3rd Robotics and Knowledge-Based Systems Workshop, St. Hubert, Quebec, October 15-18, 1995

Applying Case-Based Reasoning to Control in Robotics*

Igor Jurisica†
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4

Janice Glasgow‡
Department of Computing and Information Science, Queen's University
Kingston, Ontario K7L 3N6

Abstract

The paper describes an application of a case-based reasoning system $\mathcal{TA}$**3** *to a control task in robotics. It differs from previous methods in its approach to similarity-based retrieval, which allows for greater flexibility and for improved accuracy in performance.*

The proposed architecture is experimentally evaluated on two real world domains and the results are compared to other machine learning algorithms applied to the same problem.

Introduction

A control system capable of remembering a repeated operation is called a learning control system (Arimoto, Kawamura and Miyazaki, 1984; Bondi, Casalino and Gambardella, 1988; Wang, Soh and Cheah, 1995). A learning process can increase the number of solvable tasks (i.e., acquire new knowledge) or it can improve the efficiency of solving repeated tasks. Most current applications of machine learning in robotics are oriented to the perfection of the control process. A learning control system for manipulators, whose endpoint is moving under geometrical constraints on a surface, was proposed in (Naniwa and Arimoto, 1995). (Jang, Choi and Ahn, 1995) describes an iterative learning system for precise tracking control with guaranteed convergence.

In (Mitchell, 1990) the Theo-Agent architecture of a reactive system is proposed. The system is used in robot control and combines a stimulus-response subsystem for rapid reaction with a search-based planner for handling unanticipated situations and an explanation-based learning mechanism to construct stimulus-response rules to cover new situations. Thus, the system becomes increasingly reactive, since over time, less planning is required as the machine learning component acquires more control rules.

The Candide system (Luzeaux and Zavidovique, 1994) has been used for the acquisition of a control knowledge. The system learns a qualitative model of the system first, then, based on the mode, it creates a rule-based incremental controller that controls a given system.

The SINS system (Ram and Santamaría, 1993) is a self improving reactive control system for autonomous robotic navigation. The system is based on using case-based techniques and reinforcement learning. The learning module monitors the system and incrementally modifies the case representations to accommodate the changes. Because of the utility problem (Minton, 1988), the SINS system uses only 10 cases to allow for a real time performance.

In this paper we propose a novel learning control system called $\mathcal{TA}$**3** , which incorporates case-based reasoning (CBR). A case-based approach (Kolodner, 1993) relies on the concept that similar problems have similar solutions. Thus, facing a new problem, a case-based system retrieves similar cases stored in a case base and adapts them to fit the problem at hand.

The $\mathcal{TA}$**3** system is similar to the Theo-Agent architecture (Mitchell, 1990) in the sense that it acquires new knowledge over time to improve the control component. The difference is in the form of acquired knowledge. Theo-Agent constructs stimulus-response rules, while $\mathcal{TA}$**3** acquires episodic knowledge. $\mathcal{TA}$**3** is also similar to SINS, mainly because both system use case-based techniques. The case base of SINS consists of continu-

*This research was supported by the Information Technology Research Center of Ontario. The authors are indebted to J. Mylopoulos and D. Lauzon for helpful comments and suggestions.

†Phone: (416) 978-7589, email: juris@cs.utoronto.ca
‡Phone: (613) 545-6058, email: janice@qucis.queensu.ca

ous, time-varying, analog information which is hard to explain. Moreover, the retrieval of such knowledge is expensive which required using only a small case base. $\mathcal{TA}$**3** 's cases are easy to explain and the system also allows for larger case bases without sacrificing the performance.

Informally, a case (an instance of an episodic knowledge) comprises an input (the problem), an output (the solution) and feedback (an evaluation of the solution). Case-based reasoning involves the process of:

1. accepting a new problem description (a case without a solution and feedback) from a client (user or another part of the system);
2. retrieving relevant cases from a case base (past problems with similar input);
3. adapting retrieved cases to fit the problem at hand and producing the solution for the input problem.

The feedback component for the case is determined after the control systems loop is completed at which time the result of the applied solution can be assessed.

Considering the case-based reasoning cycle, we can say that the more similar the cases are the less adaptation is necessary. Thus, in general, the proposed solution is more precise. Now, the task is how to measure case relevance (similarity or closeness) to always guarantee retrieving only highly relevant case. In many processes, it is better to retrieve fewer cases (or none) than to retrieve less relevant cases that would result in a poor solution.

Relevance Assessment

As was already mentioned, retrieving only relevant cases is the core part of the proposed system. Next, we present a formal definition of used terms.

Definition 1 (A case) *A case ($\mathcal{C}$) is a representation of a real world object or an episode in a particular representation scheme. A case is represented as a finite set of attribute/value pairs:*

$$\mathcal{C} = \{< a_0 : V_0 >, \cdots, < a_n : V_n >\} = \{< a_i : V_i >\}_0^n = \{A_i\}_0^n, \quad (1)$$

where $< a_i : V_i >$ is an attribute/value pair (A_i).

Definition 2 (Case base) *A finite set of cases is called a case base: $\Delta = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$.*

Next, we define context to allow for less strict matching than equivalence. The goal is to retrieve not only exact matches (equivalent cases) but partial matches (similar cases) as well. In short, context is a parameter of relevance relation which maps a case base onto a set of relevant (in terms of context) cases. Context allows for controlling what can and what cannot be considered as a partial match.

Definition 3 (Context) *A context (Ω) is defined as a finite set of attributes with associated constraints on the attribute values:*

$$\Omega = \{< a_0 : CV_0 >, ..., < a_k : CV_k >\} = \{< a_i : CV_i >\}_0^k, \quad (2)$$

where a_i is an attribute name and constraint CV_i specifies the set of "allowable" values for attribute a_i.

There is little commonality between definitions of relevance used in different systems (Greiner, 1994). The most common ground is reached when relevance is defined as something useful. Our definition of relevance is as follows:

Definition 4 (Relevance) *Retrieved cases are considered relevant if they are similar to the query with respect to current context.*

In the proposed theory, similarity is considered as a relation between two cases. It is defined to supplement equivalence by allowing for partial matches. In order to control to what level the match is partial, we use context as part of similarity relation. Then, two cases are considered similar, if they satisfy a given context. We say that a case satisfies (or matches) a particular context if for each attribute specified in the context the value of that attribute in the case satisfies the constraint (i.e., the value is contained in the constraint set).

Definition 5 (Satisfiability) *A case $\mathcal{C}$* **satisfies** *a context Ω, denoted $sat(\mathcal{C}, \Omega)$, if and only if for all pairs $< a_i : CV_i > \in \Omega$, there exists a pair $< a_i : V_i > \in \mathcal{C}$ such that V_i is in CV_i:*

$$sat(\mathcal{C}, \Omega) \text{ iff } \forall a_i (< a_i : CV_i > \in \Omega \rightarrow \exists V_i (< a_i : V_i > \in \mathcal{C} \wedge V_i \in CV_i)). \quad (3)$$

This definition can be relaxed so that a case may satisfy a context if it matches the context for some specified number $m < n$ attributes rather than requiring all the attribute constraints in the context be satisfied. We call this relaxation **cardinality relaxation**. Other type of relaxation used during retrieval is **value relaxation**, where attribute values are relaxed.

Definition 6 (Similarity) *A case* $\mathcal{C}_1$ *is* **similar** *to a case* $\mathcal{C}_2$ *for a given context* Ω*, denoted:* $\mathcal{C}_1 \sim_\Omega \mathcal{C}_2$*, if and only if both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *satisfy context* Ω*:*

$$\mathcal{C}_1 \sim_\Omega \mathcal{C}_2 \;\leftrightarrow\; sat(\mathcal{C}_1, \Omega) \wedge sat(\mathcal{C}_2, \Omega). \qquad (4)$$

More details on the theory of context-based similarity are provided in (Jurišica, 1994; Jurišica and Glasgow, 1995).

The $\mathcal{TA}$3 System

Based on a novel theory of relevance assessment introduced above, we have designed and implemented a case-based reasoning system $\mathcal{TA}$**3** . The architecture of the proposed system is presented in Figure 1. Individual components are described as follows:

The CB Generator

The CB Generator generates a case base (CB). There are three main possibilities:

1. Random case generation - cases are generated at random.
2. Uniform case generation - cases are uniformly distributed across the working space.
3. Task-oriented case generation - cases are generated via learning by observing the process.

The first two approaches are considered to be off-line learning methods and the third approach is an on-line method (we shall discuss these methods in more detail in the experimental studies presented later in the paper).

The Case Base

The Case Base is a persistent storage for cases. A case is represented as a finite set of attribute/value pairs, e.g., a case in a Servo Data domain is represented as follows:

```
     MOTOR:  E
     SCREW:  D
     PGAIN:  4
     VGAIN:  5
RISE_TIME:  0.28125095
```

The Retrieval Component

The retrieval component, based on the problem description, locates relevant cases from a case base. A novel theory of relevance measure is used, based on context-based similarity assessment. The goal of retrieval is to locate relevant cases, i.e., cases that are similar to a given problem.

If we use the above example, the problem description includes first four attribute/value pairs. The retrieval algorithm locates all cases that match the problem description.

As we already mentioned, we are interested not only in exact matches but in close (similar) matches as well. It should be noted that not all attributes are equally important during matching. The attributes that are prefered for the retrieval process depends not only on the application domain but on the task as well.

Our retrieval algorithm allows for flexibly changing the importance of individual attributes, which helps in retrieving only highly relevant cases. For example in the Servo Data domain, *motor* and *screw* attributes are the most important for locating relevant cases. The other two attributes also contribute but a partially matching case (not all attribute/value pairs in a given problem match a case in a case base) on *motor* and *screw* is more relevant than a partially matching case based on *pgain* and *vgain*.

The Adaptation Module

The adaptation module takes as an input all relevant cases and produces a solution for a given problem as an output. In the current case studies, we used automatic adaptation on the basis of weighted average of attribute values from retrieved cases. In other words, retrieved cases are ordered according to relevance measure, giving the most relevant cases the highest weight. An average is computed. (Note that in some domains the automatic adaptation might not be possible/desirable because of existence of non-numerical and complex attribute values.)

The Evaluation Component

The evaluation component considers produced solution and a feedback from the user. If there is a disparity between them, the case base is updated in order to prevent inaccurate solutions in the future.

The performance of $\mathcal{TA}$**3** is evaluated on several real world domains. Here we report on two problems: 1) predicting the rise time of a servomechanism in terms of two (continuous) gain settings and two (discrete) choices of mechanical linkages, and 2) predicting the joint angles for a three-link spheric, angular robot when given desired end-effector coordinates (inverse kinematic task).

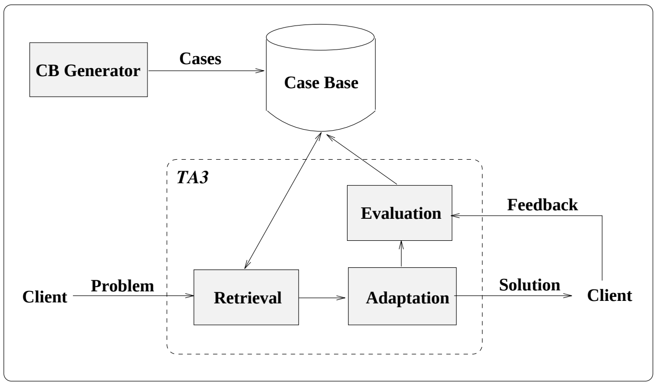

Figure 1: $\mathcal{TA}\mathbf{3}$ *architecture.*

Case Studies

Case Study 1: Servo Data

Servo Data is a data set created in 1986 by Karl Ulrich at MIT and it is publicly available on a machine learning database server.[1] Due to its availability and non-triviality, the database was used for testing of various machine learning algorithms, e.g., (Quinlan, 1992; Quinlan, 1993). Servo Data covers an extremely non-linear phenomenon - predicting the rise time of a servomechanism in terms of two (continuous) gain settings and two (discrete) choices of mechanical linkages.

Knowledge base

Characteristics of the Servo Data case base can be summarized as follows:

- 167 instances, generated by observation-based learning;
- 5 attributes per instance:
  1. motor – possible values: A,B,C,D,E;
  2. screw – possible values: A,B,C,D,E;
  3. pgain – possible values: 3,4,5,6;
  4. vgain – possible values: 1,2,3,4,5;
  5. class – possible values: real number from 0.13 to 7.10.

To create a Servo Data case base, a simulation of a servo system involving a servo amplifier, a motor, a lead screw/nut and a sliding carriage was used. The output value of the servo system simulation represents a rise time or the time required for the system to respond to a step change in a position set point. Thus, Servo Data can be characterized as a case base generated by observation.

[1] ftp:ics.uci.edu:pub/machine-learning-programs/

Evaluation of retrieval function

Even though Servo Data is not a complex (in terms of number of attributes) or a large (in terms of number of cases) domain, it is of interest since it involves continuous class prediction and various machine learning algorithms have been evaluated and compared on it. Table 1 summarizes results for $\mathcal{TA}\mathbf{3}_{Servo}$ and other machine learning techniques, obtained by using simple cross-validation (leave-one-out method) and averaged over 10 and 60 random trials respectively.

Relaxation strategy 1 was used in $\mathcal{TA}\mathbf{3}_{Servo\ 1}$, i.e., only the *motor* attribute was relaxed and only to the left side (e.g., B would be relaxed to A *or* B). $\mathcal{TA}\mathbf{3}_{Servo\ 2}$ was obtained by using relaxation strategy 2, i.e., only the *motor* attribute was relaxed but in both directions (e.g., B would be relaxed to A *or* B *or* C). It should be noted that relaxing A and E would be the same in both scenarios. In a situation where the initial relaxation did not yield a result, additional methods are used. First, the cardinality relaxation for *motor* and *screw* is tried. Second, the cardinality relaxation for *pgain* and *vgain* is performed.

Even though (direct) error comparison can be used for performance evaluation and system comparison, it is not sufficient in general. In the data presented in Table 1 there is no information on the confidence in the results, nor is there any information about the significance of the errors or about the confidence intervals. In other words, statistical evidence is missing. Thus, in addition to the comparison presented in Table 1 we also conducted a statistical evaluation of our system, the results of which are presented in Table 3.

Table 3 shows that it is advantageous to have the option of relaxing the constraints on attribute values. Moreover, considering the statistical confidence of the result, we observed that 40 trials is not sufficient for 99% confidence with a reasonable small confidence interval; thus, we used average from 60 trials.

Case Study 2: Robotic Data

Robotic Data covers an extremely non-linear phenomenon - predicting the joint angles for the three-link spheric, angular robot when given desired end-effector coordinates (inverse kinematic task). Since for this task we had robot parameters and necessary equations, we created several databases using different criteria (varying database size, distribution of instances, etc.) (Jurišica,

Method	*Average Abs. Error*	*Relative Error*
Guessing mean	1.15	1.00
Instance-based	0.52	0.26
Regression	0.86	0.49
Model trees	0.45	0.29
Neural nets (G. Hinton)	0.30	0.11
Regression+instances	0.48	0.20
Model trees+instances	0.30	0.17
NN+instances	0.29	0.11
$\mathcal{TA}\mathbf{3}_{Servo\,1}$ (60 trials)	0.118	0.073
$\mathcal{TA}\mathbf{3}_{Servo\,2}$ (60 trials)	0.09	0.056

Table 1: *Servo-mechanism domain – Absolute and relative errors on servo-data using simple cross-validation; average over 20 random trials.*

Method	*Average Abs. Error*	*Relative Error*
Instance-Based	9.29 times	4.6 times
NN + Instance-Based	3.2 times	1.96 times

Table 2: *Servo-mechanism domain – Improvement of $\mathcal{TA}\mathbf{3}_{Servo\,2}$, over neural nets + instance-based and plain instance-based approaches.*

	Average Abs. Error		*Relative Error*	
Method	*99% Confidence*	*95% Confidence*	*99% Confidence*	*95% Confidence*
$\mathcal{TA}\mathbf{3}_{Servo\,1}$ (60)	0.118 ± 0.042	0.118 ± 0.032	0.073 ± 0.032	0.073 ± 0.024
$\mathcal{TA}\mathbf{3}_{Servo\,2}$ (60)	0.09 ± 0.035	0.09 ± 0.027	0.056 ± 0.012	0.056 ± 0.009

Table 3: *Simple cross-validation – results averaged over 60 random trials of $\mathcal{TA}\mathbf{3}_{Servo}$.*

1995). Due to the practical nature of these problems, data availability and objective relevance measure, similar tasks have been used to test machine learning algorithms in the past (Aha and Salzberg, 1994).

The task under consideration is known as an inverse kinematic, e.g., computation of the robot's joint angles to ensure that the end-effector coordinates will reach a desired position. This is a complex task which might be computationally intractable for complex systems. In such situations, other methods are used to approximate the solution.

In our study we used an existing small robot as an example. The robot's parameters are presented in Table 4, where L_1, L_2, L_3 are lengths of links (robotic arms) and $\varphi_1, \varphi_2, \varphi_3$ are ranges for individual angles.

For the given architecture, the equations presented in Table 5 are used to compute the end-effector coordinates P (forward kinematic task).

The inverse kinematic computation (computation of robot's joint angles based on end-effector coordinates) does not necessarily yield a single solution. Moreover, complex kinematic structures might not have an analytical solution. In such situations, some optimality criteria need to be defined and used for the computation. The other possibility is to use previous knowledge and compute only increments from the current position. Nevertheless, the equations for the inverse kinematic of our architecture are presented in Table 6.

An inverse kinematic task is needed during planning – the robot's end-effector is at point X and must be moved to point Y. For example in welding, the usual task is to follow a particular curve during the process. Thus, the inverse kinematic must be computed reasonably fast.

Knowledge base

The Robotic Data data set can be characterized as follows (refer to Figure 2):

- 2,000 instances, uniformly generated across the entire working area;
- 9 attributes per instance:
  1. P_x: x coordinate of the end-effector (float);
  2. P_y: y coordinate of the end-effector (float);
  3. P_z: z coordinate of the end-effector (float);
  4. L_1: length of the first link (float);

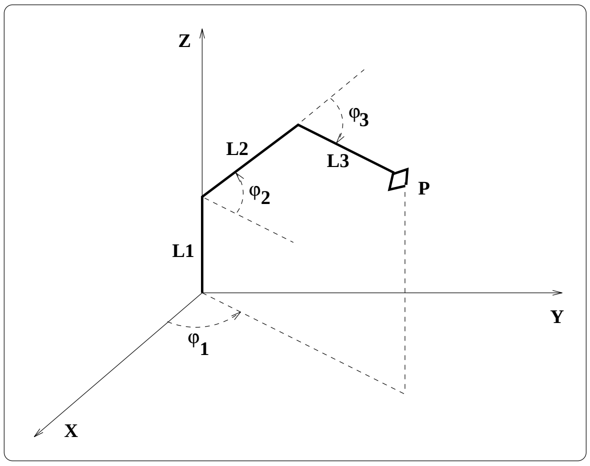

Figure 2: *Spheric angular robotic system.*

  5. L_2: length of the second link (float);
  6. L_3: length of the third link (float);
  7. φ_1: base angle (float);
  8. φ_2: shoulder angle (float);
  9. φ_3: elbow angle (float);
  10. X, Y, Z: coordinate frame.

In the first experiment, the complete knowledge base was created uniformly, i.e., all 2,000 cases were uniformly distributed over the working space of the robot. This is useful for off-line learning in order to guarantee a reasonable answer in any possible situation.

The other possibility is to generate cases randomly, which is not useful in general. The main reason is that it does not represent the real world situation; there is rarely a random move of the robotic arm in its working space.

The last possibility is to "observe" actual cases while the robot is performing normal tasks. The problem with this approach is that it requires on-line operation (or a reliable simulation as in the Servo Data case base) and that the working space will likely not be represented evenly by the case base. Thus, the system will be able to work in similar situations to the ones observed during learning, but the performance will degrade if there is any change in the robot task. On-line learning can be a useful experience added to the off-line generate knowledge base, which can be used for generalizations and for knowledge-based adaptation of retrieved cases.

Evaluation of retrieval function

During system evaluation we varied several parameters, such as the size of the case base, the den-

L_1	L_2	L_3	φ_1	φ_2	φ_3
0.203 m	0.178 m	1.178 m	$(-90^o, 90^o)$	$(-35^o, 145^o)$	$(-150^o, 0^o)$

Table 4: *Robot characteristics.*

$$
\begin{aligned}
P_x &= L_2 \cos\varphi_1 \cos\varphi_2 + L_3 \cos\varphi_1 \cos(\varphi_2 + \varphi_3) && (5)\\
P_y &= L_2 \cos\varphi_1 \cos\varphi_2 + L_3 \sin\varphi_1 \cos(\varphi_2 + \varphi_3) && (6)\\
P_z &= L_1 + L_2 \sin\varphi_2 + L_3 \sin(\varphi_2 + \varphi_3) && (7)
\end{aligned}
$$

Table 5: *Equations for the forward kinematic task.*

sity of individual cases and the distribution of individual cases. We used various adaptation techniques while we also varied the relaxation process. During evaluation a simple cross-validation (leave-one-out method) was applied and significance intervals and variances from the average values over 20 random trials were computed (see Table 7).

In $\mathcal{TA}\mathbf{3}_{Robot\ 1}$ only cardinality relaxation was used (i.e., only m of n, $m < n$, attributes were required to match). A similar approach presented in (Ortega, 1995) yielded identical results – accuracy improvement for $m < n$. $\mathcal{TA}\mathbf{3}_{Robot\ 2}$ used value relaxation (i.e., relaxing attribute values) first and if this failed then cardinality relaxation was performed.

Conclusions

The results of the experiments prove the viability of our approach. They also provide motivation for work on robot path planning, where the system not only selects a path for the robot based on the task definition but may also create a sequence of values for joint angles to be followed in order to fulfill the task.

Our experiments show that a fairly high accuracy is achieved when CBR is used in control. For applications requiring extremely high precision, a possible combination of a CBR approach for an approximate solution and an optimizer for producing an accurate solution seems feasible.

In the future, more experimentation on case base generation is necessary. This would answer the questions: How big should the case base be? – considering the tradeoff of speed and accuracy, and How should the cases be distributed in a case base? – considering either task-oriented distribution, uniform distribution or random distribution. Also, as already mentioned, we plan to use the $\mathcal{TA}\mathbf{3}$ system for a more complex task of planning the robot's actions.

References

Aha, D. W. and Salzberg, S. L. (1994). Learning to catch: Applying nearest neighbor algorithms to dynamic control tasks. In Cheeseman, P. and Oldford, R. W., editors, *Selecting Models from Data: Artificial Intelligence and Statistics IV*. Springer-Verlag.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation fo robots by learning. *Journal of Robotic Systems*, (1):123–140.

Bondi, P., Casalino, G., and Gambardella, L. (1988). On the iterative learning control theory for robotic manipulators. *IEEE Journal of robotcs and Automation*, 4(1):14–22.

Greiner, R. (1994). *AAAI Fall Symposium Series on Relevance*. AAAI Press, Menlo Park, CA.

Jang, T., Choi, C., and Ahn, H. (1995). Iterative learning control in feedback systems. *Automatica*, 31(2):243–248.

Jurišica, I. (1994). Context-based similarity applied to retrieval of relevant cases. Technical Report DKBS-TR-94-5, University of Toronto, Department of Computer Science, Toronto, Ontario.

Jurišica, I. (1995). Retrieval and explanation similarity functions: Performance evaluation. Technical report, University of Toronto, Department of Computer Science, Toronto, Ontario. In preparation.

Jurišica, I. and Glasgow, J. (1995). Context-based similarity for case-based reasoning. submitted.

Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.

$$
\begin{aligned}
\varphi_1 &= \arctan \frac{P_y}{P_x} && (8)\\
\varphi_2 &= \arctan \frac{P_z - L_1}{\sqrt{P_x^2 + P_y^2}} \pm \arccos \frac{L_2^2 - L_3^2 + P_x^2 + P_y^2 + (P_z^2 - L_1)^2}{2L_2\sqrt{P_x^2 + P_y^2 + (P_z^2 - L_1)^2}} && (9)\\
\varphi_3 &= \pm\left[\pi - \arccos \frac{L_2^2 + L_3^2 - P_x^2 - P_y^2 - (P_z^2 - L_1)^2}{L_2 L_3}\right] && (10)
\end{aligned}
$$

Table 6: *Equations for the inverse kinematic task.*

Method	*Average Abs. Error*			*Relative Error*		
	φ_1	φ_2	φ_3	φ_1	φ_2	φ_3
$\mathcal{TA}\mathbf{3}_{Robot\ 1}$	0.4855 ± 0.2128	0.0	0.0	0.00586863 ± 0.00257	0.0	0.0
$\mathcal{TA}\mathbf{3}_{Robot\ 2}$	0.0	2.5 ± 1.0957	0.0	0.0	0.0274725 ± 0.0120	0.0

Table 7: *Robotic domain – absolute and relative errors on robotic data using simple cross-validation; average over 20 random trials (95% confidence intervals).*

Luzeaux, D. and Zavidovique, B. (1994). Process control and machine learning: Rule-based incremental control. *IEEE Transactions on Automatic Control*, 39(6):1166–1171.

Minton, S. N. (1988). Quantitative results concerning the utility of explanation-based learning. In *Proc. of AAAI-88*, pages 564–569, St. Paul, MN. Also published in Journal of Artificial Intelligence, 42, 1990, pages 363-381.

Mitchell, T. M. (1990). Becoming increasingly reactive. In *Proc. of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, CA.

Naniwa, T. and Arimoto, S. (1995). Learning control for robot tasks under geometric endpoint constraints. *IEEE Transactions on Robotics and Automation*, 11(3):432–441.

Ortega, J. (1995). On the informativeness of the DNA promoter sequences domain theory. *Journal of Artificial Intelligence Research*, 2:361–367. Research Note.

Quinlan, J. R. (1992). Learning with continuous classes. In *Proc. 5th Australian Joint Conference on AI*, pages 343–348.

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *Proc. of the 10th International Conference on Machine Learning*, Amherst, MA.

Ram, A. and Santamaría, J. C. (1993). Continuous case-based reasoning. In *Proc. of the AAAI-93 Workshop on Case-Based Reasoning*, Washington, DC.

Wang, D., Soh, Y. C., and Cheah, C. C. (1995). Robust motion and force control f constrained manipulators by learning. *Automatica*, 31(2):257–262.