

# Towards a Probabilistic Asynchronous Linear Control Theory

Daniel Polani

Adaptive Systems Research Group  
Department of Computer Science  
University of Hertfordshire  
d.polani@herts.ac.uk

**Abstract.** A framework for asynchronous stochastic linear control theory is introduced using a simple example motivated by the early RoboCup soccer server dynamics. Worst and average case scenarios are studied and it is demonstrated that they fit smoothly into the framework of standard synchronous control theory.

## 1 Introduction

In the early versions of the RoboCup Soccer Simulator, a central problem was the synchronization of the soccer clients with the soccer simulator. The games were run in real-time, but the UDP/IP protocol used does not guarantee that the communication packets will reach the other side such as to guarantee a well-defined temporal relation between the agent state and the soccer server state [11]. An important question was how a consistent world view could be maintained [12]. Commands sent by the agents to the server, as well as the agent world model might not be synchronized with the server cycle. In tournaments from 1999 onwards, faster machines, networks and better synchronization algorithms, like that developed by CMU [14] increasingly made this view obsolete and allowed agents to become synchronized with the simulator.

In the present paper, we will revisit the original problem. But why return to a problem which seems to be no longer relevant? There are several reasons for that: First, the existence of precisely timed world states is a model which is a coarse approximation of physical reality. In newer versions of the soccer server simulator it is considered to introduce a continuous time, where actions will be incorporated into the physical dynamics *as they arrive*; this means that they will not be buffered to be processed only at the fixed time steps of the simulation update. Second, although current hardware robots are often organized in synchronized time steps due to technical reasons, it is far from clear that, say, their sensorics and processing systems should obey this principle of equal and consistent timing. Different concepts of timing and of time processing exist in biological systems and may, among other aspects, be responsible for the phenomenon of consciousness [6]. Third, the concept of what constitutes a worldly “reality” is a generally intriguing question. In the extreme cases of relativity theory, the concept of “now” loses all its meaning without additional assumptions. The unsynchronized relation between world and agent time in the early soccer server can be interpreted as a kind of “unorganized relativistic system” For this reason, we believe that the question of studying systems with a stochastic asynchrony bears an interest in its own right. In the present paper, we will study a linear control model for a strongly specialized class of stochastic asynchronous control systems as a starting point for a generalized theory of stochastic asynchronous control.

Asynchronous systems are of interest in the development of highly integrated systems where tight timing schedules make asynchrony an integral part of the dynamics [2]; such systems are usually of digital nature and analyzed using Petri networks [9]. In continuous systems, the typical approach is linear [13] or non-linear [10] control theory, typically assumed to be synchronous. Another line of research studies the dynamics in spiking neural networks [8] exhibiting phenomena of asynchronicity not unlike technical systems. Hassibi et al. [4] introduce a method for the control of asynchronous dynamical systems. The method considers asynchronous events in the control loop whose behaviour is only restricted by an event rate. In their work, few assumptions are made and stochastic aspects of time delays are not modelled. These, however, will be part of the present work. The relation between models studied in this paper and switched systems [7] as well as Markovian jumping systems [1] will be studied in a later paper.

## 2 A Scenario

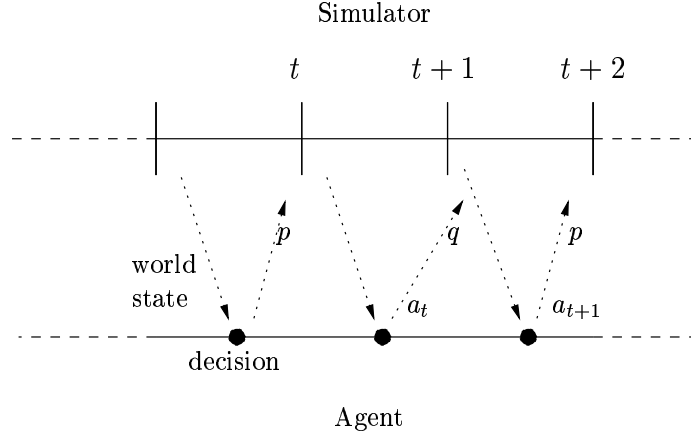
### 2.1 The Synchronization

We concentrate on a simple scenario in the framework of the early soccer server to illustrate the approach. Assume that at integer times  $t = 1, 2, 3 \dots$  the server world state is updated according to its dynamics [3], taking into account all the agent action commands that arrived before  $t$ . After time  $t$ , the server sends out its updated world state to the agents. Upon receiving this update, the agents decide on an action to take and send it to the server which will incorporate it into the simulation step at time  $t + 1$ . We now make several assumptions: 1. The server sends out world state information after each simulation step. This does not conform to the standard setting of the soccer server, but serves to simplify our present considerations. 2. The world state information sent out at time  $t$  is processed by the agent which then issues an action command. 3. The action command from 2., called  $a_t$ , has a probability of  $p$  of reaching the server in time for the next simulation step. In that case, it will be included in calculation of the next world state.  $a_t$  has a probability of  $q = 1 - p$  of not reaching the server in time, and instead after time  $t + 1$ . What now happens depends on whether the following action command  $a_{t+1}$  reaches the server in time or not. If  $a_{t+1}$  is late again, then at time  $t + 2$  the server performs action  $a_t$  already in its buffer. If  $a_{t+1}$  is in time, though, it will overwrite  $a_t$  in the server buffer and will be performed at time  $t + 2$  (this is the case shown in Fig. 1).

### 2.2 An Example for a Control Variable

As example, let us assume that we wish to control the orientation  $\phi$  of the agent, in approximation for small angles so that we can assume that  $\phi \in \mathbb{R}$ . The goal is now a linear control rule that will stabilize  $\phi$  to 0. Consider the synchronous control case. If the orientation after time  $t$  is  $\phi_t$ , then sending  $a_t = \Delta\phi_t$  results in a new orientation  $\phi_{t+1} = \phi_t + \Delta\phi_t$  after the next update. The control rule is linear in  $\phi$ :  $a_t := -c\phi_t$ , the update rule  $\phi_{t+1} = \phi_t - c\phi_t$ . The best possible control is achieved setting  $c := 1$ , immediately stabilizing the angle at 0.

Consider now the asynchronous case with a synchronization pattern as in Sec. 2.1. Represent the state of the system by the vector  $(\phi_t, \phi_t^B)^T$ . Here,  $\phi_t^B$  is the possible action from before time  $t$  stored in the server buffer. If no action is stored set  $\phi_t^B$  to 0.  $\phi_t^B$  is not known to the agent. As in the synchronous case, set  $a_t := -c\phi_t$ . Now there are two possibilities: 1. With probability  $p$ , the action will reach the server in time to be processed at  $t + 1$ . The new system state will



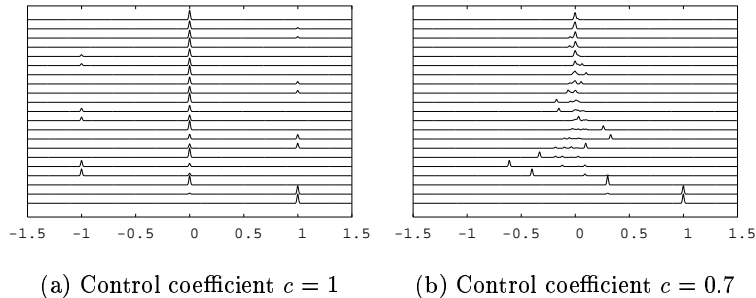
**Fig. 1.** The simulation-control cycle. The arrows from the server time-line to the agent denote sensor world information sent from server to agent, the arrows from the agent time-line to the server denote the action commands. With probability  $p$ , action  $a_t$  reaches the server in time for the update at  $t + 1$ , with probability  $q = 1 - p$ , it is too late and stays in the buffer until after  $t + 1$ . If the following command  $a_{t+1}$  is late again,  $a_t$  in the buffer will be processed at time  $t + 2$ . The present figure, however, shows the case where  $a_{t+1}$  reaches the server in time and overwrites  $a_t$  in the buffer; thus, at time  $t + 2$   $a_{t+1}$  will be carried out.

now be  $(\phi_{t+1}, \phi_{t+1}^B)^T = (\phi_t - c\phi_t, 0)^T$ . The 0 in the second component indicates the empty action buffer. 2. With probability  $q = 1 - p$ , the action will be late for the update at  $t + 1$ . Thus, the new orientation will be obtained using the action in the server action buffer, and the present command will end up in the action buffer:  $(\phi_{t+1}, \phi_{t+1}^B)^T = (\phi_t + \phi_{t+1}^B, -c\phi_t)^T$  In case 1., the transition matrix for the states is given by  $T_p = \begin{pmatrix} 1 & -c & 0 \\ 0 & 0 & 0 \end{pmatrix}$  In case 2., the transition matrix is given by  $T_q = \begin{pmatrix} 1 & 1 \\ -c & 0 \end{pmatrix}$

### 2.3 Solving the Control Problem: Example

The linear control of a single variable is the simplest possible control scenario in the synchronous case and a special instance of a control problem. In our example  $c := 1$  solves the problem. In the asynchronous case, this is not anymore the case. Assume an extreme delay probability, e.g., that the probability that the action command is received in time  $p = 0.1$ .

Figure 2(a) shows the dynamics of the system. From bottom to top the state information for times  $t = 0, \dots, t = 20$  are shown. The graphs show the probability distribution that the agent has a given orientation at a certain time. The bottommost line shows the probability distribution at time  $t = 0$ . The agent starts with the orientation  $\phi_0 := 0$  and the server with an empty action buffer. The command to correct the orientation by  $-c\phi$  reaches the server in time only with probability  $p = 0.1$ . We see that in the second graph from the bottom. With probability 0.1, the agent has assumed the desired orientation of 0. In the rest of the cases, the agent has remained in its original orientation since the action command has not reached the server.



**Fig. 2.** Example for asynchronous control of the orientation if the probability  $p$  that the action command reaches the server in time is  $p = 0.1$ . The time axis is  $t = 0$  to  $t = 20$  from bottom to top; the diagram shows the probability that the agent has a certain orientation if it issues a change orientation command  $-c\phi$  in each step. For more details, see text.

Now, if still in state  $\phi = 1$ , the agent will reissue a reorientation command while there is already such a command in the queue. In the following step  $t = 2$ , the agent will then have assumed orientation 0, because now the original  $t = 0$  reorientation action has finally reached the server (or, either, it had already reached  $\phi = 0$  and does not need reorientation). However, since in 90% of the cases the agent had reissued a reorientation command in time step  $t = 1$ , the system overshoots; this process repeats itself in the next cycles. It takes considerable time until the probability for  $\phi = 1$  or  $\phi = -1$  becomes negligible. While therefore in the undelayed and synchronous control problem it makes sense to set  $c = 1$ , this setting is too “aggressive” in the asynchronous case. For comparison, the same control problem is shown in Fig. 2(b) for  $c = 0.7$ .

### 3 Towards a Methodology for Asynchronous Linear Control

Is there a consistent method to handle the asynchronous stochastic control case? The present paper will demonstrate an approach in the concrete example, because the concretization is more useful in bringing the point across. The principle is not limited to the present case and can easily be extended to different and more general systems. Since we do not have a single-valued system variable like in deterministic control theory, but a probability distribution of the system variable, we have a variety of choices for the criterium according to which we wish to control. Here, we will discuss the worst case scenario and the average case scenario.

#### 3.1 The Worst Case Scenario

The worst case scenario is not concerned with the probability that a certain state is assumed, as long as it is assumed; the worst possible state (i.e. the state that is the farthest away from the desired one) determines the quality of the control. To explain this, consider the case of Fig. 2(a). Here, even for large times  $t$  there is still a nonzero probability for the orientation still to be at  $\phi \in \{-1, 1\}$ , i.e.

as far away from the target state as in the beginning. Though this probability decays exponentially, for the worst case scenario this is not relevant. It is the slowest decaying branch of the probability dynamics that counts. In the worst case scenario, one is interested to control the system in such a way as to maximize the decay of the slowest decaying branch. The present section demonstrates the approach, using the matrix lub-norm (“lub” = lowest upper bound).

**Decay Calculation via LUB-Norm in a Simple Case** Assume for a minute that  $p$  were 1, i.e. we had a synchronized system with the transition matrix as above  $T_p = \begin{pmatrix} 1-c & 0 \\ 0 & 0 \end{pmatrix}$ . Then, for a general initial state  $x_0$ , the state at time  $t$  is given by  $T_p^t x_0 = \underbrace{T_p \dots T_p}_{t \text{ times}} x_0$ . An upper bound for the worst case is obtained via

the lub-norm  $\|\cdot\|$  of matrices<sup>1</sup>. Obviously  $\|T_p^t\| = |1-c|^t$ . The strongest decay in this case is obtained by  $c := 1$  as done above.

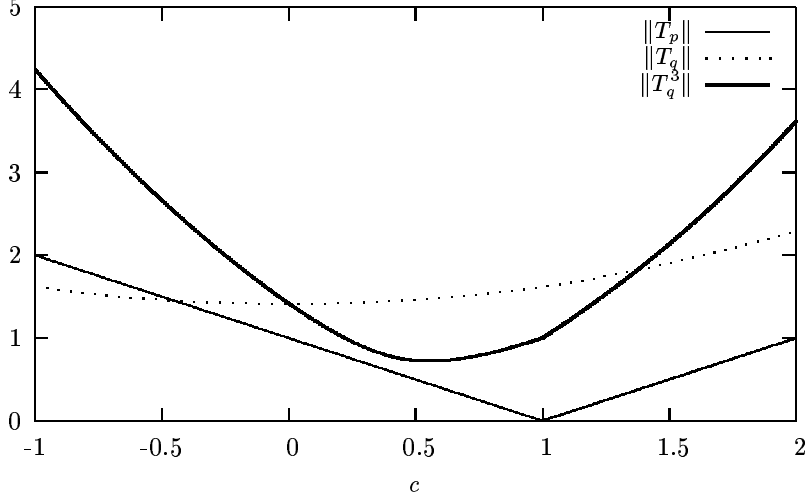
**The General Case** In our scenario, neither  $p$  nor  $q$  vanish. Thus the set of states that may have a nonzero probability at a time step  $t$  is given by  $\{x \mid x = T_{k_t} T_{k_{t-1}} \dots T_{k_1} x_0, \text{ with } k_i \in \{p, q\} \text{ for } i = 1 \dots t\}$ . This is the set of all possible states  $x$  arising via all possible combinations of  $t$  delayed and undelayed simulation updates. Call such a sequence of updates a *simulation run*. We now show how one can obtain increasingly accurate decay boundaries which can be used to determine the optimal control variable  $c$ . The coarsest possible boundary can be calculated via the property of submultiplicativity of the lub norm (Footnote 1). We obtain for the state after any possible  $t$ -step simulation runs  $\|x_t\|_2 = \|T_{k_t} T_{k_{t-1}} \dots T_{k_1} x_0\|_2 \leq \|T_{k_t}\| \|T_{k_{t-1}}\| \dots \|T_{k_1}\| \|x_0\|_2 \leq \left(\max_{k \in \{p, q\}} \|T_k\|\right)^t \|x_0\|_2$ .

In this approximation, the decay rate is bounded by  $\max_{k \in \{p, q\}} \|T_k\|$ . We know  $\|T_p\| = |1-c|$  and have to calculate the lub-norm of  $T_q = \begin{pmatrix} 1 & 1 \\ -c & 0 \end{pmatrix}$ . One obtains  $T_q^T T_q = \begin{pmatrix} 1+c^2 & 1 \\ 1 & 1 \end{pmatrix}$ . This gives the set of eigenvalues  $\{\frac{c^2}{2} + \frac{\sqrt{c^4+4}}{2} + 1, \frac{c^2}{2} - \frac{\sqrt{c^4+4}}{2} + 1\}$ , the square root of the larger of the which is the lub-norm of  $T_q$ . The results are shown in Fig. 3. The larger of the two values  $\|T_p\|$  and  $\|T_q\|$  is the estimate for the worst-case decay factor achievable by selecting a given value  $c$ . The figure shows that for no choice of  $c$ ,  $\|T_q\|$  becomes smaller than 1. I.e., it is not possible to construct a decay rate below 1 considering only a one-step lookahead. However, by iterating  $T_q$  three times, there are, in fact, values of  $c$  for which the lub norm of  $T_q^3$  drops below 1 (Fig. 3<sup>2</sup>).

We therefore see that the lub-norm provides us only with an upper bound for the worst case scenario; if we wish more accurate boundaries, one has to construct the possible simulation runs (sequences of  $ps$  and  $qs$ ) and calculate the pertinent lub-norms. Calculating the lub-norms for  $T_p T_p T_p$ ,  $T_p T_p T_q$ ,  $T_p T_q T_p$ ,  $T_p T_q T_q$ ,  $T_q T_p T_p$ ,  $T_q T_p T_q$ ,  $T_q T_q T_p$ ,  $T_q T_q T_q$  gives the plot shown in Fig. 4. This procedure can be extended to longer sequences of  $p$  and  $q$  and used to determine the pertinent control coefficient  $c$ .

<sup>1</sup> For a matrix  $A$ , the lub-norm is defined as  $\|A\| = \sup_x \frac{\|Ax\|_2}{\|x\|_2}$  with  $\|\cdot\|_2$  the Euclidean metric. One obtains  $\|A\| = \sqrt{\lambda_{\max}}$  with  $\lambda_{\max}$  the largest eigenvalue of  $A^T A$ . The lub-norm fulfils the criterium of *submultiplicativity*,  $\|AB\| \leq \|A\| \|B\|$  [5].

<sup>2</sup> An alternative way of showing that would have been to consider the eigenvalues of  $T_q$  and seeing that their modulus is smaller than 1; however, this does not allow us to identify the best value of  $c$ .



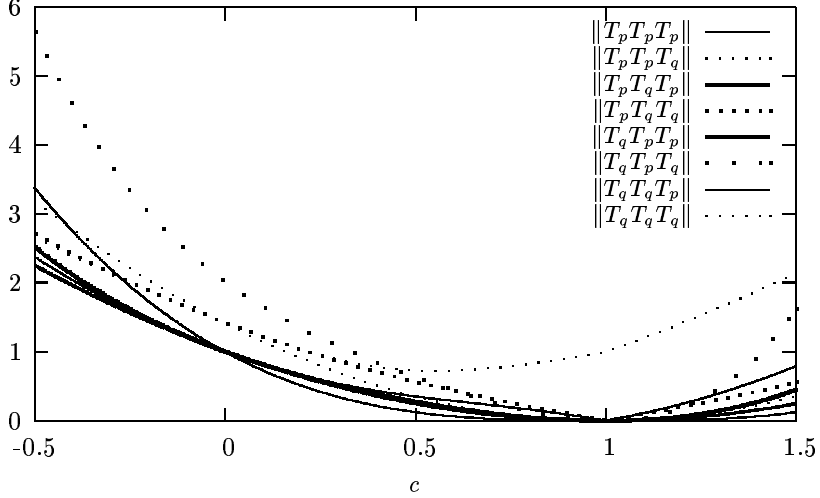
**Fig. 3.** The values of  $\|T_p\|$ ,  $\|T_q\|$  and  $\|T_q^3\|$  for different values of  $c$ .

**Reduction to State Variable** The above calculation has still not exploited further options to reduce the boundary. Since we are not actually interested in having the total norm  $\|x\|$  of the state vector decaying, but actually only the first component,  $x_1$  which is the orientation (the second component is the buffered action which we are not interested in), what we are actually seeking is to maximize the worst case decay for  $PT_{k_t}T_{k_{t-1}}\dots T_{k_1}$  where  $P$  is the projection of the current total state vector onto the orientation component. For a more accurate estimate, the control variable  $c$  should minimize the lub-norm of above term. We will not proceed to do that here, but the idea is analogous to Sec. 3.1.

### 3.2 The Average Case Scenario

The average case scenario differs from the worst case one that probabilities are incorporated in calculating the state vector whose behavior is going to be controlled. A simulation with  $t$  updates results in a vector  $T_{k_t}T_{k_{t-1}}\dots T_{k_1}x_0$  with probability  $k_t k_{t-1} \dots k_1$  with  $k_i \in \{p, q\}$  (note that we abuse notation by using  $p$  and  $q$  as index names for the  $T$ , but as numerical values for the  $k$ ).

For the expectation value for the state vector we have to sum up over all the possible probabilities and combination of  $p$  and  $q$ , obtaining  $\mathbf{E}x = \sum_{(k_t, \dots, k_1) \in \{p, q\}^t} k_t \dots k_1 T_{k_t} \dots T_{k_1} x_0 = (pT_p + qT_q)^t x_0$ , where the first equation is the probability-weighted sum over all possible mutually disjoint  $t$ -step histories and the second equation arises from the binomial theorem. This quantity is a matrix operator obtained by averaging  $T_p$  and  $T_q$  according to their weight and iterating it  $t$  times. To this operator, the methods from Sec. 3.1 can be applied again in a straightforward manner. Thus, we can smoothly incorporate the average case scenario in the methodology developed above. The same holds for the projection  $P$  of the average state on the orientation component.



**Fig. 4.** Plot of the lub-norms for all three-step simulation runs. Since it is difficult to distinguish the varieties of different graphs, it should be mentioned that the maximum of the lub-norms is dominated by  $T_p T_p T_p$  for smaller values of  $c$  and by  $T_q T_q T_q$  for larger values of  $c$ .

## 4 Conclusion

The methodology developed here to describe and solve the linear stochastic asynchronous control problem has been described given a specific example, to make it more accessible, but is not restricted to it. It smoothly encompasses both description of worst case and average case. Standard analysis methods from linear control theory [13] can be easily applied to the formalism, thus making available the toolbox of standard control theory to the stochastic asynchronous case, thereby complementing the formalism of standard control theory.

## 5 Summary and Outlook

A formalism to solve a linear stochastic asynchronous control problem was presented in the framework of a simple, but paradigmatic case. It was discussed in view of the worst and average case scenario. Future work will study the relation of the present framework to existing work on switched systems and Markovian jumping models [7, 1]. In addition, it will focus on the relation of information theory, control and time and combine it to extend the information-theoretic perspective of control theory from [16, 15] towards a better understanding of how information and time are related in a control loop. We believe that these questions are not just of mainly academic interest, but will instead lead to a better fundamental understanding of the role of time in intelligent information processing.

*Acknowledgement* The author thanks the anonymous reviewers for their helpful comments.

## Bibliography

- [1] Boukas, E. K., and Yang, H., [1995]. Stability of discrete time linear systems with Markovian jumping parameters. *MCSS*, 8:390–402.
- [2] Cortadella, J., Kishinevsky, M., Burns, S. M., and Stevens, K., [1999]. Synthesis of asynchronous control circuits with automatically generated relative timing assumptions. *ICCAD*, 324–331.
- [3] Corten, E., Heintz, K. D. F., Kostiadis, K., Kummeneje, J., Myritz, H., Noda, I., Rieki, J., Riley, P., Stone, P., and Yeap, T., [1999]. *Soccerserver Manual Ver. 5 Rev. 00 beta (for Soccerserver Ver.5.00 and later)*. <http://www.dsv.su.se/johank/RoboCup/manual/>, March 19, 2001
- [4] Hassibi, A., Boyd, S. P., and How, J. P., [1999]. Control of Asynchronous Dynamical Systems with Rate Constraints on Events. *38th IEEE Conference on Decision and Control*, 2:1345–1351.
- [5] Huppert, B., [1990]. *Angewandte lineare Algebra*. De Gruyter.
- [6] Koch, C., and Crick, F., [2001]. The neural basis of consciousness. In *Intl. Encyclopedia of the Social and Behavioral Sciences*, 2600–2604. Elsevier.
- [7] Liberzon, D., and Morse, A., [1999]. Basic Problems in Stability and Design of Switched Systems.
- [8] Maass, W., [2003]. Computation with spiking neurons. In Arbib, M. A., editor, *Handbook of Brain Theory and Neural Networks*, 1080–1083. Cambridge: MIT Press. Second edition.
- [9] Murata, T., [1989]. Petri Nets: Properties, analysis and applications. *Proceedings of the IEEE*, 541–580.
- [10] Nijmeijer, H., and van der Schaft, A. J., [1990]. *Nonlinear Dynamical Control Systems*. New York Berlin Heidelberg: Springer-Verlag.
- [11] Noda, I., Matsubara, H., Hiraki, K., and Frank, I., [1998]. Soccer Server: A Tool for research on Multi-Agent Systems. *Journal of Applied Artificial Intelligence*, 12(2–3).
- [12] Polani, D., Weber, S., and Uthmann, T., [1998]. The Mainz Rolling Brains in RoboCup '98: A Direct Approach to Robot Soccer. In *Workshop "RoboCup" at the KI-98*.
- [13] Sontag, E. D., [1990]. *Mathematical control theory; Deterministic finite dimensional systems (Texts in Applied Mathematics)*, vol. 6. New York: Springer-Verlag.
- [14] Stone, P., Veloso, M., and Riley, P., [1999]. The CMUnited-98 Champion Simulator Team. In Asada, M., and Kitano, H., editors, *RoboCup-98: Robot Soccer World Cup II*. Berlin: Springer Verlag. To Appear.
- [15] Touchette, H., and Lloyd, S., [2000]. Information-Theoretic Limits of Control. *Phys. Rev. Lett.*, 84:1156.
- [16] Touchette, H., and Lloyd, S., [2001]. Information-theoretic approach to the study of control systems. *IEEE Transactions on Information Theory*. Submitted. <http://xxx.lanl.gov/abs/physics/0104007>, Feb 2003