

Context Identification for Efficient Multiple-Model State Estimation of Systems With Cyclical Intermittent Dynamics

Sarjoun Skaff, *Member, IEEE*, Alfred A. Rizzi, *Member, IEEE*, Howie Choset, *Member, IEEE*, and Matthew Tesch, *Student Member, IEEE*

Abstract—This paper presents an approach to accurate and scalable multiple-model (MM) state estimation for hybrid systems with intermittent, cyclical, multimodal dynamics. The approach consists of using discrete-state estimation to identify a system's dynamical and behavioral contexts and determine which motion models appropriately represent current dynamics and which individual and MM filters are appropriate for state estimation. Furthermore, the hierarchical structure of the dynamics is explicitly encoded, which enables detection not only of rapid transitions between motion models but of higher level behavioral transitions as well. This improves the accuracy and scalability of conventional MM state estimation, which is demonstrated experimentally on a mobile robot that exhibits fast-switching, multimodal dynamics.

Index Terms—Hidden Markov models, hybrid estimation, multiple-model (MM) filtering, timed automata.

I. INTRODUCTION

ACCURATE and scalable state estimation for hybrid systems with cyclical intermittent dynamics is a key-enabling technology for reactive control and system health monitoring. A motivating example of such a system is a legged mobile robot that exhibits different behaviors, such as walking and jogging, and requires robust state estimation for successful control.

In order to accurately estimate the state of these systems, this paper presents a novel, hierarchical approach that combines classification with discrete and continuous state-estimation techniques. This approach is directed at hybrid systems with fast-switching multimodal dynamics and excels in situations where conventional continuous-only approaches fail. This paper also introduces the concept of explicitly tying motion models to the dynamics they represent and demonstrates in experiment that this concept achieves significant accuracy and scalability gains for multiple-model (MM) filtering systems.

Conventional state-estimation approaches consist of representing multimodal dynamics with a collection of motion models and performing state estimation with MM filters [22], [24].

The MM approach associates multiple Kalman filters (KFs) with each mode and runs all filters simultaneously. It averages the individual filters' output weighted by their relative likelihood and generates a consolidated state estimate.

Unfortunately, the conventional MM approach has three main problems when applied to hybrid systems, such as mobile robots. First, this approach requires the activation of the entire set of filters, which can become computationally intractable for robots with a large number of modes. Second, MM filters can produce inaccurate state estimates when sudden changes in the dynamics lead to sharp and recurrent variations in model accuracy, as individual filters may not converge at the same rate at which the dynamics change. Third, MM filters can generate inaccurate state estimates in the presence of dynamics that are not appropriately represented by any of the available models; in realistic settings, these unmodeled dynamics, such as disturbances and transients, can dominate locomotion dynamics.

This research addresses these shortcomings by leveraging the state-estimation designer's *a priori* knowledge of the discrete structure of the dynamics to help MM filters select appropriate motion models. The approach consists of representing the dynamics with a *hierarchical structure*, identifying the current structure by recognizing patterns in sensor data that correspond to specific dynamics, and inferring from the structure which models, if any, are appropriate for estimation. This approach constitutes a framework in which information about the discrete component of hybrid dynamics is combined with information about the continuous component to improve the overall estimation accuracy and scalability.

Such a hybrid discrete-continuous approach has been adopted by others, such as Hofbaur and Williams [9] and Verma [31], but their work does not exploit the discrete hierarchy inherent to cyclical intermittent dynamics. This research focuses on systems, which exhibit behaviors, such as walking and jogging, that introduce cyclic temporal structure to their dynamics. This structure is identified and then used to improve the accuracy of estimation and enable the rapid convergence of filters during frequent discrete state transitions.

The following sections describe the problem addressed by this research, provide a brief overview of related research in continuous-, discrete-, and hybrid-state estimation, describe the context-based estimation framework, provide empirical examples of the limitations of conventional estimation, and demonstrate in experiment that context-based estimation improves the accuracy and scalability of MM estimation systems.

Manuscript received April 30, 2010; accepted August 25, 2010. Date of publication October 7, 2010; date of current version February 9, 2011. This paper was recommended for publication by Associate Editor K. Yamane and Editor J.-P. Laumond upon evaluation of the reviewers' comments.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: sarjoun@ri.cmu.edu; arizzi@ri.cmu.edu; choset@ri.cmu.edu; mtesch@ri.cmu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2010.2073011

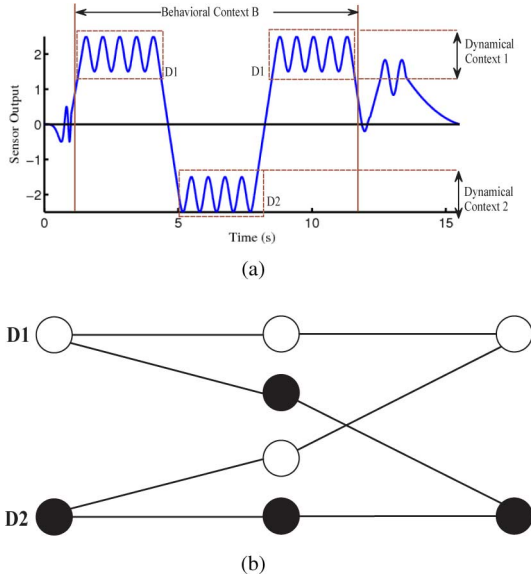


Fig. 1. Dynamics are represented with a hierarchy of dynamical and behavioral contexts. (a) Two-mode sensor output profile. (b) Two-mode MM system. To prevent exponential increases in the number of filters, the hypotheses are collapsed during each iteration (GPB2).

II. PROBLEM STATEMENT

The problem and its solution are described in more detail with the help of a generic hybrid system whose sensor output profile is described by Fig. 1(a). The assumption is that available models appropriately represent the system's dynamic modes D1 and D2, which, e.g., might be free flight and ground stance. The state can be estimated with the MM filters based on D1 and D2 models [see Fig. 1(b)] under two conditions. First, the system should primarily exhibit the dynamics described by D1 and D2, and second, the weights of individual filters should shift appropriately with each transition between D1 and D2.

The first condition is only fulfilled between the vertical dashed lines in Fig. 1(a), where the system operates in steady state. Before and after that region, startup and stopping dynamics dominate and would cause the divergence of the D1–D2 estimator. It is therefore necessary to verify that current dynamics are appropriately represented by available models prior to using these models for estimation.

The second condition is only fulfilled if transitions between D1 and D2 are slow enough to allow the individual filters to converge and compute correct weights. If individual filters compute incorrect weights, then inaccurate estimates are combined with accurate estimates, which reduces overall accuracy. This motivates the development of a system that assigns correct weights even if they are incorrectly computed by individual filters.

These problems are addressed with an estimation framework that first identifies the dynamics and then determines which filters to use. The approach introduces the notion of *context* and represents the dynamics with a hierarchy of contexts as follows: *dynamical* contexts represent the dynamics that are described by one model and *behavioral* contexts represent specific sequences and frequencies of transition among dynamical contexts. This definition explicitly ties dynamics to the models that

represent them, and therefore, identifying a robot's dynamical or behavioral context determines which model or set of models are appropriate for estimation.

For dynamical contexts, identification is achieved through the classification of sensor measurements into sets of previous measurements that correspond to different dynamic modes. The underlying principle is that locomotion dynamics strongly affect the signal of onboard sensors, therefore analyzing that signal provides information about the dynamics themselves. Assuming that distinct dynamics produce distinct sensor measurements, comparisons with previous measurements generated by known dynamics enables the identification of current dynamics and dynamical contexts.

For behavioral contexts, the identification approach is based on the observation that hybrid locomotion dynamics induce spatial and temporal structure in the signal generated by onboard sensors. Recognizing these structures leads to the identification of the dynamics and their associated behavioral contexts. This is a pattern recognition problem solved by using hidden Markov models (HMMs) and timed automata to recognize spatial and temporal structures, thereby identifying the behavioral context.

Context identification addresses the shortcomings of MM approaches for mobile-robot estimation. The scalability is improved by first identifying the behavioral context, which decreases the number of applicable MM estimators [see Fig. 2(a)], and, second, identifying the dynamical context, which decreases the number of individual filters activated within a single MM estimator [see Fig. 2(b)].

The accuracy is improved by identifying dynamical contexts at a bandwidth comparable with the sensors' update rate to capture abrupt transitions in the dynamics. Behavioral context identification further improves accuracy by preventing the use of inappropriate MM filters in situations where none of the available models accurately represent the dynamics (see Fig. 3).

III. RELATED WORK

Context-based state estimation enables accurate and scalable state estimation by combining discrete- and continuous-state estimation tools. This section provides an overview of these tools, analyzes the accuracy and scalability limitations of MM filters and compares the context-based estimation framework with other hybrid-estimation techniques.

A. Multiple-Model Estimation

Estimating the state of systems with multiple modes of operation typically involves associating hypotheses to mode transitions and evaluating the most likely hypothesis at every sampling step. Each hypothesis consists of a transition from a particular mode at the previous sampling step to a particular mode at the current step. A filter based on the model of the current mode is associated to each hypothesis, and its prior is the previous output of the filter based on the model of the preceding mode. The relative likelihoods of all filter outputs are computed at each sampling step, and the highest likelihood output is adopted as the system's state estimate.

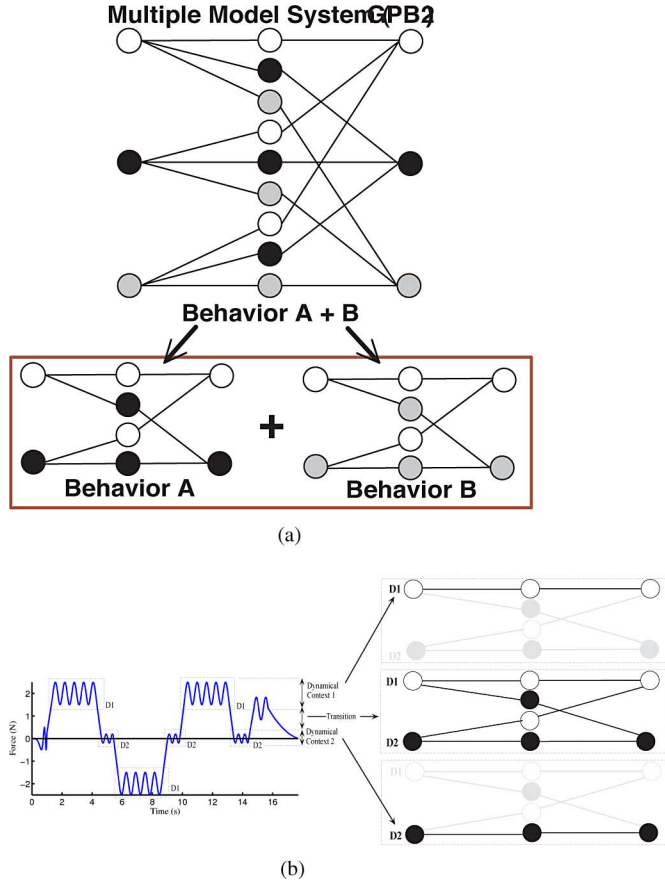


Fig. 2. Behavioral context identification also increases estimation scalability by replacing large-scale MM systems with a collection of small-scale systems. (a) By identifying the dynamical context, the estimation system activates only appropriate filters, which reduces computational cost. (b). (a) Improved scalability from behavioral context identification. (b) Improved scalability from dynamic context identification.

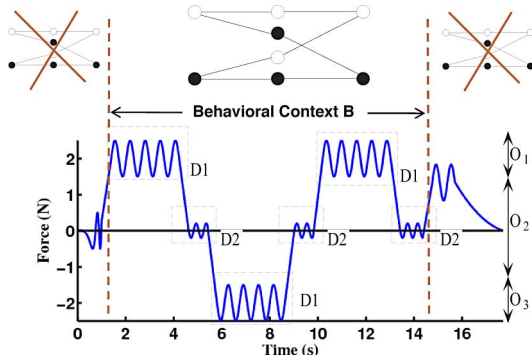


Fig. 3. Identification of a system's behavioral context enables the deployment of MM filters only when appropriate, which increases estimation accuracy.

Unfortunately, the number of hypotheses increases exponentially with the number of steps, which has motivated extensive research in seeking methods to reduce the number of hypotheses. The most accurate of these techniques is the generalized pseudo-Bayesian 2 (GPB2) algorithm [6], [17], [21] expressed in the KF framework, which collapses the hypotheses at each sampling step into a set whose cardinality is equal to the number of modes. This way, the number of hypotheses is maintained

Algorithm 1 Steps of a GPB2 update [6]

$$[x_{i,j}, P_{i,j}] = \text{filter}_i(x_j) \quad (1)$$

$$p_{i,j} = \frac{1}{\sqrt{2\pi S_{i,j}}} \exp\left(-\frac{1}{2} r_{i,j} S_{i,j}^{-1} r_{i,j}^T\right) \text{ (Likelihood)}$$

$$T_{i,j} = \text{Prob}(i_t | j_{t-1}) \text{ (Transition Probability)} \quad (2)$$

$$\text{Prob}_{i,j} = \frac{p_{i,j} T_{i,j} \text{Prob}_j}{\sum_j p_{i,j} T_{i,j} \text{Prob}_j} \text{ (Probability}(j_{t-1} | i_t)) \quad (3)$$

$$x_i = \sum_j x_{i,j} \text{Prob}_{i,j} \text{ (Mode State)} \quad (4)$$

$$P_i = \sum_j \text{Prob}_{i,j} \left(P_{i,j} + (x_{i,j} - x_i)(x_{i,j} - x_i)^T \right)$$

$$\text{Prob}_i = \frac{\sum_j p_{i,j} T_{i,j} \text{Prob}_j}{\sum_i \sum_j p_{i,j} T_{i,j} \text{Prob}_j} \text{ (Probability}(i_t)) \quad (5)$$

$$x = \sum_i x_i \text{Prob}_i \text{ (Consolidated State)} \quad (6)$$

$$P = \sum_i \text{Prob}_i \left(P_i + (x_i - x)(x_i - x)^T \right)$$

constant from one sample step to the next, which improves tractability.

The GPB2 estimation starts with evaluating hypotheses about the system being in one of its N dynamic modes at the previous sampling step and transitioning into one of the same N modes at the current step. An individual KF based on the model of the current mode is associated with each hypothesis, and therefore, the total number of active filters (and of hypotheses) is N^2 , which is a permutation among the N modes.

Described more formally, each GPB2 iteration starts with the assumption that any mode could have been in effect at time $t-1$, and any mode could be in effect at time t . For a system with N modes, a bank of N^2 filters is updated and the output of all filters is consolidated. Algorithm 1 details the steps involved in a cycle. Here, filter_i is based on the model of mode i ; i_t represents the hypothesis that mode i is in effect at time t ; $x_{i,j}$ is the state estimated by filter_i and whose prior is the output of filter_j ; r is the residual (innovation); and S and P are the innovation and process covariances, respectively.

A two-model GPB2 cycle is shown schematically in Fig. 1(b). The two initial hypotheses, which are represented with the first column of white and black circles, spawn four hypotheses and as many KFs (1) and (3), which are represented by the middle column of four circles. To prevent the exponential growth of the number of hypotheses, the four hypotheses are collapsed back to the original two [see the last column of two circles and (4) and (5)] [17]. At every iteration, a best state estimate can be extracted from the GPB2 by summing the output of all individual filters weighted by each filter's likelihood (6) [6].

The quadratic relationship between the number of modes and the number of filters, which is compounded with the computational overhead of Kalman filtering, limits the scalability of the GPB2. This makes the algorithm impractical for systems with a large number of modes and motivates the development of the context-based approach.

B. Hybrid State Estimation

Hybrid state estimation is an extension of MM estimation, where the discrete and continuous modes of the dynamics are simultaneously estimated. Since a separate model is typically associated with each discrete mode, this approach reduces the number of active filters when a discrete mode is identified or excluded.

Current approaches have a double limitation. First, they do not detect situations where the entire model set is inadequate to represent current dynamics and, therefore, can generate wrong estimates by mistakenly relying on inaccurate models. Second, convergence of the discrete estimators may be slower than the sensors' update rate, which limits their applicability to fast-switching dynamics.

Hofbaur and Williams [9] present a hybrid probabilistic approach, where a bank of KFs is used to estimate continuous state, and a probabilistic automata is used to estimate discrete state. This automata associates probabilities to the transitions between discrete modes, which captures some *a priori* knowledge to improve the performance of the algorithm. They also include an algorithm that dynamically selects certain hypotheses to prevent the exponential increase of hypotheses. However, any hierarchical dynamic structure is not explicitly encoded, and their techniques are not adapted for fast detection of rapidly switching discrete states.

A hierarchical hybrid approach is proposed by Verma [31] in the field of fault diagnosis, where continuous dynamics are used for robust detection of discrete states of failure. Verma's approach is based on particle filter estimation, where algorithms use hierarchical techniques to ensure that unlikely but important hypotheses are not pruned away. One algorithm considers the risk to ignore a hypothesis in addition to the probability that it is correct, while another approach uses *a priori* knowledge to segment the continuous state space at a variable resolution so that unlikely hypotheses represent samples in coarser regions. However, Verma's hierarchy tackles a reverse problem; rather than identifying discrete states to robustly estimate the continuous state, which uses continuous state estimation to inform the detection of certain discrete states.

Another approach to hybrid estimation applied to legged robots was proposed by Singh and Waldron [28] concurrently to this work [29]. They combine readings from multiple sensors to compute the Froude number, which is then used as criterion for mode selection. This approach benefits from the ability to detect sharp mode transitions and, therefore, switch modes at high bandwidth to enable accurate estimation for a quadruped robot. However, unlike the context-based framework, their technique does not detect situations where none of the available models are appropriate for estimation. Furthermore, the use of the Froude number as unique criterion for mode switching seems to limit the scope of their approach to quadruped locomotion.

Outside of robotics, the target-tracking community has made extensive contributions to MM estimation in general (e.g., [3], [11], [23], [24]) and hybrid estimation in particular [18]–[20]. Of direct relevance is the work on variable-structure MM state estimation [20], where Li *et al.* group models in mode-specific

model sets and propose techniques to activate and deactivate variable structure interacting MM (VSIMM) filters specific to each model set. Their approach organizes models in a hierarchy of modes to improve the scalability of MM estimation, which is similar to the context-based approach proposed here. It is worth noting that the VSIMM method to deactivate filters could serve to extend the context-based approach to situations where dynamical contexts do overlap.

However, the VSIMM technique is limited in two ways. First, model predictions and transition probabilities are used to decide when to activate new MM filters. This supposes that the existing models can represent all the dynamics, and therefore, it is not possible to identify situations where the dynamics cannot be represented by any of the available models. Such situations arise frequently for mobile robots, where unpredictable transients can dominate locomotion dynamics. Second, the limited sensor information available for target-tracking (such as aircraft position) constrains estimators to rely on model estimation alone to identify mode switching.

C. Classification

Statistical classification is a deterministic discrete-state estimation technique used in this research to efficiently identify dynamical contexts. Classification consists of grouping individual data points in classes based on training sets of previously labeled data points. Applied to robotics, classification is used to infer a robot's discrete state from the analysis of data generated by onboard sensors. An example is Lenser's nonparametric time-series approach, which consists of learning statistical correlations between pairs of consecutive sensor data points and a given environmental feature (e.g., type of terrain), and using these statistical models to classify current pairs of data points [16].

The advantage of classification is that the discrete state is estimated at the rate at which sensor information is made available to the system. In other words, state estimation can be performed at a bandwidth similar to that of the sensors update rate, which provides for timely identification of dynamical contexts. The disadvantage is that building classes can be complex and time consuming, which requires large training sets for accurate classification.

D. Hidden Markov Models

Behavioral context identification is performed in part by the HMMs and is used to recognize patterns in sensor data that correspond to specific contexts.

An HMM is a probabilistic graphical model that undergoes transitions among its N states and generates discrete observations¹ (see Fig. 4). State estimation is used for an HMM is preformed by computing a probability distribution $\alpha_k(i)$ over its states i with the *forward* algorithm [25], which expresses the

¹HMMs are generative models, as they describe processes that generates observations. These are the observations that the models predict and which, when compared with actual measured observation, allow the computation of a probability distribution over the HMMs' discrete states.

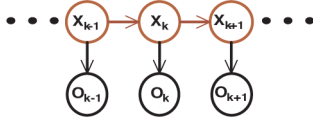


Fig. 4. Generic representation of HMMs that transition among discrete states (X) and generate discrete observations (O).

probability of the HMM process being in a state i at step k as

$$\alpha_k(j) = \frac{\left[\sum_{i=1}^N \alpha_{k-1}(i) a_{ij} \right] b_j(O_k)}{\sum_{j=1}^N \left\{ \left[\sum_{i=1}^N \alpha_{k-1}(i) a_{ij} \right] b_j(O_k) \right\}}$$

$$\sum_{i=1}^N \alpha_k(i) = 1 \quad (7)$$

where the transition probabilities a_{ij} express the probability of transitioning from state i at step $k-1$ to state j at step k , and the observation probabilities $b_j(O_k)$ express the probability that observation symbol O_k is generated by state j at step k . Equation (7) is initialized with $\alpha_1(i) = \pi_i b_i(O_1)$, where π_i is the initial probability of state i and $\sum_{i=1}^N \pi_i = 1$. Transition and observation probabilities can be tuned manually or learned from labeled data using algorithms such as Baum–Welch [25].

The probabilistic expression of HMMs provides robustness to sensor noise, as current probability distributions mix observation probabilities $b_j(O_k)$ with model predictions $\sum_{i=1}^N \alpha_{k-1}(i) a_{ij}$. This property is used for robust recognition of a signal's spatial structure, as shown in the next sections, but not of its temporal structure, because HMMs do not explicitly model dwelling time in a state and cannot track a history of transitions among states. Capturing time could be performed by semi-Markov processes (SMPs), but SMPs have a high computational cost and are impractical for estimation [5], [32]. An alternative solution is to use timed automata.

The HMMs are extensively used in the robotics literature, including in humanoid robotics where robot programming by demonstration shares similarities with the approaches taken here. Lee *et al.* explore the use of multiple layers of HMMs to recognize human behavior and correspondingly modify a robot control trajectory for appropriate interactions [14]. Kulić *et al.* use an extension of HMMs, i.e., factorial HMMs, and focus on the clustering and recognition of demonstrated motion primitives and dynamically sizing the number of HMM states [13]. Perhaps, the most similar work is that of Aarno and Kragic, who take an explicit hierarchical, layered HMM approach to gesture recognition [1]. However, in all of these approaches, the HMMs are designed to recognize discrete gestures and are not to improve the choice among and speed of filters for low-level continuous state estimation. Furthermore, these systems do not explicitly use their hierarchy to improve the accuracy and scalability of state estimation for cyclical dynamics.

E. Finite-State Automata

A finite-state automaton (FSA) is a deterministic graphical model that undergoes input-triggered transitions among its

states and generates discrete observations [12]. The context-based framework uses automata as a complement to the HMMs, tracking transitions among HMM states and capturing in-state dwelling time. The automaton's estimation mechanism will be made clear in the example of Section VI.

This brief overview of related research highlights the accuracy and scalability limitations of MM filtering and presents discrete-state estimation techniques used to overcome these limitations.

IV. CONTRIBUTIONS

This research's main contribution is to enable accurate and scalable hybrid state estimation through hierarchical representation and classification of cyclical intermittent dynamics. Conventionally, classification and HMMs are used for speech recognition [15] and visual object identification [30], and MM estimation is used for aircraft fault detection and radar tracking [4], [24]. The context-based framework applies these techniques to robotic systems with intermittent and cyclical dynamics.

The framework uses statistical classification to identify a robot's dynamical context at high bandwidth. As a proof of concept, this research implements a simplified form of classification, where classes are defined manually from clusters of sensor data that correspond to dynamical contexts. This enables the explicit encoding of designer knowledge of the dynamics into the classification, as will become apparent in Section V, which reduces the need for large training sets. Experimental results demonstrate that classification enables the successful identification of dynamical contexts even when hybrid systems abruptly switch from one to another. These classification techniques also provide a framework to exploit rich sets of information from onboard sensors to identify mode switching. This complements model estimation as a switching criteria and, crucially for mobile robotics, enables high-bandwidth mode identification.

The approach for behavioral context identification combines HMMs and timed automata to capture both the spatial and temporal structure of patterns in sensor data. Experiments also show that this approach enables accurate identification of behavioral contexts.

The context-based approach extends existing hybrid estimation work by explicitly encoding the hierarchical structure of cyclical dynamics, thereby capturing not only single-mode transitions but higher level behavioral transitions as well. For instance, the context-based approach differentiates between walking and running (high-level behavioral contexts) by analyzing flight and stance dynamics (low-level dynamical contexts), as well as the patterns of transitions between them. This enables the activation of a reduced number of concurrent filters and the detection of situations, where neither walking nor running are occurring. Furthermore, unlike automata-only discrete-state estimation, the classification approach adopted for dynamical context identification provides for fast detection of discrete states. Such combination of fast convergence and hierarchical estimation is necessary for accurate and scalable state estimation for systems with fast-switching cyclical dynamics.

V. DYNAMICAL CONTEXTS

Classification enables the identification of dynamical contexts at a bandwidth close to the sensors' update rate. The approach calls to construct statistical models that map sensor measurements to the dynamics that induced these measurements and hence to their corresponding model. Such statistical models consist of sets in the sensor space,² each formed by clustering measurements generated while the system operates in one of its dynamical contexts. Clustering can be performed by operating that system in a controlled environment, where the dynamics are steady and can be appropriately represented by their corresponding motion models. Measurements generated by onboard sensors are then clustered into sets labeled after the different dynamical contexts. This way, each set of measurements corresponds to a unique dynamical context whose dynamics are appropriately represented by an available model. With this setup, dynamical contexts are identified whenever measurements can be classified in one of the sets. Constructing the classes is done offline and may require some effort, but the classification itself is computationally inexpensive and can be performed in real time.

The strategy is clarified with the help of the generic system presented in Section II. Fig. 1(a) shows that dynamical context 1 corresponds to force measurements that oscillate between 1 and 3 N, and context 2 measurements oscillate between -1.5 and -3 N. This observation suggests that the system's dynamical contexts can be identified by classifying current measurements in one of these sets; e.g., a single measurement of force equal to 2 N enables the immediate identification of context 1. In addition to the two contexts, the figure illustrates an intermediary transition context, which expresses the lack of appropriate motion models to represent the transient dynamics.

A. Impact on MM Accuracy and Scalability

Dynamical context identification helps MM filters assign appropriate weights to individual estimates even when the weights are computed incorrectly by the KFs. This improves the accuracy of the consolidated state estimate and can improve the scalability of MM systems by only activating appropriate KFs.

Assigning appropriate weights to individual filters can be done by modifying the parameters of the MM algorithm online. For the GPB2 described in Algorithm 1, (5) shows that the weights $Prob_{i,j}$ are a function of the transition probabilities $T_{i,j}$, the output likelihoods $p_{i,j}$, and the previous weights $Prob_j$. Of these terms, only $T_{i,j}$ is a parameter set by the designer; it can, therefore, be modified to affect the desired changes in $Prob_{i,j}$. If the system operation is classified as being in context i , then model i and its corresponding filter are known to be appropriate and the other filters inappropriate, as different models are assumed to describe distinct dynamics. Thus, assigning nonzero weights to the other filters incorporates knowingly inaccurate estimates to the consolidated output, which decreases its accuracy. The problem is avoided by setting the transition probabilities from any filter to filter i equal to one and to all

Algorithm 2 GPB2 Modifications. T and F stand for true and false, respectively.

<pre> 1. for $i \in \{D1, D2\}$, $j = \bar{i}$ { 2. if (identified mode = i) { 3. $T_{i,j} = T_{i,i} = 1$ 4. $T_{j,i} = T_{j,j} = 0$ 5. $x = x_i = x_{ii}$ 6. $flag_i = T$, $flag_j = F$ 7. } 8. if(no mode identified){ 9. for $i \in \{D1, D2\}$, $j = \bar{i}${ 10. if($flag_i = F$){ 11. $x_i = x_{jj}$ 12. $P_i = P_{jj}$ 13. } 14. } 15. } 16. $flag_{D1} = flag_{D2} = T$ 17. if($\dot{z} < 0$){$i = D2$; $j = D1$} 18. else{$i = D1$; $j = D2$} 19. $T_{i,j} = 0.7$; $T_{j,j} = 0.3$ 20. $T_{i,i} = 1$; $T_{j,i} = 0$ </pre>	<pre> 17. for $i \in \{D1, D2\}$, $j = \bar{i}${ 18. if($flag_i$){ 19. update $filter_i(x_i)$ 20. } 21. if($flag_j$){ 22. update $filter_j(x_j)$ 23. } 24. if($flag_i$){ 25. update $filter_j(x_i)$ 26. } 27. } </pre>
---	--

Execute Algorithm 1
Iterate

other filters equal to zero. This leads to the maximal weight of one being assigned to the output of filter i , and exactly zero to all other filters. Thus, by modifying $T_{i,j}$ based on which dynamical context is identified, the GPB2 is essentially reduced to a single-filter system, where the filter corresponds to the current dynamical context.

More formally, dynamical context information is incorporated into the GPB2 by changing the transition probabilities ($T_{i,j}$) in Algorithm 1 as a function of the mode. If mode i corresponds to the identified context and mode j represents all other modes, then set $T_{i,j} = T_{i,i} = 1$ and $T_{j,i} = T_{j,j} = 0$. This means that transitioning into the identified mode and staying in it has a probability of one, and transitioning into a wrong mode and staying in it has a probability of zero. As expected, this produces $Prob_i = 1$ and $Prob_j = 0$.

As a consequence of this strategy, the best state estimate generated by the GPB2 is equal to the state estimate of the accurate individual filter, i.e., $x = x_i = x_{i,i}$. This formalizes the observation that once the system is in the identified mode, it is expected to remain in it until a change of context is detected. In other words, the only valid hypothesis is (i_t, i_{t-1}) , and the output of the filter corresponding to mode i constitutes the sole output of the GPB2. By ignoring the contribution of inaccurate mode states, the accuracy of the consolidated state is not decreased unnecessarily.

Significantly, ignored filters do not have to be activated as they no longer impact state estimates. This effectively reduces MM filters to a single-model filter. Without dynamical context identification, MM filters need to run the entire bank of filters in order to generate combined estimates. Here, when the dynamics change, the contexts switch accordingly, which triggers the activation of a new filter and deactivation of an old one. This

²A sensor space is the space of all possible measurements.

reduces the computational overhead of MM filters and enhances their scalability.

In situations where the dynamical context cannot be identified, such as during transitions between dynamical contexts, the transition probabilities are restored to their nominal value, all filters are reactivated, and the GPB2 resumes normal operation.

Algorithm 2 gives a step-by-step description of the procedure for the D1-D2 system. Line 2 of the algorithm corresponds to situations where the dynamical context and, hence, the mode, is identified. Transitions to that mode are set to one, and transitions out of the mode to zero. The GPB2 output is now strictly the output x_{ii} of the individual $filter_i(x_i)$ corresponding to the identified mode.

Line 7 corresponds to the situation where the dynamical mode cannot be identified. The individual filter variables (states and covariances) are reset to the last estimates to properly initialize the nominal operation of the GPB2. As for transition probabilities, they are now a function of the state of the system to express the knowledge that during transitions, the system in D1 is more likely to switch to D2 than to stay in D1, and *vice versa*. Setting transition probabilities as a function of state improves the accuracy of the GPB2, as it encodes information about the dynamics that the GPB2 algorithm would be unable to capture from its model set.

Lines 17 through 25 are the statements that perform the selective activation of the appropriate filters.

VI. BEHAVIORAL CONTEXTS

Context-based state estimation applies principally to hybrid systems with cyclical dynamics, as the repetitive nature of such dynamics induce spatial and temporal structure in sensor signal, which can be exploited to identify the behavioral context. The spatial and temporal components are defined as follows: *Spatial* structure is the sequence of transition among salient points, or symbols, in a data stream, and *temporal* structure is the rate of transition among these symbols. The underlying assumption here is that different dynamics induce distinct structures, so recognizing these structures uniquely identifies the dynamics and the behavioral context.

The description of the identification approach is carried step-by-step in the following sections with the help of the generic hybrid system introduced earlier.

A. Hidden Markov Models

Consider the sensor output profile of Fig. 3(a), and recall that the behavioral context B corresponds to the steady-state region, where D1–D2 filters are appropriate for estimation. The first step to identify this context is to discretize the continuous force measurements into symbols that appear recurrently when the system is in behavioral context B. As shown in the figure, sensor symbols can be O_1 , O_2 , and O_3 , roughly corresponding to the top, middle, and bottom regions of the sensor output profile, respectively.

The second step is to build a Markov chain model of a process that could generate these sensor symbols. A first model may contain the states H , $M1$, L , and $M2$, which generates the sym-

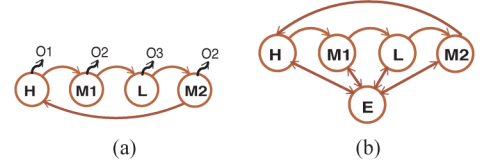


Fig. 5. HMM describing discrete process of a simple system. Self-transitions are omitted here for clarity but are discussed in more detail in Section VII-C.

bols O_1 , O_2 , O_3 , and O_2 , respectively [see Fig. 5(a)]. H and L correspond to high and low forces, $M1$ corresponds to medium forces that result from transitioning from high to low forces, and $M2$ corresponds to low-to-high transitions. Obviously, the same symbol O_2 is now generated by two different states $M1$ and $M2$, but the two states can be disambiguated by tracking the sequence of sensor symbols over time. If O_2 appears after O_1 , then the system is in $M1$, and if it appears after O_3 , then it is in $M2$.

The third step is to specify HMM parameters as per Section III-D, with the transition and observation probabilities a_{ij} and $b_j(O_k)$ designed to predict observation symbols in the expected sequence. For example, $a_{M2|L} = 1$, $a_{M2|j} = 0 \forall j \neq L$, $b_{M1}(O_2) = b_{M2}(O_2) = 1/2$, and $b_H(O_2) = b_L(O_2) = 0$. This enables the HMM to run the forward algorithm (7), processes the symbols O_k extracted from data discretization, and infers the probability distribution α over the states.

When the system is operating in steady state, the observed symbol sequence is expected to match the sequence predicted by the model. Likewise, the sequence of the HMM states³ is expected to match the state transitions described by the model. When the dynamics vary from steady state, the sequence of symbols also varies from the predictions and leads to out-of-order state sequences. Therefore, verifying the order-of-state sequences helps recognize the behavioral context.

In order to enable the explicit detection of out-of-order transitions, the HMM model is augmented with an error state E , as shown in Fig. 5(b). The error state has low-probability two-way transitions to all states and all observations have a uniform distribution over it, i.e., the error state is equally likely to generate all sensor symbols. The observation probabilities are designed such that the probability to observe a symbol conditioned on a state that should not generate it is lower than the symbol's probability conditioned on the error state. This means that the likelihood to generate a specific symbol by the error state is greater than the probability to generate that same symbol by a state that should not generate it. For example, $b_E(O_3) > b_H(O_3)$, and more generally, $b_E(O_k) > b_j(O_k)$ if O_k is different from the symbol generated by state j , where j is an index among the states.

This property ensures that if the wrong sequence of sensor symbols are observed, then the HMM would transition to the error state. For example, assume the model predicted that the

³In this paper, the sequence of the HMM states is defined as the sequence of most likely states estimates by (7). At each step, the most likely state is $\arg\max_{1 \leq i \leq N} (\alpha_i)$.

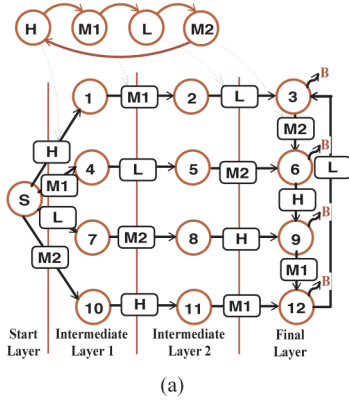


Fig. 6. HMM output (top) such as H , $M1$, L , and $M2$ serve as input (rectangles) that trigger transitions between the FSA states (circles). The structure of the FSA (bottom) is organized in p layers in order track input sequences over p steps. The output flag “B” indicates that behavioral context B is identified. Reset transitions back to S are omitted for clarity. (a) HMM output states serve as FSA input.

system would transition to state j , but the observed symbol cannot be generated by j . This is an indication that the state sequence is out of order, and therefore, the observation probabilities ensure that $\alpha_E > \alpha_j$. In other words, the HMM assigns the highest likelihood to the error state E when the sequence of states is out of order. Thus, whenever the system is in E , it is not expected to be in the behavioral context that corresponds to the HMM model (context B in this case).

B. Finite-State Automata

Detecting a single out-of-order transition is sufficient to recognize that the system is not operating in the expected behavioral context. However, contexts cannot be positively identified from observing a single in-order transition; a minimum number of transitions is necessary to avoid false positives.

Unfortunately, the HMM cannot track a sequence of transitions for longer than one time step because of the Markov assumption (the current state contains all the information about a system, and therefore, the future only depends on this single time step). This motivates the use of FSA to perform book-keeping. Here, HMM states are treated as inputs that trigger transitions among FSA states (see Fig. 6). The FSA state structure is organized in p layers, which is designed to recognize a correct sequence of inputs over p steps. Layers are defined as follows: The first layer contains the starting state; the final layer contains all the states reached after p numbers of correct transitions; and intermediate layers contain states reached after n correct transitions, with $n < p$.

To see how this FSA can identify p successful transitions, assume that the initial input event transitions the automaton from the starting state S to a target state in the first intermediate layer. Consecutive occurrences of the same input cause self-transitions, but new in-order inputs induce transitions to the next layer. After p correct transitions, the automaton reaches the final layer and outputs a success flag, which identifies the behavioral context (“B” in this case), and new in-order inputs maintain the automaton in the final layer. At any time, out-of-

order inputs reset the automaton, and the behavioral context is no longer identified. In summary, success flags indicate that the signal’s spatial structure is recognized.

C. Timed Automata

The FSA model designed to recognize the spatial structure of the signal can also be used to recognize the temporal structure. The temporal analysis mechanism turns the FSA into a timed automaton that measures delays separating consecutive inputs. Delays are measured by a clock reset each time the automaton enters a new state. Since HMM states correspond to automaton inputs, the automaton simultaneously measures its own state duration as well as the HMM’s.

Temporal information can be used as a timeout that triggers a transition out of a state if the duration exceeds a predefined bound. It also enables time-sensitive transitions, whereby the target state is selected as a function of both the input and the duration in the previous state.

VII. CONTEXT IDENTIFICATION EXPERIMENTS

Validation experiments are conducted on RHex [26], i.e., a six-legged dynamic robot able to walk and jog, among other behaviors. These experiments demonstrate that the classification and the HMM-timed automaton approaches are successful to identify dynamical and behavioral contexts and that contextual information improves the accuracy and scalability of the MM estimators.

The task at hand is to automatically identify RHex’s behavior and to estimate the robot’s height while jogging on a horizontal surface using low-cost onboard accelerometers. The hierarchical description of the dynamics in the previous sections leads to the natural segmentation of RHex’s dynamics into behavioral contexts (walking, jogging, or neither) and dynamical contexts (flight phase of a jog, stance phase, or neither). This segmentation applies to a class of dynamic robots, as represented in Fig. 7, and provides a two-tier improvement of accuracy and scalability. First, behavioral context identification reduces the size of individual MM systems and ensures that such systems are only used when appropriate. Second, dynamical context identification activates only the individual filters that are appropriate for the current dynamics and assigns correct weights even when individual filters compute them incorrectly.

A. Dynamical Analysis of RHex’s Jogging Behavior

RHex’s height is estimated from acceleration measurements generated while jogging, using GPB2 filters and jogging motion models.

When jogging, RHex synchronizes its legs three by three to generate a stable alternating tripod gait. The jogging behavior alternates flight and stance phases akin animal running, which leads to oscillate accelerations, as shown in Fig. 8 (steady dynamics). Flight dynamics are those of a ballistic projectile, where only gravity acts on the body, and are responsible for the negative accelerations close to g (-9.8 m/s^2). Positive accelerations are due to the stance dynamics, where the body is

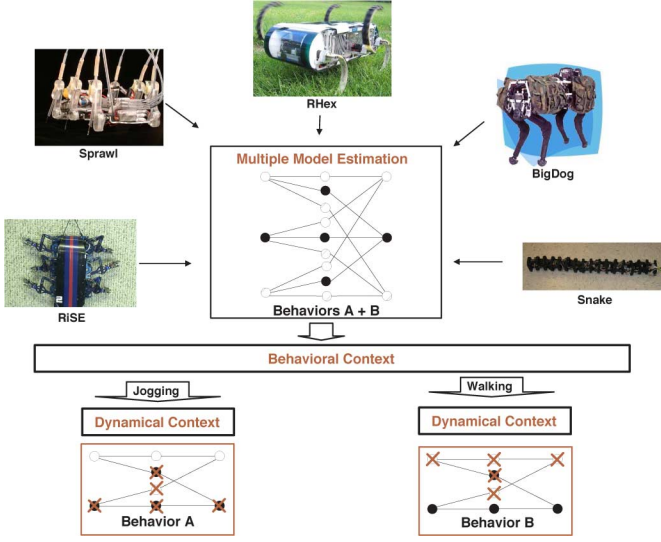


Fig. 7. Context-based estimation applies to systems such as mobile robots that have recurrent and rapidly switching dynamics. Examples of recurrent dynamics are hyper-redundant robots, which are also known as snake robots [7], [8], [27]. Examples of rapidly switching dynamics span a spectrum of legged robots, such as Sprawl [10], which is a small-scale robot for horizontal running; RHex [26], which is a medium-scale robot for horizontal locomotion; Big Dog, which is a large-scale robot also for horizontal locomotion; and RiSE [2], which is a medium-scale robot for vertical locomotion. For all these systems, identifying the behavioral context specifies which small-scale MM system to activate, and identifying the dynamical context determines which filter within the small-scale MM to activate.

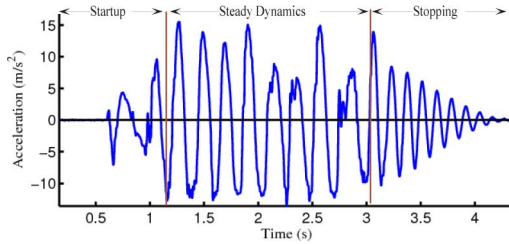


Fig. 8. Output of vertical accelerometer while RHex executes jogging behavior.

subject to the ground's reaction force. The motion models used for the flight and stance phases are the ballistic projectile and the mass-spring system, respectively, as shown in Fig. 9.

At the end of the experiment, the motors stop and the robot bounces on all six legs until it comes to rest. This induces damped oscillations of the stopping dynamics referred to as the *stand* phase, which are represented with a mass-spring-damper model. In summary, the jogging behavior can be modeled with a collection of three models:

$$\ddot{z} = \begin{cases} -g, & \text{flight} \\ -K_z(z - z_0)/M - g, & \text{stance} \\ -2K_z(z - z_0)/M - g - (D/M)\dot{z}, & \text{stand} \end{cases}$$

where z is the state that represents the height of the robot; K_z is the virtual spring constant of the mass-spring system; z_0 is the robot's height at rest; M is the robot mass; D is a viscous damping parameter; and g is a gravity.

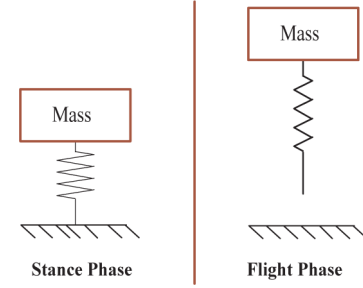


Fig. 9. RHex's vertical motion can be modeled as a virtual mass-spring system. During stance, the spring compresses and causes the mass to rebound. In flight, the virtual spring detaches from the ground, and the mass describes a ballistic trajectory.

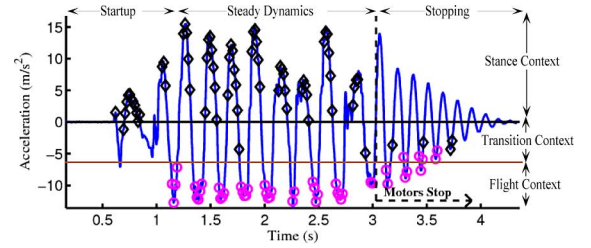


Fig. 10. Output of leg strain gauges (circles, diamonds, and stars) is overlaid on the vertical oscillatory accelerometer data. The output of strain gauges is used to define classes that correspond to dynamical contexts.

These models enable the estimation of RHex's height using a three-mode GPB2 system. However, since the dynamics change rapidly, each of the model is only appropriate over a limited period of time. Therefore, the GPB2 needs to determine precisely when each of the models is appropriate to generate correct height estimates. This is achieved through the identification of the robot's dynamical context.

B. Dynamical Context Identification

RHex's jogging behavior has three dynamical contexts: flight, stance, and stand. These contexts can be identified by constructing sets of acceleration measurements that correspond to each of the dynamic modes. These sets are labeled as flight, stance, or stand, depending on whether the data points were generated while in the flight, stance, or stand contexts. This can be done in a laboratory setting where the context can be identified with specialized sensors used to label sets of accelerometer data points. In this case, the specialized sensors consist of strain gauges attached to the legs to measure their deflection; three compressed legs indicate stance (tripod configuration), extended legs indicate flight, and six compressed legs indicate stand. Fig. 10 overlays the output of these strain gauges on the acceleration plot. A visual inspection of the steady-state portion of the plot shows that, as expected, and with few exceptions, the flight context is detected when the acceleration approaches gravity, and the stance context is detected when the acceleration is positive.

Since classification consists essentially of a straightforward comparison between current data and previously constructed classes, the robot's dynamical context can be identified at the rate at which the data are made available by the sensors.

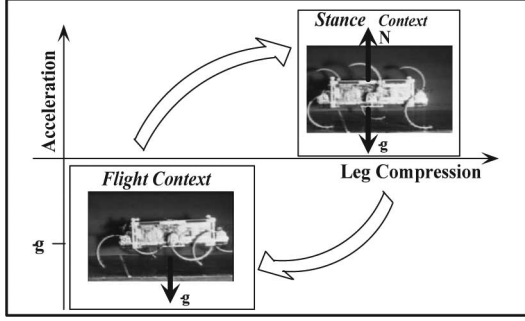


Fig. 11. Jogging hexapod robot alternates flight and stance dynamical contexts. The contexts are identified by classification of leg compression and body-acceleration measurements.

It is possible to simplify the estimation problem by considering only steady-state dynamics, which is made possible by the use of behavioral contexts, as demonstrated in the next section. This reduces the system to a two-mode GPB2 based on the flight and stance models. The simplified setup enables dynamical context identification through the classification of acceleration measurements alone, with no need for specialized sensing.

For instance, inspection of Fig. 10 shows that the flight and stance dynamical contexts can be defined as follows:

- 1) flight context: $\ddot{z} < -6 \text{ m/s}^2$;
- 2) stance context: $\ddot{z} > 0$;

where the vertical acceleration $\ddot{z}$ is measured by onboard accelerometers. This approach is captured pictorially in Fig. 11.

Thus, accelerometer output can be immediately classified into one of the two dynamical contexts, and consequently a corresponding two-mode MM system that estimates RHex's height can selectively activate the flight or stance filters. This is in contrast with conventional MM systems that run all filters simultaneously and assign probabilistic weights to their output. In this example, accelerations between -6 and 0 m/s^2 indicate that the robot is transitioning between flight and stance, and therefore, the context cannot be determined with certainty. In these situations, the MM system activates all filters based on both models and resumes conventional operation.

It is worth noting that the strain gauge setup is complex and onerous to install on robots, and its use is generally restricted to the laboratory. Therefore, strain gauges can be used to establish the classes, but not to identify the dynamical context when the robot is deployed in the field.

This approach to dynamical context identification suffers from two shortcomings. First, classification is susceptible to sensor noise, which can force a data point to fail to identify a context and be classified in the transition region. Fortunately, the consequence is simply that all MM filters get activated, thus reverting back to the nominal operation of the MM system. This leads to suboptimal estimates but still constitutes an acceptable solution over short periods of time.

The second limitation is that this approach requires different dynamics to be represented by distinct classes, which is not always the case. One approach to disambiguate potentially overlapping classes is to perform the classification task over a history

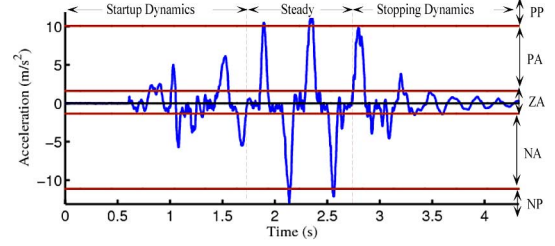


Fig. 12. Output of lateral accelerometer. The jogging gait induces positive and negative amplitudes during left- and right-tripod stances and periods of near-zero accelerations during flight.

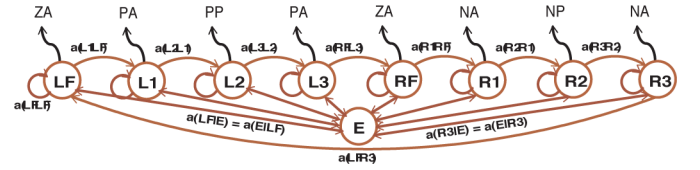


Fig. 13. Markov-chain model of the process that generates jogging symbols. The error state is added to capture out-of-order state transitions.

of measurements rather than just with current data points. This is precisely the technique used to identify behavioral contexts; therefore, behavioral context identification can also be used to recognize dynamical contexts.

C. Behavioral Context Identification

RHex's lateral acceleration while jogging is shown in Fig. 12. Available jogging motion models are only appropriate over the steady-state region indicated in the figure; therefore, the task is to identify the robot's jogging behavioral context in order to use jogging filters only when appropriate.

To this end, the acceleration data are first discretized, as shown in the figure. The observation symbols are positive (PA), negative (NA), and zero (ZA) accelerations and positive (PP) and negative (NP) peaks. The PA corresponds to accelerations larger than 1 m/s^2 , NA corresponds to accelerations smaller than -1 m/s^2 , and ZA spans the range in between.

A Markov chain model of the process that generates these symbols is presented in Fig. 13. To accurately identify the jogging behavioral context, it is important to verify that jogging accelerations reach the expected amplitude; therefore, the right and left tripods are represented by three states: The first state (R1 or L1) corresponds to the ascending acceleration, the second state (R2 or L2) corresponds to the peak acceleration, and the third state (R3 or L3) corresponds to the descending acceleration. Thus, R1, R2, and R3 (or L1, L2, and L3) generate the symbols PA, PP, and PA (or NA, NP, and NA), respectively. LF corresponds to the flight phase during transitions from left to right tripods, and RF corresponds to the flight phase during transitions from right to left, and E is the error state.

Transition probabilities are determined as follows. Self-transitions a_{ii} to state i are computed by counting the number $\bar{s}$ of symbols that occur when the system is in each state. The expected number of observations in a state, conditioned on starting in that state, is $\bar{s} = \sum_{s=1}^{\infty} sp(s)$, where p is the probability

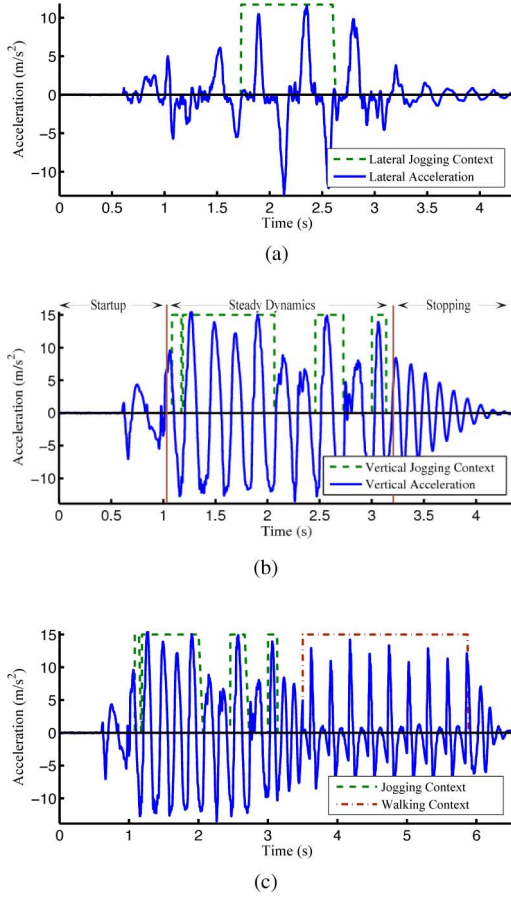


Fig. 14. Behavioral contexts identified along multiple dimensions and over multiple behaviors. (a) Lateral jogging behavioral context. (b) Vertical jogging behavioral context. (c) Jogging and walking behavioral contexts.

to undergo s self-transitions to the same state. This equation can be rewritten as $\bar{s} = d(a_{ii})^{d-1}(1 - a_{ii}) = 1/(1 - a_{ii})$ [25]; therefore, $a_{ii} = 1 - 1/\bar{s}$. For example, $R2$ generates on average 22 PP symbols; therefore, $a_{R2|R2} = 1 - 1/22 = 0.9432$.

The transition probability from one state to the next is set equal to the transition probability from the same state to E to ensure that the HMM detects out-of-order transitions. The probabilities for E are set as $a_{E|E} = 0.999$, and $a_{E|i} = (1 - a_{E|E})/N$, where $N = 8$, the number of states excluding E , and $i \in N$. The observation are set following the example of Section VI-A, and the probability distribution α is computed recursively with (7).

A timed automaton similar to Fig. 6 tracks the sequence of most likely HMM states over three layers, measures the in-state dwelling time, and outputs jogging flags when the jogging behavioral context is identified. The results of Fig. 14(a) show that the approach successfully recognizes the steady-state region over which jogging models can be used for estimation.

Identifying behavioral contexts along different dimensions is necessary when distinct motion models are used along each dimension. In this case, the same HMM-automaton technique successfully identifies behavioral contexts from acceleration measurements along RHex's *vertical* axis, as shown in Fig. 14(b). The technique is shown in Fig. 14(c) to also enable accurate

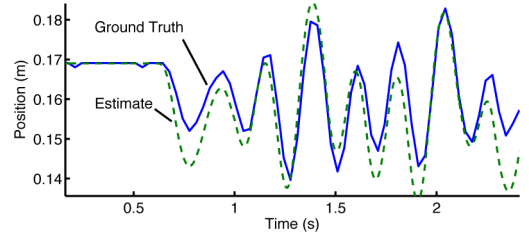


Fig. 15. GPB2 based on ballistic flight and mass-spring stance models generates accurate state estimates. Over 12 experiments, rms errors have a mean 1.85 cm and a variance of 0.37 cm.

behavioral context identification for RHex's walking behavior, as well as for other behaviors that transition between walking and jogging. The variety of dynamical situations in which the technique is successfully applied provide empirical evidence of the approach's robustness for context identification.

VIII. CONTEXT-BASED STATE ESTIMATION

Successful identification of dynamical and behavioral contexts enable accurate and scalable MM estimation. The following set of experiments use the results of the previous section to demonstrate that context information provides significant performance gains to estimate the state of RHex.

A. Height Estimation in Steady State

The first experiment demonstrates that using the simple models of Section VII-A under steady-state conditions produces accurate height estimated for RHex when it jogs.

Given that the steady-state portion of the jogging dynamics can be described by the flight and stance models, RHex's height is estimated using a two-mode GPB2 (see Algorithm 1). Each iteration of the GPB2 starts with the hypothesis of the robot being in the flight or in the stance modes at the end of the previous iteration. Each hypothesis then transitions into one of the two modes at the current iteration, which leads to a total of four hypotheses and four KFs.

Each of the KFs outputs individual estimates of height $z_{i,j}$ and likelihoods $p_{i,j}$. Next, a probability for each hypothesis is computed and used to scale the individual estimates $z_{i,i}$ and consolidate them into the intermediary mode-specific estimates z_i . This step is commonly referred to as hypothesis collapsing, as it reduces the number of hypotheses back to the original number of two.

As the GPB2 is running through the cycle of expanding and collapsing hypotheses, it is possible to extract a unique state estimate at any time. This is simply done by consolidating the intermediate mode-specific estimates z_i weighted by their hypothesis probability into the system's best estimate z .

This approach leads to satisfactory results, with height estimates closely matching ground-truth measurements,⁴ as can be visually verified in Fig. 15. The mean and standard

⁴The ground-truth measurement system consists of high-speed cameras that register the position of LEDs placed on the robot body. The cameras cover a limited surface area, which explains the short experimental runs (about 4 s for jogging). Nevertheless, these results are useful because the available space is

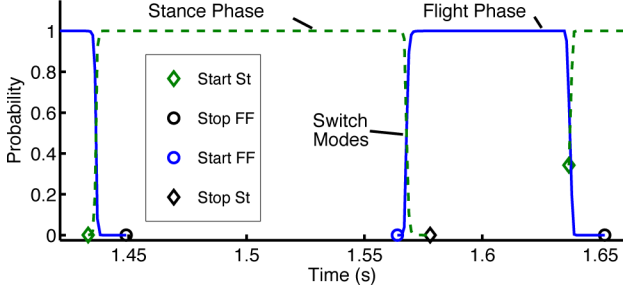


Fig. 16. Plots of $Prob_i$ for RHex while jogging using the modified GPB2 algorithm. When the dynamical context is identified, only the accurate filter is activated, which reduces the computational cost.

deviation of the rms difference between the estimated height and the ground-truth measurement over 12 experiments is 1.85 and 0.37 cm, respectively. The mass-spring parameters of the stance model are set to the actual mass of the robot (8.5 Kg), and the spring constant that yields acceptable state estimates is found to be 6800 N-m, which is a value close to the physical spring constant estimated at 6600 N-m.

B. Dynamical Context Identification Reduces the Number of Active Filters

This experiment shows that identifying the dynamical context improves the scalability of MM filtering. The dynamical contexts identified in Section VII-B are used to modify the GPB2 of the previous experiment such that only appropriate filters are activated.

Algorithm 2 describes how the GPB2 algorithm is modified to incorporate contextual information. For RHex, the transition probabilities are set as a function of the state of the robot as follows. If RHex's velocity is negative, the descending robot is expected to touch down; therefore, the transition probabilities are biased toward transitioning into stance.⁵ Conversely, positive velocities bias transitions to flight. Setting transition probabilities as a function of state improves the accuracy of the GPB2, as it encodes information about the dynamics that the GPB2 algorithm is unable to capture from its model set.

The identification of the dynamical context is performed using strain gauges. When the robot is determined to be in flight or in stance, the modified GPB2 only activates the flight or the stance filter. However, when RHex is about to touch down or lift off, the strain gauges are unable to identify the mode; therefore, all filters are activated, and the GPB2 resumes nominal operation.

Since the conventional GPB2 algorithm is able to accurately estimate the height of RHex while jogging at steady state, the modified algorithm does not noticeably increase its accuracy. However, the results of the modified algorithm indicate improvement in scalability, as evidenced by Fig. 16. The plot of mode probabilities shows that when the dynamical context is identified, only the accurate filter is activated. Conversely, during

large enough to allow the robot to achieve steady-state motion before exiting the cameras' field of view.

⁵The specific values of $T_{i,j} = 0.7$ and $T_{j,i} = 0.3$ are determined empirically.

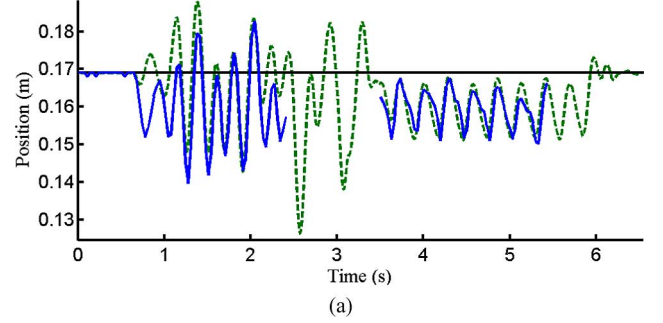


Fig. 17. Identification of the dynamical context enables the selective deployment of a jogging GPB2 and of a walking KF. The resulting height estimates over the combination of jogging and running behaviors have a similar accuracy as when the two behaviors are executed separately. (a) Context-based height estimates.

transitions from one dynamical context to another, the nominal GPB2 is operated with all filters active.

An interesting exercise is to try to identify RHex's dynamical context with acceleration measurements alone. State estimates obtained with this contextual classification are virtually indistinguishable from the previous results. The mean and standard deviation of the rms error over 12 experiments are 2.09 and 0.39 cm, respectively, which compare favorably with the values obtained with the help of strain gauges.

C. Behavioral Context Identification Allows Reduced-Scale MM Systems

Behavioral context are shown to improve the estimation scalability of multiple-behavior systems. Here, the task consists of estimating RHex's height as it jogs *and* walks.

The acceleration profile for this two-behavior experiment is shown in Fig. 14(c); accelerations up to 3.5 s are induced by jogging dynamics and by walking dynamics thereafter.

If behavioral information were not available, the experiment would require a three-model conventional MM system; two models for jogging, as described in the previous sections, and one for walking. An appropriate walking model is defined as follows:

$$\ddot{z} = \begin{cases} -K_{z,2}(z - z_{0,2})/M - (D_2/M)\dot{z} - g, & \text{2-tripod} \\ -K_{z,1}(z - z_{0,1})/M - (D_1/M)\dot{z} - g, & \text{1-tripod} \end{cases}$$

where $K_{z,1}$, D_1 , $z_{0,1}$ and $K_{z,2}$, D_2 , $z_{0,2}$ are the virtual spring constant, damping coefficients, and rest lengths for one tripod in stance and two tripods in stance, respectively. Unlike the jogging behavior, walking uses a single KF, whose model is modified by changing its parameters, depending on the tripod.

Information about the behavioral context is available, and the results of Section VII-C can be directly used here; when the jogging context is identified, then the flight-stance GPB2 of the jogging behavior is used; when walking is identified, the single KF of the walking behavior is used.

The result of this context-based state estimation are shown in Fig. 17. The accuracy of the estimates and the computational complexity of the filtering system are consistent with

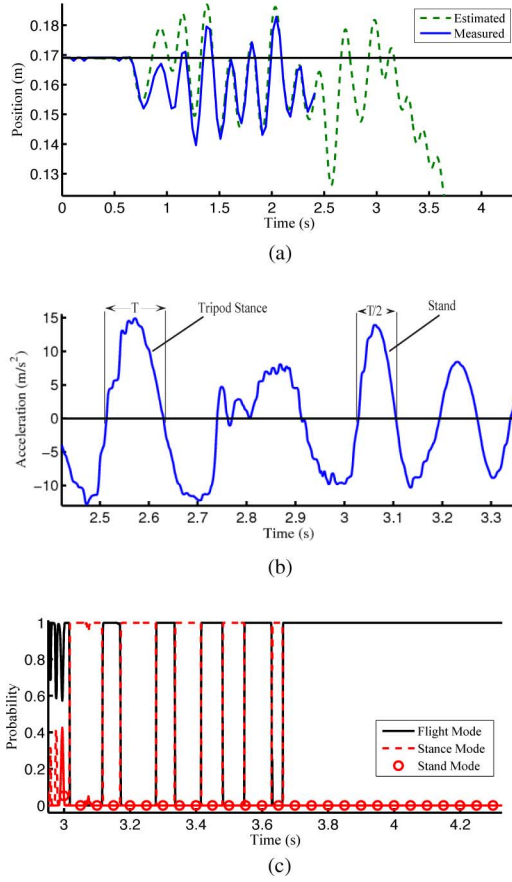


Fig. 18. Conventional GPB2 estimation fails when individual filters do not have enough time to converge. Although the limited workspace did not allow measurements after 2.5 s, the motors stop at 3 s, and the robot should rest at a height of 0.17 m. (a) Height estimates. (b) Oscillation periods. (c) GPB2 mode probabilities.

single-behavior results, even though they are conducted on a multiple-behavior system. The maximum number of filters used whenever the behavioral context is identified is four, which is not higher than when the robot executes jogging as a sole behavior.

D. Identification of Dynamical Context Enables Accurate Estimation

The estimation results obtained for jogging under steady-state conditions are extended to the entire run, which include starting and stopping dynamics. A conventional three-model, nine-filter GPB2 system is built, and its state estimates are plotted in Fig. 18(a). The plot shows that height estimates closely match ground-truth measurements over the steady-state region, but diverge rapidly thereafter. Efforts to tune GPB2 and model parameters such as D could not improve the results reported. The cause of the divergence can be found in Fig. 18(b), which plots the acceleration before and after the motors stop at 3 s. The plot reveals that the acceleration's period of oscillation shortens after stopping, which is explained by the robot that contacts the ground with all six legs and, therefore, behaving as a stiffer system. The new periods prove too short for individual KFs to converge, which causes the filters to output incorrect likeli-

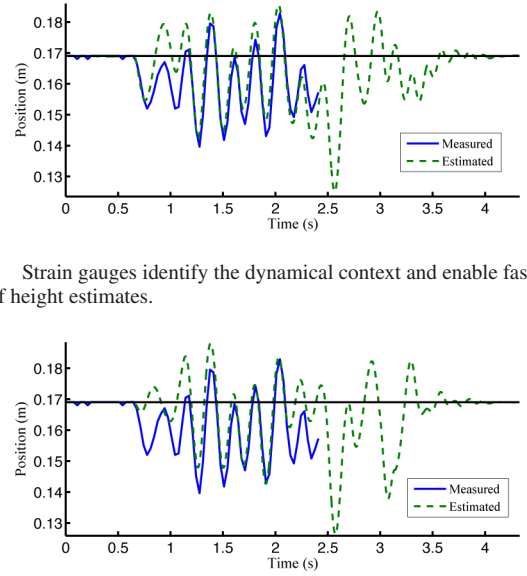


Fig. 19. Strain gauges identify the dynamical context and enable fast convergence of height estimates.

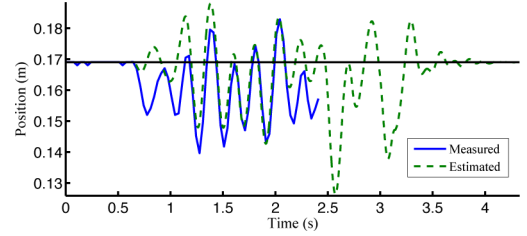


Fig. 20. Behavioral context information enables the selective activation of the jogging GPB2 and the stopping spring-damper model and leads to accurate and scalable estimation.

hoods and the GPB2 to assign incorrect weights to individual estimates. Fig. 18(c) shows that after the motors stop, the GPB2 assigns the highest weight to the flight mode instead of the stand mode, which justifies the downward slope of the height estimates.

The problem of slow convergence can be resolved by identifying the flight–stance–stand dynamical contexts from the leg-strain gauges, as per Fig. 10. When the strain gauges identify the context, the three-model GPB2 is reduced to a one-model filter (the filter corresponding to the identified context) that outputs accurate estimates. Thus, by incorporating contextual information, the mode probabilities are correctly and rapidly specified, which enables the successful estimation of the height throughout the experiment.

The results are reported in Fig. 19, which shows that the height estimates follow an oscillatory dissipative motion that ends at the robot's rest height of z_0 . The ground-truth measurement system's limited workspace does not allow the quantitative validation of the estimate's accuracy, but the fact that the estimates have the same frequency as the measured acceleration and that they converge to the rest height as expected, is an indication of validity.

E. Identification of Behavioral Context Prevents Divergence

The same estimation task is performed but without the use of strain gauges. For this, assume that the robot is either jogging or coming to rest and that motor state is unavailable. The strategy is to use a flight–stance GPB2 when the jogging behavioral context is identified and a stance model when the robot is no longer jogging. In other words, the behavioral context would identify the steady-state portions of the behavior, where the two-model GPB2 can be used with satisfactory accuracy.

Information about the behavioral context is provided by Fig. 14(b), and the estimation results are shown in Fig. 20.

The plot shows that even though height estimates are not as accurate as when using the strain gauges, they still converge back to the rest height.

The difference in accuracy is due to the fact that the behavioral contexts provide higher level information about the dynamics than do the dynamical contexts, but the results are still substantially better than when neither context is used.

Incorporating behavioral context information also reduces computational requirements, as it enables the deployment of a two-model instead of the three-model GPB2, thus reducing the number of filters from nine to four and improving overall scalability.

IX. CONCLUSION

Context-based estimation combines the conventionally separate fields of filter design and pattern recognition to enable accurate and scalable state estimation for systems with hybrid dynamics. Implementation examples provided in this paper show that simple approaches to pattern recognition (using HMMs) and to filtering (using KFs) lead to accurate estimates, which suggest that the designer does not need to acquire in-depth knowledge of each technique to generate satisfactory results.

Contexts are robustly and efficiently identified through discrete-state estimation using data generated by onboard sensors. The identification of the dynamical context determines which model is appropriate for estimation at a bandwidth comparable with sensors' update rate. This information is used to correctly set individual weights and improve estimation accuracy for fast-switching systems.

Information about the dynamical context also improves the scalability of the MM filters by enabling the selective activation of some filters and deactivation of others. Since dynamical contexts determine which model appropriately represents current dynamics, only individual filters based on the appropriate model are activated. This reduces the computational cost of conventional MM systems that run all filters at all times, thereby improving scalability.

Behavioral contexts are an abstracted description of a robot's behavior, which provides high-level understanding of the robot's health (is the robot jogging as commanded?) and allows for closed-loop behavioral control. The identification of the behavioral context also determines which collection of models appropriately represent the current switching dynamics. This prevents using inappropriate MM filters in the presence of unmodeled dynamics, which further reduces the risk of estimation failure and improving accuracy.

The additional advantage of knowing a system's behavioral context is that the MM estimators no longer need to activate all available filters but only the filters that correspond to the current context. This allows the deployment of small-scale estimators, which further increases the scalability of the estimation system.

Extensions will further improve the accuracy of discrete models by constructing HMMs that span multiple behaviors. In this configuration, the probability distribution α over HMM states becomes a measure of confidence in each state, as well as by extension in the contexts themselves. As such, α can be inter-

preted as a distance metric among behavioral contexts, which measures "how far" a robot's operation is from specific contexts. The distance metric can be used to close the loop on behavioral controllers, where a controller adjusts its parameters in ways that reduce distance to a target context, thereby improving control quality.

Another extension to this study is to apply these estimation techniques to a wider variety of systems, such as wall-climbing robots or snake-like robots. Our group has extensive experience in developing controllers for snake robots, but outside the low-level PID loop, we have done nothing to close the loop. Our goal is to close the loop at the behavioral level, which naturally would require an MM approach. However, the results in this paper, by themselves, are not sufficient for snake robots. These systems have many similar dynamic modes, which are not clearly delineated by simple sensor signatures; therefore, the classification of dynamic modes must employ a more complex feature vector than currently has been used.

REFERENCES

- [1] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robot. Auton. Syst.*, vol. 56, pp. 692–705, 2008.
- [2] K. Autumn, M. Buehler, M. Cutkosky, R. Fearing, R. J. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, A. Saunders, and D. E. Koditschek, "Robotics in sensorial environments," *Proc. SPIE*, vol. 5804, pp. 291–302, May 2005.
- [3] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. (Mathematics in Science and Engineering). Boston, MA: Academic, 1988.
- [4] M. E. Campbell and S. Brunke, "Nonlinear estimation of aircraft models for on-line control customization," in *Proc. IEEE Aerosp. Conf.*, pp. 2/621–2/628, vol. 2, Mar. 2001.
- [5] L. Campo, P. Mookerjee, and Y. Bar-Shalom, "State estimation for systems with sojourn-time-dependent Markov model switching," *IEEE Trans. Autom. Control*, vol. 36, no. 2, pp. 238–243, Feb. 1991.
- [6] C. B. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 3, pp. 418–425, May 1978.
- [7] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Trans. Robot. Autom.*, vol. 11, no. 6, pp. 781–793, Dec. 1995.
- [8] S. Hirose and A. Morishima, "Design and control of a mobile robot with an articulated body," *Int. J. Robot. Res.*, vol. 9, no. 2, pp. 99–114, 1990.
- [9] M. W. Hofbauer and B. C. Williams, "Mode estimation of probabilistic hybrid systems," in *Proc. Intl. Conf. Hybrid Systems: Comput Control*, New York: Springer-Verlag, 2002, pp. 253–266.
- [10] S. Kim, J. E. Clark, and M. R. Cutkosky, "Isprawl: Design and tuning for high-speed autonomous open-loop running," *Int. J. Robot. Res.*, vol. 25, no. 9, pp. 903–912, 2006.
- [11] T. Kirubarajan and Y. Bar-Shalom, "Target tracking using probabilistic data association-based techniques with applications to sonar, radar and eo sensors," in *Handbook of multisensor data fusion* (The Electrical Engineering and Applied Signal Processing Series), D. Hall and J. Llinas, Eds. Boca Raton, FL: CRC, 2001, pp. 203–242, ch. 10.
- [12] Z. Kohavi, *Switching and Finite Automata Theory*, 2nd ed. New York: McGraw-Hill, 1978.
- [13] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive Hidden Markov chains," *Int. J. Robot. Res.*, vol. 27, no. 7, pp. 761–784, 2008.
- [14] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication with impedance control for physical human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 1535–1542.
- [15] K.-F. Lee, "Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system," Ph.D. dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-88-148, Apr. 1988.

- [16] S. Lenser, "On-line robot adaptation to environmental change," Ph.D. dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Aug. 2005.
- [17] U. Lerner, "Hybrid bayesian networks for reasoning about complex systems," Ph.D. dissertation, Comput. Sci. Dept., Stanford Univ., Stanford, CA, Oct. 2002.
- [18] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, Apr. 1996.
- [19] X. R. Li, "Multiple model estimation with variable structure: Some theoretical considerations," in *Proc. 33rd IEEE Conf. Decis. Control*, Orlando, Dec. 1994, pp. 1199–1204.
- [20] X. R. Li, X. Zhi, and Y. Zhang, "Multiple-model estimation with variable structure part iii: Model-group switching algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 225–241, Jan. 1999.
- [21] P. S. Maybeck, "Stochastic models, estimation and control," in *Mathematics in Science and Engineering*, vol. 1, San Francisco, CA: Academic, 1979.
- [22] P. S. Maybeck, "Stochastic models, estimation and control," in *Mathematics in Science and Engineering*, vol. 2, San Francisco, CA: Academic, 1982.
- [23] P. S. Maybeck and P. D. Hanlon, "Performance enhancement of a multiple model adaptive estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 4, pp. 1240–1254, Oct. 1995.
- [24] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.
- [25] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [26] U. Saranlı, "Dynamic locomotion with a hexapod robot" Ph.D. dissertation, Univ. Michigan, East Lansing, MI, Sep. 2002.
- [27] E. Shamma, A. Wolf, and H. Choset, "Three degrees-of-freedom joint for spatial hyper-redundant robots," *J. Mech. Mach. Theory*, vol. 41, no. 2, pp. 170–190, Feb. 2006.
- [28] S. P. N. Singh and K. J. Waldron, "A hybrid motion model for aiding state estimation in dynamic quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4337–4342.
- [29] S. Skaff, "Context-based state estimation for hybrid systems with intermittent dynamics" Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2007.
- [30] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," presented at the 9th Int. Conf. Computer Vision, Nice, France, 2003.
- [31] V. Verma, "Tractable particle filters for robot fault diagnosis" Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, May 2005.
- [32] H. L. S. Younes, "Verification and planning for stochastic processes with asynchronous events," Ph.D. dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Jan. 2005.

Authors' photographs and biographies not available at the time of publication.