Proceedings of the 31st Conference
on Decision and Control
Tucson, Arizona • December 1992

Visual Observation for Hybrid Intelligent Control Implementation

Tarek M. Sobh*, Ruzena Bajcsy† and John R. James‡

Abstract

We address the problem of design and implementation of a discrete event dynamic system (DEDS) observer for the execution of commands sent to a robotic arm during grasping and screwing tasks of an assembly operation. We discuss the resulting robot arm mechanism as a hybrid intelligent system. We argue that the non-intrusive observation mechanism offers reliability and robustness advantages over other sensor systems used to detect errors. Hybrid systems contain both continuous and discrete components. Hybrid systems analysis differs from conventional digital control systems analysis in that the discrete component is appropriately modeled using a finite-state machine (DEDS) to describe high-level machine dynamics and the continuous component contains both analog and digital models of the lower-level continuous variable dynamic system (CVDS) portion of the device. Another description of hybrid systems is that they are networks consisting of continuous physical devices controlled by discrete digital programs, requiring digital-analog transfer of information for their control (logical-linguistic techniques may be used to derive control laws for continuous systems and machines are structured as an interconnected network of continuous, nonlinear systems and finite-state automata). Our work has focused on the use of the DEDS linguistic approach for abstracting high-level knowledge concerning the current state of the machine in the presence of errors, mistakes and uncertainties in the manipulation system.

1 Introduction

A previous paper has discussed a new framework and representation for the general problem of observation [9]. That paper also asserted that the system being studied can be considered a "hybrid" one, due to the fact that we need to report on *distinct* and *discrete* visual states that occur in the *continuous, asynchronous* and three-dimensional world, from two-dimensional observations that are sampled periodically. In this work we expand that discussion to elaborate on the application of the new framework to detect error states and sequences.

The problem of observing a moving agent has been addressed in the literature extensively. It was discussed in the work addressing tracking of targets and determination of the optic flow [1], recovering 3-D parameters of different kinds of surfaces [7], and also in the context of other problems [2]. However, the need to recognize, understand, and report on different visual steps within a dynamic task was not sufficiently addressed. In particular, there is a need for high-level symbolic interpretations of the agents' actions. A previous work [9] closely examined the possibilities for errors, mistakes, and uncertainties in the visual manipulation system, observer construction process and event identification mechanisms, leading to a DEDS formulation with uncertainties.

*Robotics and Vision Laboratory, Department of Computer Science, University of Utah, Salt Lake City, Utah 84112.
†GRASP Laboratory, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA 19104.
‡United States Army Training and Doctrine Command, VA.
Acknowledgments : This research was supported in part by Air Force AFOSR Grants 88-0244, 88-0296; Army/DAAL Grant 03-89-C-0031PRI; NSF Grants CISE/CDA 88-22719, IRI 89-06770; DARPA Grant N0014-88-0630 and DuPont Corporation.

2 DEDS

Discrete event dynamic systems (DEDS) are dynamic systems (typically asynchronous) in which state transitions are triggered by the occurrence of discrete events in the system. DEDS are usually modeled by finite state automata with partially observable events together with a mechanism for enabling and disabling a subset of state transitions [4,6]. We propose that this model is a suitable framework for many vision and robotics tasks, in particular, we use the model as a high-level structuring technique.

3 Observer Construction

Manipulation actions can be modeled efficiently within a discrete event dynamic system framework. We use the DEDS model to preserve and make use of information we know about the way in which each manipulation task should be performed. Each state in the automaton would represent a symbolic description of a stage in the manipulation process. In order to know the current state of the manipulation process we need to observe the sequence of events occurring in the system and make decisions regarding the state of the automaton. The goal will be to make the system a strongly output stabilizable one and/or construct an observer to satisfy specific task-oriented visual requirements.

4 Error States and Sequences

We utilize the observer framework to recognize error states and sequences. The idea behind this recognition task is to be able to report on ***visually incorrect*** sequences. In particular, if there is a predetermined observer model of a particular task under observation, then it would be useful to determine if something went wrong with the manipulation actions. The goal of this reporting procedure could be to alert operators or possibility to supply feedback to the manipulating robot so that it could correct its actions. Some examples of errors in manipulation include unexpected behaviour of the system, such as objects falling unexpectedly from the manipulating hand during a grasp and lift operation or some visual errors like unexpected occlusions between the observer camera and the manipulation environment.

There are a number of ways in which these problems could be reported. One such way can be to comply with the navigation strategy that was described in [8] in order to capture the current state (i.e. incrementally update belief in the current state and sequence as event matches occur while "navigating" the automata). If no match occurs, then the error would have to be reported. The correct sequences of automata state transitions can be formulated as the set of strings that are acceptable by the observer automaton. This set of strings represents precisely the language describing all possible visual task evolution steps.

5 Conclusions

The underlying mathematical representations of complex computer-controlled systems is still insufficient to create a set of models which accurately captures the dynamics of the system over the entire range of system operation. We remain in a situation where we must tradeoff the accuracy of our models with the manageability of the models. Closed-form solutions of mathematical models are almost exclusively limited to linear system models. Computer simulations of nonlinear and discrete-event models provide a means for off-line design of control systems through iterative search but such simulations cannot perform exhaustive search due to the complexity of the problem. Guarantees of system performance are limited to those regions where the robustness conditions apply. These conditions may not apply during startup and shutdown or during periods of anomalous operation. Excellent results are available for cases where adequate mathematical models are known and the system is operating "close enough" to a linear region. Also, effective tools are available to model high-level system changes as a finite state machine. Several attempts to improve our modeling capabilities are focused on mapping the continuous world into a discrete one However, repeated results are available which indicate that large interactive systems evolve into states where minor events can lead to a catastrophe. We are left with the result that there is a pressing need for a more adequate theory and mathematical basis for representing and predicting the performance of hybrid dynamical systems. Some current work has focused on providing a mathematical basis for the coupling of numerical and symbolic computing [3,5]. In the near term we will probably be able to mathematically prove (automatically verify) that the implementation of a subset of software for computer-controlled systems performs to specifications but will have to use conventional metrics for verification of the majority of the software being used. In this paper we have summarized a new framework and representation for the general problem of observation, emphasizing its' application to determining error states and sequences for the highest levels of computer-controlled systems. We assert that the framework provides a means for the explicit realization of transitions from low-level to high-level, goal-oriented knowledge in computer-controlled systems.

References

[1] P. Anandan, "A Unified Perspective on Computational Techniques for the Measurement of Visual Motion". In *Proceedings of the 1st International Conference on Computer Vision*, 1987.

[2] F. Chaumette and P. Rives, "Vision-Based-Control for Robotic Tasks", In *Proceedings of the IEEE International Workshop on Intelligent Motion Control*, Vol. 2, pp. 395-400, August 1990.

[3] Kohn, W and A. Nerode, Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design, March 1992.

[4] C. M. Özveren, *Analysis and Control of Discrete Event Dynamic Systems : A State Space Approach*, Ph.D. Thesis, Massachusetts Institute of Technology, August 1989.

[5] Platek, R and A. Nerode, "Distributed Intelligent Control", Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design, March 1992.

[6] P. J. Ramadge and W. M. Wonham, "Modular Feedback Logic for Discrete Event Systems", *SIAM Journal of Control and Optimization*, September 1987.

[7] T. M. Sobh and K. Wohn, "Recovery of 3-D Motion and Structure by Temporal Fusion". In *Proceedings of the 2nd SPIE Conference on Sensor Fusion*, November 1989.

[8] Sobh, T. M. "Active Observer: A Discrete Event Dynamic System Model for Controlling an Observer Under Uncertainty", Ph. D. Thesis, Department of Computer and Information Sciences, University of Pennsylvania, December 1991.

[9] Sobh, T. M. and R. Bajcsy, "A Model for Visual Observation Under Uncertainty", Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design, March 1992.