The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Interconnected Performance Optimization in Complex Robotic Systems

Florian Rohrmüller, Omiros Kourakos, Matthias Rambow
Dražen Brščić, Dirk Wollherr, Sandra Hirche, Martin Buss

Abstract— **The overall performance of a robotic system is commonly expressed by a single scenario-specific metric which is supposed to be optimized. However, the metric describing the performance of a single subtask within a scenario may be different. Nevertheless, the scenario performance is most likely dependent on the subtask performances but a mutual transformation is not straightforward in general, especially in complex robotic systems. This leads to what we call the common pricing problem, i.e. the problem to determine the functional relationship among a set of different performance criteria and then account for this relationship in the various optimizations throughout all system layers. In this paper we present an approach to first learn a probabilistic model of the metric interdependencies, and thereafter utilize this model for performance estimation and optimal task parameterization during planning and execution respectively. The proposed method is validated in a simulation.**

I. INTRODUCTION

Nowadays the diversity of available robotic hardware is growing rapidly leading to robotic systems with stronger heterogeneity. Accordingly, robots become more and more functional resulting in a larger variety of performable tasks and in increasing multi-tasking capabilities.

While this enables a wide scope of possible robotic applications, it also poses strong demands to the respective control framework. To really exploit its available capabilities and maximize the overall system performance, a robot needs to possess an operational framework that enables an efficient resource allocation and task execution. The system performance is in general scenario specific and can for example be the completion time as in the DARPA Grand Challenge [1] or the number of scored goals as in the RoboCupSoccer [2]. While such scenario criteria represent a mean to measure the global performance, respective local performance criteria may be entirely different when considering the specific subtasks the robot needs to perform in order to fulfill the overall task. The performance of a localization subtask for example is commonly measured by the uncertainty of its pose estimate [3]. Especially in mobile robots the localization performance indirectly influences the global metric [4], such as completion time or the number of scored goals. Another case are subtasks with conflicting goals. The obstacle-avoidance subtask may have a negative influence on the overall performance, but it is crucial for the system safety. As a result, there exists an interdependency between the performance metrics of each module as well as the overall system performance metric. Another domain where these interdependencies are apparent are multi-robot systems. In these systems a complex task can be decomposed in many simpler highly diverse tasks where each might have a different performance criterion. However, a common single scenario metric is required in order to solve the task allocation problem [5].

Incorporating these metric interdependencies into the various optimizations occurring throughout the system is usually not straightforward. One possibility is to formulate a multi-objective optimization problem and solve it by identifying the Pareto-optimal frontier. Evolutionary algorithms are a common approach to this problem [6], [7], but they usually are computationally expensive. Additionally, these algorithms assume that the functional relationship among the different objectives is given, which is not the case in general. An analytic relation between the different metrics might be hard to identify or too complex to model. A straightforward and frequently used approach to model this relationship is to simply combine the metrics by a subjective choice of weights from the designer [8]–[10]. Thereby these approaches reduce the problem to a single-objective optimization problem whose solution is biased by the chosen weights. There exist recent approaches that incorporate estimation methods for those unknown functions [11], but they concentrate only on identifying the Pareto-optimal set and not on explicitly determining the functional relationships.

In this paper we address both the problem of how to model the functional interdependencies between the various performance metrics throughout the system and how to use this model during system operation to optimize the performance. We call this problem the common pricing problem and we tackle it using a probabilistic approach. First, a probabilistic model of the interdependencies is learned. The generation of the model is based on [4] which provides a quantitative determination of the interdependencies within a set of arbitrary metrics through the usage of Bayesian Networks (BN). Furthermore, a method to tightly connect the planning with the task execution of a robotic system is proposed. The probabilistic model is used to provide a performance estimation of the common global performance metric to the planning part and optimal task parameterization to the task execution part. The presented approach provides a generic solution to the common pricing problem that allows for a modular task-oriented system design.

The remainder of the paper is organized as follows: In

Florian Rohrmüller, Omiros Kourakos, Matthias Rambow, Dražen Brščić, Dirk Wollherr, Sandra Hirche and Martin Buss are with the Institute of Automatic Control Engineering (LSR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany {rohrmueller, omiros.kourakos, rambow, drazen, dw, hirche, mb}@tum.de.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

Section II the problem setting is presented, followed by the learning mechanism for the performance interdependencies in Section III and the performance estimation and task parametrization in Section IV. Section V presents simulation results for a box-pushing scenario.

II. Problem Setting

In order to cope with the complexity robotic systems are commonly composed of various modules responsible for different types of tasks. Each module i is parameterized to optimize some module-specific objective c_i. In general those modules share common variables, which can be the mutual input/output but also some system state or parameter. From this sharing of information results a dependence of the performance c_j of module j on the performance c_i of another module i. This leads to a performance interdependence throughout the system which needs to be taken into account during performance optimization. More specifically, let the vector $\mathbf{c} = (c_0(), \ldots, c_l())^T$ be the combination of all $l+1$ performance criteria in the system. The performance $c_i(\Theta, \mathbf{c}'_i)$ of a module i is dependent on a vector $\Theta = (\theta_1, \ldots, \theta_m)^T$ of m adjustable system parameters θ and on the vector $\mathbf{c}'_i = (c_0(), \ldots, c_{i-1}(), c_{i+1}, \ldots, c_l())^T$ of all other performance criteria.

Optimizing $\mathbf{c}$ represents a multi-objective optimization problem. A solution that minimizes all $c_i, i \in \{0, \ldots, l\}$, can not be found in general, but it is rather given by the optimization of one criteria under constraints $\mathbf{k} = [k_1, .., k_l]$ on the others:

$$\begin{aligned} &\min_{\Theta} c_0(\Theta, \mathbf{c}'_0) \\ &\text{subject to } c_j(\Theta, \mathbf{c}'_j) \leq k_j,\ \forall j \in \{1, \ldots, l\}. \end{aligned} \tag{1}$$

In case the analytic forms of $c_i(\Theta, \mathbf{c}'_i)$ are known for all i, the metric interdependencies, at least part of them, could be simply removed by substitution. However, a major problem of complex robotic systems is that this functional relationship is often hard to determine or too complex to be modeled. In other words the relationship $c_i(\Theta, \mathbf{c}'_i)$ is usually not known; so its derivation is the first core problem addressed in this paper.

The mapping of global and local performance metrics to the indices i is not fixed but rather dependent on the current operational phase of the system. This leads to the second major problem in our focus, namely how to utilize a derived $\mathbf{c}$ during the planning and execution phases of the system.

In order to address these problems, we utilize the probabilistic approach, shown in Fig. 1 on the right, to learn $\mathbf{c}$ from experimental data (Sec. III). This learned functional interdependence is thereafter utilized during system operation, shown in Fig. 1 on the left, to solve the multi-objective optimization. The latter is split into a planning step (Sec. IV-A) and an execution step (Sec. IV-B), which differ with respect to the ordering of the criteria and the used constraints.

III. Learning of the performance dependencies

As mentioned above an exact description of the interdependence between different metrics is in general not known. Consider for example a mobile robot that is supposed to push a box along a marked path. For this it needs to track the path as well as the box with a camera. Assume the global performance metric is the distance reached within a specified time. In contrast, the performance of a tracking module is commonly expressed by the accuracy of its pose estimate. The reached distance is likely to be dependent on the achieved tracking accuracy, but to model this relation is certainly not straightforward. Therefore a method is required to obtain the functional relationship $c_i(\Theta, \mathbf{c}'_i)$ for all i within a set of arbitrary metrics $\mathbf{c}$.

In this respect, we utilize the probabilistic system interdependence analysis described in [4] to learn a probabilistic model of the metric interdependencies. First a set of performance metrics needs to be specified for each task the robot can perform. During system operation these criteria are permanently computed and logged. The collected performance data serve as input for the performance interdependence analysis. The metric values are first discretized and then a Bayesian Network (BN) structure, which best reflects the interdependencies between the metrics, is searched. As proposed in [4], a combination of a Markov Chain Monte Carlo [12] and K2 [13] search is used to identify the best structure. As quality measure the Bayesian Information Criterion (BIC) [14] is used. The BIC evaluates the likelihood of the data being generated by the given structure, penalized by the complexity of the structure. The structure with the highest BIC value is chosen and then trained with the gathered metric data by sequential parameter update.

The number and size of the intervals used for the discretization of the metric values should not be too small, in order to maintain the contained information. However, in case they are set too large, the respective probability distributions become too flat what makes the determination of mutual interdependencies hard, as well. A possible solution to select the intervals is for example to use an entropy-based approach, such as [15].

While the BN indicates only the qualitative interdependence between the metrics, a quantitative evaluation is obtained using information-theoretic analysis based on the relative mutual information

$$\eta(X,Y) = \frac{I(X,Y)}{H(X,Y)}, \tag{2}$$

where $I(X,Y)$ is the mutual information of X and Y and $H(X,Y)$ their joint entropy. The relative information entropy $\eta(\cdot,\cdot)$ satisfies the requirements of a distance metric and thus can be used to identify and compare the strength of the interdependencies among all pairs within a set of metrics. The BN is subsequently used during system operation as described next.

IV. Performance Estimation and Task Parameterization

After the offline learning and training of the BN, inference can be applied to it to determine how a change of one metric affects the state of another one. This is used online for the performance estimation and task parameterization.

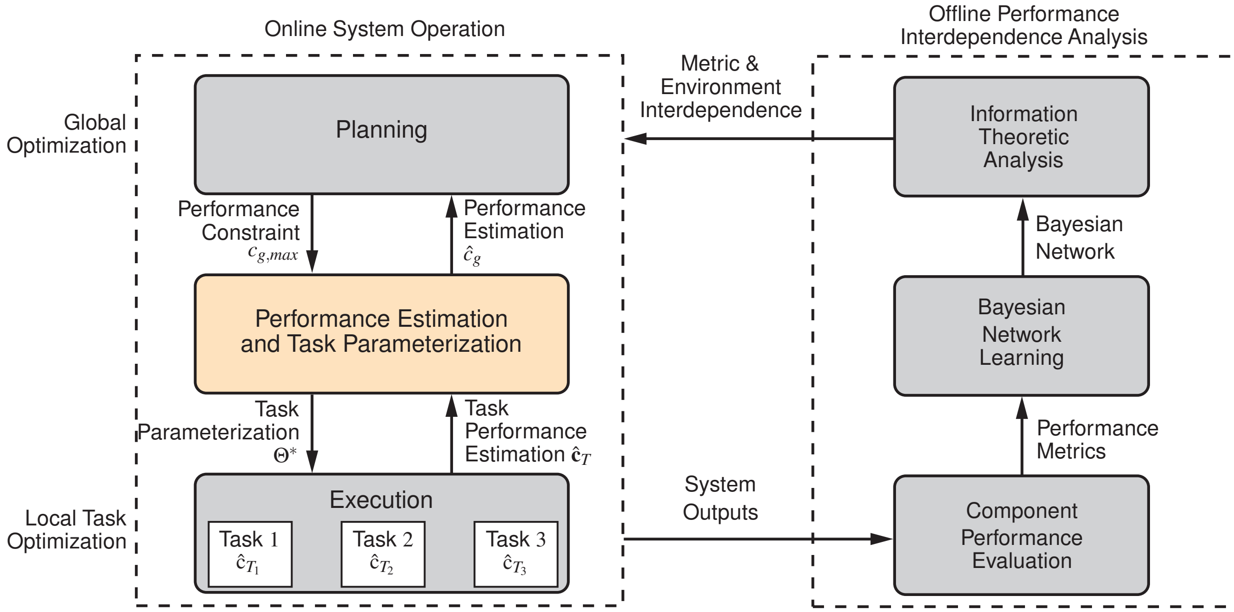


Fig. 1. Overview of the presented approach: the performance estimation and task parameterization utilizes the metric interdependencies of the performance interdependence analysis [4] to (i) provide a performance estimate to the planning module and (ii) optimize the task parameters of the execution.

A. *Performance Estimation*

During the planning phase the robot aims to determine from a set of possible plans the one which yields in the best achievable performance. The respective criteria is scenario specific and commonly reflects the progress of the overall global mission. In the following it is referred to as global cost c_g and w.l.o.g. we set $\mathbf{c} = (c_g, \mathbf{c}_T)$ with $c_g = c_0$ the global metric and $\mathbf{c}_T = (c_1, ...c_l)$ the local ones.

For the estimation of c_g inference on the trained BN is used. The conditional probabilities $Pr(c_i = u_i \mid \Theta = z_k, c_j = u_j)$ for a metric c_i given metric c_j and parameters Θ are calculated. This is done for all values u_i and u_j the metrics and values z_k the parameters can take according to their discretization. In the following C_i denotes the discrete probability distribution of a discretized metric c_i.

These distributions are used to derive the performance estimate

$$\hat{c}_g = \min_{\Theta} \mathrm{E}\left[C_g \mid \Theta, C_T(\Theta)\right], \tag{3}$$

where $\hat{c} = \mathrm{E}[C]$ is the expected value of a probability distribution C. Equation (3) is a special case of (1) with $c_0 = c_g$ and equality constraints $k_T = \mathrm{E}[C_T(\Theta)]$ for $T \in \{1, \ldots, l\}$.

Based on $\hat{c}_g$ the planning module derives the best plan and forwards it to the execution modules.

B. *Optimal Task Parameterization under Global Metric Constraints*

After the planning phase is completed, the planner assigns the task to the task execution. In this section it is assumed that the planner has generated a plan that requires a global performance c_g that is no worse than $c_{g,max}$ in order to classify the task as successful.

During the execution phase all modules required to perform a task are active. If the metric interdependencies are neglected, then each module will try to optimize their specific c_T. This optimization represents another case of (1) with a different ordering compared to Sec. IV-A. However, neglecting the metric interdependencies and directly optimizing c_T may likely result in an exceedance of the worst case global performance metric $c_{g,max}$. In consequence, to ensure that the previous worst case global performance requirement is met, the system interdependencies are taken into account and an inequality constraint $k_g = c_{g,max} > \hat{c}_g$ according to (3) is set. An additional reason to set a worst case global performance constraint is to improve robustness by allowing room for local optimization when external disturbances are apparent.

Continuing with the previous example of the box-pushing robot, c_g corresponds to the negative distance the box has been pushed after a specified time. The negation is only used for simplification to have a consistent cost character of the metrics. Let us further assume that the task is classified as successful as long as the robot manages to push the box farther than a threshold s_{min}. Then the planning module sets a bound $c_{g,max} = -s_{min}$ what loosens the constraints for the tracking module, since the best c_g is no longer needed to be achieved while still ensuring that the task is completed. Thereby the tracking module of the robot is able to take means to improve its pose estimate as long as $\hat{c}_g \leq c_{g,max}$ is ensured.

The used parameters Θ_T are adjusted in order to optimize the expected c_T while satisfying the global constraint k_g. Thereby the dimension of the frontier of Pareto-optimal solutions $\mathbf{c}^*$ is reduced from $\mathbb{R}^{l+1} \rightarrow \mathbb{R}^l$.

During the task execution the set

$$\Lambda_g = \left\{\Theta \mid Pr(c_g > c_{g,max} \mid \Theta) < \varepsilon\right\} \tag{4}$$

of parameters, for which the probability that c_g will exceed $c_{g,max}$ is smaller than a threshold ε, is derived. The threshold $\varepsilon \in [0,1]$ is chosen by the designer. Λ_g defines the feasible set of values that satisfy the global cost constraint. From this parameter set Λ_g the best parameters Θ that optimize the local costs must be chosen. This is a multi-objective optimization problem. One approach to determine a single Pareto-optimum for this problem, is by iteratively optimizing the local costs based on some specified priority or by iteratively calculating Λ_i for $i \in \{0,\ldots,l-1\}$:

$$\Lambda_i = \{\Theta \mid \Theta \in \Lambda_{i+1} \wedge Pr(c_i > k_i \mid \Theta) < \varepsilon\}\,. \tag{5}$$

This is sequentially reduced until a single task metric c_0 is left. Among all $\Theta \in \Lambda_0$ the parameter set

$$\Theta_0^* = \underset{\Theta \in \Lambda_0}{\operatorname{argmin}} \mathrm{E}\left[C_0 \mid \Theta\right] \tag{6}$$

which yields the best task performance c_0 is chosen. Thereby a unique point on the frontier of Pareto-optimal solutions is determined.

With (3) and (6) the required performance estimation and task parameterization are given respectively. The proposed approach considers the system interdependencies as black-box functions which need to be approximated. The structure learning of the Bayesian Network results in a model that captures the most important interactions between the variables while trying to keep the modeling complexity low. As a result even complex relations can be modeled. This makes the proposed approach very flexible, especially for modular design of robotic tasks. However, the learning of the structure of Bayesian Network is computationally expensive and can only be performed offline. The presented approach has been validated in a simulated box-pushing scenario, as explained in the next section.

V. VALIDATION IN SIMULATED BOX-PUSHING

One important aspect of the performance estimation and task parameterization, presented in Sec. II, is the interaction between the planning and the execution modules. We highlight the efficacy of our concept for cost determination through a box-pushing scenario which is used by several research groups in the field of multi-robot task allocation and coalition formation as a benchmark, e.g. [5], [16]. The next section describes the scenario setup followed by the results.

A. Box-Pushing Scenario

In our specific setup the target is to push a rigid box as far as possible along a straight line within a specified time without introducing any rotational or translational errors in the box pose Fig. 2. For this purpose, two separate tasks, which try to minimize the rotational and translational error respectively, are considered. Each task calculates a force F that will be exerted on the longitudinal side of the box

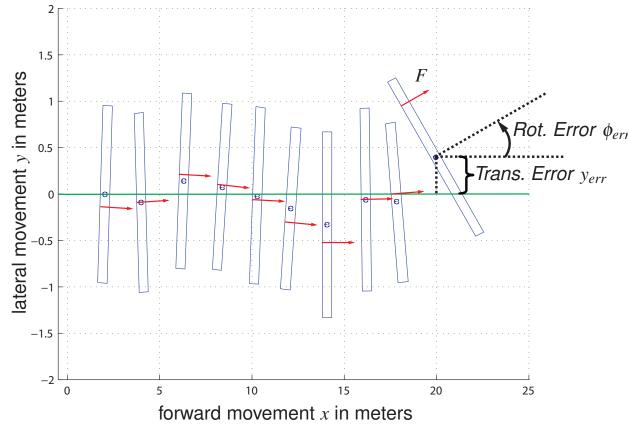


Fig. 2. Example box trajectory: the horizontal axis ($y = 0$) is the desired trajectory, the dot indicates the box center and the arrow the applied force, here shown for a single robot.

in order to affect the box position and orientation. For simplification of the robot strategy the forces are applied perpendicular to the face of the box, and can be applied to a pre-specified fixed number of contact points. The magnitude of F depends on the robot's maximum speed v_{max}, the robot's maximum force F_{max} and the current box velocity v_{box} and is modeled as

$$F = \begin{cases} \left(1 - \frac{v_{box}}{v_{max}}\right) F_{max} & \text{, if } v_{box} \leq v_{max} \\ 0 & \text{, otherwise.} \end{cases} \tag{7}$$

Note that v_{max} and F_{max} belong to the set of parameters Θ of the tasks. The desired point of contact is calculated at each time instant based on the current translational and rotational errors $y_{err} = y^* - \hat{y}(t)$ and $\phi_{err} = \phi^* - \hat{\phi}(t)$, respectively. Here $y^* = 0$ and $\phi^* = 0$ are the desired translational and rotational values of the center of mass of the box; $\hat{y}(t)$ and $\hat{\phi}(t)$ are the measured translational and rotational values differing from the true value $y(t)$ and $\phi(t)$ by zero mean Gaussian noise $n_{trans}(t)$ and $n_{rot}(t)$ with the corresponding variances σ^2_{trans} and σ^2_{rot}, i.e.

$$\begin{aligned} \hat{y}(t) &= y(t) + n_{trans}(t) \text{ and} \\ \hat{\phi}(t) &= \phi(t) + n_{rot}(t). \end{aligned}$$

If the desired contact point differs from the actual point, the robot decides to change its position and no force is exerted, otherwise it keeps pushing. In the multi-robot case, an additional criterion for choosing the desired point is whether it is already occupied by another robot or not.

For the evaluation of our approach we compare a single-robot system with a coalition of two less capable robots in terms of maximum force F_{max}. All robots have the same task knowledge, which from the perspective of the execution module means that the task plans for the box-pushing task are implemented the same way on all robots. The box is modeled as a rigid body with equally distributed mass and a velocity proportional friction. The mass and friction

$\sigma^2_{rot}[degree]$	$\sigma^2_{trans}[m]$	$F_{max} * nrRobots[N]$	$v_{max}[\frac{m}{s}]$
5	0.1	4	0.4
10	0.3	8	0.7
15	0.5	12	1

TABLE I
SIMULATION PARAMETERS

coefficient are kept constant. Additionally, the robots always stay in contact with the box. All the parameters settings used for the simulations are shown in Table I. For each possible parameter/noise configuration we run 100 simulations with a runtime of $t_{sim} = 100$ seconds each. After each simulation run we record the average translational error

$$\overline{c}_{trans} = \frac{1}{t_T} \int_0^{t_T} \|y^* - \hat{y}(t)\|^2 dt, \tag{8}$$

and the average rotational error

$$\overline{c}_{rot} = \frac{1}{t_T} \int_0^{t_T} \left\|\phi^* - \hat{\phi}(t)\right\|^2 dt, \tag{9}$$

which together form the task-specific cost vector

$$\mathrm{c}_T = \begin{pmatrix} \overline{c}_{rot} \\ \overline{c}_{trans} \end{pmatrix}. \tag{10}$$

In addition, we measure the distance the box manages to travel within the fixed simulation time. We define the common global performance metric c_g as the negative distance. Thus, minimizing the global cost c_g is equivalent to maximizing the distance $x(t = t_{sim})$ traveled within the fixed simulation time

$$\mathrm{c}_g = -x(t = t_{sim}). \tag{11}$$

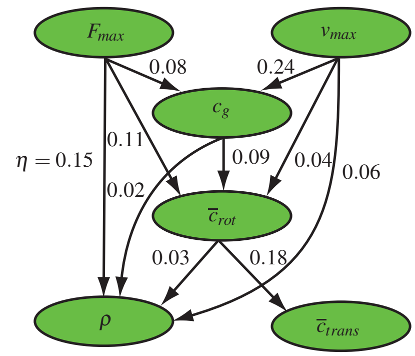
In case $\hat{\phi}(t = t_i) > \frac{\pi}{2}$ the run is considered as failed and a penalty of $c_g = 30m$ is assigned.

As a further cost metric the resource ratio ρ is introduced, which is the number of not actively pushing robots divided by the number of robots allocated to the task. In other words, frequent position switching results in a higher value of ρ.

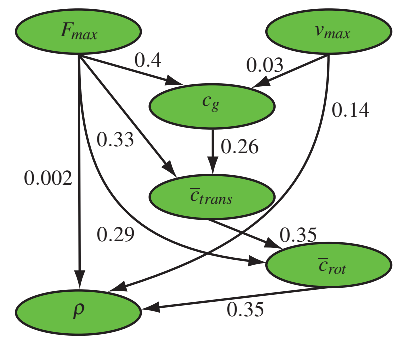
As stated in Sec. III, the metrics need to be discretized for the search and training of the Bayesian Network. Here five discretization levels are chosen.

B. Performance Interdependence Analysis

In order to obtain the mutual interdependence between the gathered metrics retrieved during the simulation runs, the system interdependence analysis of Sec. III is applied. Fig. 3 shows the learned BNs for the single-robot and the two-robot case, from which the dependencies between the parameters $\Theta = (F_{max}, v_{max})^T$ and the performance metrics c_{rot}, c_{trans} and c_g are identified qualitatively. In both BNs F_{max} as well as the v_{max} are the root nodes and influence all other metrics. This indicates the suitability of the two parameters for the system control. The main difference for the two-robot case in Fig. 3(b) compared to the single-robot case in Fig. 3(a) is the more intense relation of $\overline{c}_{trans}$ to the other metrics.



(a) Single Robot



(b) Two Robots

Fig. 3. Learned Bayesian Networks.

C. Performance Estimation and Task Parameterization

Table II and Table III present the results obtained by applying the Performance Estimation (PE) and the Task Parameterization (TP) described in Sec. IV-A and IV-B to the box-pushing scenario. For the task parameterization, $\varepsilon = 0.9$ was set and the worst case global performance bound $c_{g,max}$ was chosen equal to $-3m$ and $-6m$ for the single-robot and two-robot task, respectively. The proposed methods are compared with two alternative approaches. In the first approach the dependencies are not considered at all, i.e. the Bayesian Networks are not used (No BN). In the second approach the task specific performance metric c_{rot} is optimized without any constraint on c_g, i.e. $\min_{\Theta} \mathrm{E}[C_{rot} \mid \Theta]$, denoted as TO, Task Optimization. The columns of the tables give the expected values of the performance metrics given the set of task parameters $\Theta^* = (F^*_{max}, v^*_{max})^T$ that are optimal with regard to the respective criteria. An exception is the first row, where the expectation is taken over all observations independent of Θ, i.e. $\mathrm{E}[C_i]$, since the dependencies of the metrics upon the task parameters are not known.

In the single-robot case, the performance estimation module estimates a c_g that is 461% smaller than the estimated c_g in the TO approach where c_{rot} is optimized. This indicates that neglecting the interdependencies and only optimizing the local task metrics might have a negative impact on the global performance metric, as already mentioned in Sec. IV-B. Of course optimizing over c_g comes at the expense of the task specific performance metrics c_{rot} and c_{trans} which

	$E[C_g \mid \Theta^*]$	$E[C_{rot} \mid \Theta^*]$	$E[C_{trans} \mid \Theta^*]$	F^*_{max}	v^*_{max}
No BN	−9.2833	0.0049	0.0068	*	*
TO	−3.0002	0.0040	0.0060	4	0.4
PE	−16.8400	0.0090	0.0075	8	1
TP	−3.400	0.0040	0.0060	4	0.7

TABLE II

PERFORMANCE ESTIMATES AND OPTIMAL PARAMETER SET FOR ONE ROBOT.

	$E[C_g \mid \Theta^*]$	$E[C_{rot} \mid \Theta^*]$	$E[C_{trans} \mid \Theta^*]$	F^*_{max}	v^*_{max}
No BN	5.2141	0.1318	0.2041	*	*
TO	−6.0010	0.0072	0.0068	2	0.7
PE	−12.4100	0.0192	0.0075	4	0.7
TP	−6.3500	0.0072	0.0068	2	1.0

TABLE III

PERFORMANCE ESTIMATES AND OPTIMAL PARAMETER SET FOR TWO ROBOTS

increase by 125% and 25% respectively. Accordingly, for the two-robot case, the c_g is 106% smaller, while c_{rot} and c_{trans} deteriorate by 166.6% and 10.2%, respectively, when comparing the PE to the TO approach. From these examples, the trade-off between global task performance and local task performance metrics is clear. When the proposed task parameterization (TP) is applied, the estimated c_g significantly increases compared to the PE approach for both the single-robot and two-robot cases. However, the worst case of $c_{g,max}$ being equal to $-3m$ and $-6m$, respectively, is guaranteed with a probability of 90%. Additionally, the task specific costs are much smaller than in the PE approach. While the performance estimation module offers the planner the best achievable global performance estimate, the task parameterization allows the system to trade-off between local and global task performances.

In the approach where no dependencies have been learned (No BN), the robot is not aware of the relation between the performance metrics and the task parameters. If the task parameter choices have a significant impact on the global performance, then a bad estimate is computed. For example, in the two-robot case, if the task parameters are chosen aggressively, i.e. very high values of both F_{max} and v_{max}, then there is a high failure rate of the task. This high failure rate results in a very high value of the estimated global performance metric. As a result, a planning module would deduce that using two robots to complete the pushing task is much worse than using a single robot. However, if the dependencies are taken into account, in the two-robot case a better worst case performance $c_{g,max} = -6$ than the single-robot case $c_{g,max} = -3$ (TP in Table III and II, respectively) can be guaranteed.

VI. CONCLUSION

The paper addresses the problem of achieving coherence among the various performance optimizations occurring throughout the subtask modules of a complex robotic system. A probabilistic approach is presented, that handles the problem by learning the interdependencies among the performance metrics from gathered system data. Thereafter, the learned model is used to tightly couple the optimizations in the planning and execution layers of a robotic system by solving a multi-objective optimization. This is achieved during system operation by inferring a global cost and by optimizing task parameters while still guaranteeing performance bounds of higher prioritized tasks. The method is systematic in the sense that it can combine metrics of different units without requiring any subjective intervention of the system designer.

Future work will focus on investigating the scalability of the method and the use of incremental learning of the parameters of the Bayesian Network.

VII. ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys* (`www.cotesys.org`).

REFERENCES

[1] DARPA, "Urban challenge - rules," 2007.
[2] "Robocupjunior soccer rules 2009," 2009.
[3] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation - mobile robot navigation with uncertainty in dynamic environments," 1999.
[4] F. Rohrmüller, G. Lidoris, D. Wollherr, and M. Buss, "System interdependence analysis for autonomous mobile robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.
[5] B. P. Gerkey and M. J. Matarić, "Sold!: Auction Methods for Multi-robot Coordination," *IEEE Transactions On Robotics And Automation*, vol. 18, October 2002.
[6] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, pp. 369–395, 2004.
[7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," Tech. Rep. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
[8] J. Andersson, "A survey of multiobjective optimization in engineering design," tech. rep., Department of Mechanical Engineering, Linktjping University. Sweden, 2000.
[9] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2366 – 2371, May 2006.
[10] G. Rodriguez and C. R. Weisbin, "A New Method to Evaluate Human-Robot System Performance," *Autonomous Robots*, vol. 14, no. 2-3, pp. 165–178, 2003.
[11] S. Shan and G. G. Wang, "An efficient pareto set identification approach for multi-objective optimization on black-box functions," *Journal of Mechanical Design*, vol. 127, pp. 866–874, 2005.
[12] K. Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems*, 1999.
[13] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, 1992.
[14] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, 1978.
[15] E. Clarke and B. Barton, "Entropy and MDL discretization of continuous variables for Bayesian belief networks," *International Journal of Intelligent Systems*, vol. 15, no. 1, pp. 61–92, 2000.
[16] B. Kannan and L. E. Parker, "Metrics for quantifying system performance in intelligent, fault-tolerant multi-robot teams," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.