found to be insignificant. The proposed analysis identifies, in this case, a design choice of the system.

In order to assess the environmental influence on the indicators, two additional BNs were trained using the data from the *Sidewalk* and *Pedestrian Zone* examples, respectively. A comparison of η, which is also shown in Figure 11, reveals the differences for the two scenes. A stronger influence of H^m on var$(\angle(w^1, \phi_r))$ and var(ϕ_r) in the *Pedestrian Zone* is identified. The presence of moving people results in higher map uncertainty, less directed, with a more variable planned path and consequently more complex robot motion. On the other hand, n_v is more strongly related to n_w in the *Sidewalk* scene. In this specific situation the robot has to navigate through narrow passages, where a maximum clearance path is desired. Consequently, the nodes of the Voronoi graph are more frequently used.

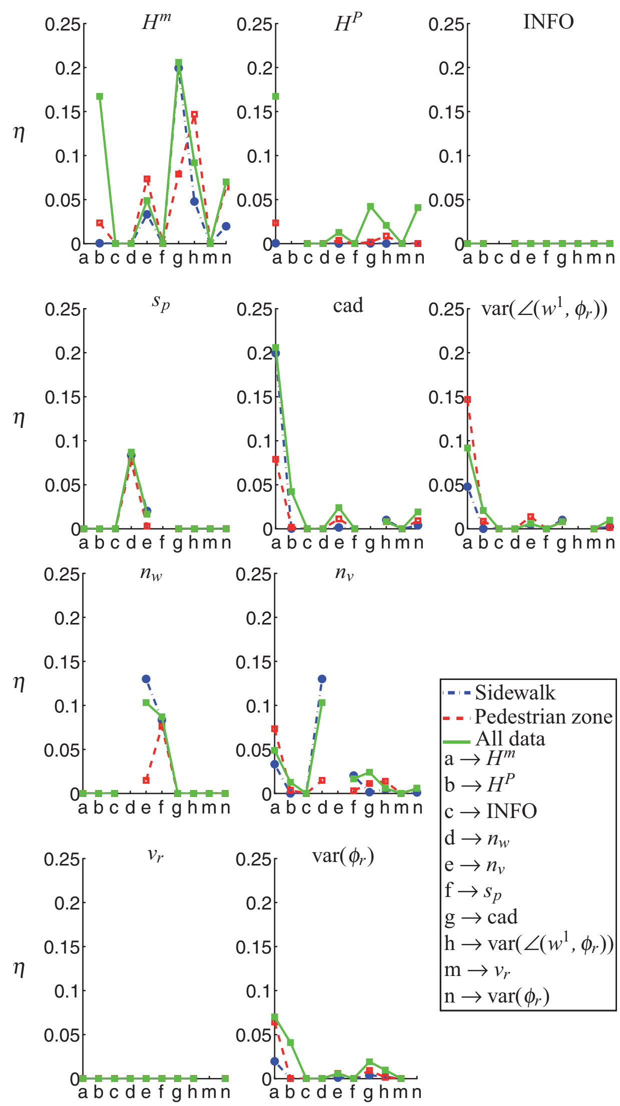

Fig. 11. Learned dependency values $\eta(X_i, X_j)$ (vertical axis) for all indicators, where the ith graph shows the dependencies of indicator i to all indicators j (horizontal axis). Since $\eta(X_i, X_i) = 1$, these values were skipped for illustrative purposes.

More detailed information about the influence of specific indicators can be extracted from the learned BNs by examining the marginal distributions of the indicators of interest while setting other indicators to specific values. This way the behavior of specific system components can be predicted for various environments and the robustness of the system can be evaluated.

This is shown for the influence of H^m on H^P and var(ϕ_r). Figure 12 shows the marginal distributions, which are calculated from the learned BN by applying Bayesian inference, for all assigned values of H^m. When map uncertainty increases, H^P also increases. The learned BN captures the interconnection between localization and mapping, which constitutes the SLAM problem. When perceptual uncertainty increases, the motion of the robot becomes more variable as indicated by the uniformly distributed predicted

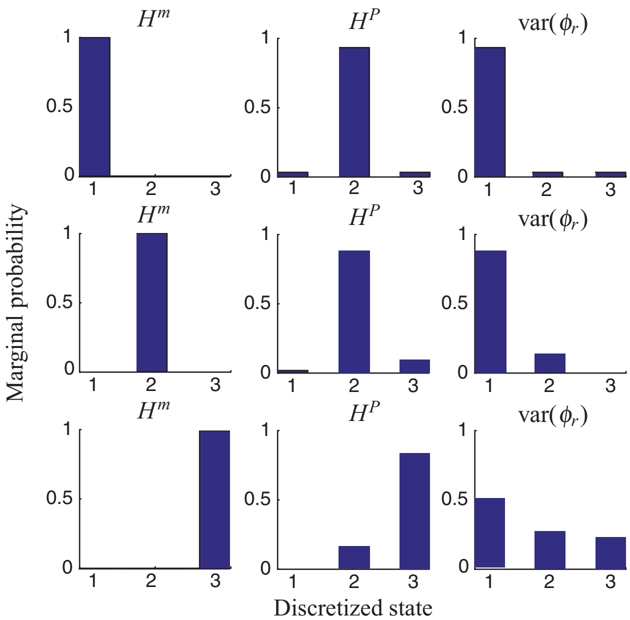

Fig. 12. The marginal distributions of the dependent indicators H^P and var(ϕ_r) as calculated from the learned BN, for assigned values of H^m.

states of var(ϕ_r). However, it can be seen that even with high uncertainty it is predicted that the variance of the robot's motion will not be unacceptably high for most situations. Therefore, the planning algorithms can be expected to be robust even with uncertain environment models. Such information is very useful for making design choices. For example, if it was predicted by the learned BN that even with low perceptual uncertainty the generated path of the robot will be very variable, then the system designer would have to reconsider the planning algorithms. In the case of ACE, it has been determined that the performance of all system components is sufficient in both environments.

In summary, the interdependence analysis of system state indicators and the environment identified map uncertainty H^m as an indicator with a very strong influence on the system. Consequently, the intuitive assumption is verified that knowledge of the environment – in this case map knowledge – is a crucial factor for the robustness of an autonomous robotic system. Also it is shown that simpler and local complexity indicators such as v_r and INFO cannot characterize the behavior of the ACE robot. In general, by using the proposed method for system analysis, several indicators can be tested in respect to their representation ability. By using the learned BN and inference techniques, predictions can be made about the behavior of performance indicators given the values of others as evidence. However, the results of the analysis reflect only the system interdependencies in the examined environments and for the executed tasks. Even though these may provide an indication of the system's behavior in different environments or for different tasks, a direct transfer is not coherent in general. Instead, new system data needs to be gathered followed by a reapplication of the analysis.

6. Conclusion and future work

A method for system interdependence analysis has been introduced. It aims at learning and quantitatively evaluating the coherence between performance indicators of different system components of autonomous robots, as well as the influence of environmental parameters on the system. The presented approach allows the identification of the limitations of an autonomous robotic system. The complexity of the environment determines the requirements for the robotic hardware and algorithms necessary to perform a given task. Conversely, the capabilities of a robotic system define the environments where it can operate and the tasks it can handle. In this respect the knowledge gained is useful for system redesign, and also during system operation. This way the robot can anticipate failures, by predicting the effects that its actions would have and correctly adjusting its behavior. The proposed analysis provides an alternative to deriving the deterministic system model, which may be quite hard for complex systems, or it can be used to verify the latter.

To validate the proposed approach, component performance indicators for the navigation system of the autonomous mobile robot ACE were derived and system interdependence analysis was performed based on experimental data from an extended field experiment. For this specific system, it has been shown that some of the proposed indicators have very strong representational capabilities, for example, the map uncertainty. At the same time indicators have been proved to be unsuitable for the mutual performance evaluation of the system's components, for example, the robot's speed. Furthermore, the influence of the environment on the performance indicators has been identified. Such knowledge is primarily useful for the improvement of the examined system itself but it is also transferable to similar systems, at least qualitatively. A quantitative transfer would be only valid under identical circumstances, which is hard to guarantee for practical systems.

Further steps should concentrate on the generality of indicators, in the sense that some of them are suitable for representing the performance of systems designed for other purposes. This would allow the specification of application-independent benchmark tests with respect to system robustness, in order to facilitate system comparability. Concerning the method itself, different algorithms for the BN structure search may be evaluated, for example, to see whether they provide a better tradeoff between complexity and solution quality. This would improve the scalability of the approach. Additionally, instead of a static network, a dynamic Bayesian network may be learned, which would also allow the examination of temporal interdependencies of dynamic systems. The discretization also has scope for future research. For example, methods to determine an optimal discretization; networks with continuous nodes may also be considered.

Acknowledgments

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys* (www.cotesys.org) and the *EU STREP project IURO* (www.iuro-project.eu). The authors would like to thank Dr Kühnlenz and the rest of the ACE team for their support during the field experiments.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

Anderson G and Gang Y (2007) A proposed measure of environmental complexity for robotic applications. *Proceedings of the International Conference on Systems, Man and Cybernetics*, Montreal, pp. 2727–2732.

Bauer A, Klasing K, Lidoris G, Mühlbauer Q, Rohrmüller F, Sosnowski S, Xu T, Kühnlenz K, Wollherr D and Buss M (2009) The Autonomous City Explorer: towards natural human–robot interaction in urban environments. *International Journal of Social Robotics* 1(2): 127–40.

Christensen HI and Förstner W (1997) Performance characteristics of vision algorithms. *Machine Vision and Applications* 9(5-6): 215–18.

Clarke E and Barton B (2000) Entropy and MDL discretization of continuous variables for Bayesian belief networks. *International Journal of Intelligent Systems* 15(1): 61–92.

Cooper G and Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9(4): 309–47.

Cover T and Thomas J (1991) *Elements of information theory*. John Wiley & Sons.

Crandall J and Goodrich M (2003) Measuring the intelligence of a robot and its interface. *Proceedings of Performance Metrics for Intelligent Systems (PerMIS)*, Gaithersburg.

Cuisenaire O (1999) Distance transformations: fast algorithms and applications to medical image processing. PhD thesis, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium.

DARPA (2007) Urban challenge – rules.

Held J, Lampe A and Chatila R (2006) Linking mobile robot performances with the environment using system maps. *Proceedings of the International Conference on Intelligent Robots and Systems*, Beijing.

Huang H, Messina E, Wade R, English R, Novak B and Albus J (2004) Autonomy measures for robots. *Proceedings of the International Mechanical Engineering Congress & Exposition*, Anaheim.

Jacoff A, Messina E and Evans J (2002) Performance evaluation of autonomous mobile robots. *Industrial Robot: An International Journal* 29(3): 259–67.

Lampe A and Chatila R (2006) Performance measure for the evaluation of mobile robot autonomy. *Proceedings of the International Conference on Robotics and Automation*, Orlando.

LaValle SM (2006) *Planning Algorithms*. Cambridge, UK: Cambridge University Press. Available at http://planning.cs.uiuc.edu/.

Lidoris G, Rohrmüller F, Wollherr D and Buss M (2009) The Autonomous City Explorer (ACE) project – mobile robot navigation in highly populated urban environments. *Proceedings of the International Conference on Robotics and Automation*, Kobe.

Munoz N, Valencia J and Londono N (2007) Evaluation of navigation of an autonomous mobile robot. *Proceedings of the Workshop on Performance Metrics for Intelligence Systems*, Washington.

Murphy K (1999) Bayesian map learning in dynamic environments. *Advances in Neural Information Processing Systems*.

Murphy K, et al. (2001) The Bayes net toolbox for MATLAB. *Computing Science and Statistics* 33(2): 1024–34.

Philippsen R and Siegwart R (2003) Smooth and efficient obstacle avoidance for a tour guide robot. *International Conference on Robotics and Automation*, Taipei.

Robinson R (1977) Counting unlabeled acyclic digraphs. *Combinatorial Mathematics V* 622/1977: 28–43.

RoboCup (2009) Robocupjunior soccer rules 2009.

Rohrmüller F, Kourakos O, Rambow M, Brščić D, Wollherr D, Hirche S and Buss M (2010) Interconnected performance optimization in complex robotic systems. *Proceedings of the International Conference on Intelligent Robots and Systems*, Taipei.

Ross S, Pineau J, Paquet S and Chaib-draa B (2008) Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research* 32(1): 663–704.

Roy N, Burgard W, Fox D and Thrun S (1999) Coastal navigation: mobile robot navigation with uncertainty in dynamic environments. *Proceedings of the International Conference on Robotics and Automation*, Detroit.

Russell S and Norvig P (2002) *Artificial intelligence: a modern approach*. Prentice-Hall.

Schwarz G (1978) Estimating the dimension of a model. *Annals of Statistics* 6(2): 461–4.

Srinivasa S, Ferguson D, Weghe M, Diankov R, Berenson D, Helfrich C and Strasdat H (2008) The robotic busboy: steps towards developing a mobile robotic home assistant. *Proceedings of the International Conference on Intelligent Autonomous Systems*, Baden.

Stachniss C, Grisetti G and Burgard W (2005) Information gain-based exploration using Rao-Blackwellized particle filters. *Robotics Science and Systems*, Cambridge.

Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G, et al. (2006) Stanley: the robot that won the DARPA Grand Challenge. *Journal of Field Robotics* 23(9): 661–92.

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, et al. (2008) Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Robotic Systems* 25(8): 425–66.

van den Berg J, Ferguson D and Kuffner J (2006) Anytime path planning and replanning in dynamic environments. *Proceedings of the International Conference on Robotics and Automation*, Orlando.