# Reconstruction of Low-Complexity Control Programs from Data

Florent Delmotte and Magnus Egerstedt

{florent,magnus}@ece.gatech.edu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

*Abstract*— **In this paper we study the problem of generating control programs, i.e. strings of symbolic descriptions of control-interrupt pairs (or modes) from input-output data. In particular, we take the point of view that such control programs have an information theoretic content and thus that they can be more or less effectively coded. As a result, we focus our attention on the problem of producing low-complexity programs by recovering the strings that contain the smallest number of distinct modes. An example is provided where the data is obtained by tracking ten roaming ants in a tank.**

## I. INTRODUCTION

As the complexity of many control systems increases, multi-modal control has emerged as a useful design tool. The main idea is to define different modes of operation, e.g. with respect to a particular task, operating point, or data source. These modes are then combined according to some discrete switching logic and one attempt to formalize this notion is through the concept of a *Motion Description Language* (MDL) [4], [6], [9], [11].

Each string in a MDL corresponds to a control program that can be operated on by the control system. Slightly different versions of MDLs have been proposed, but they all share the common feature that the individual atoms, concatenated together to form the control program, can be characterized by control-interrupt pairs. In other words, given a dynamical system

$$\begin{array}{l} \dot{x} = f(x,u),\ x \in \mathbb{R}^N,\ u \in U \\ y = h(x),\ y \in Y, \end{array} \tag{1}$$

together with a control program $(k_1,\xi_1),\ldots,(k_z,\xi_z)$, where $k_i : Y \rightarrow U$ and $\xi_i : Y \rightarrow \{0,1\}$, the system operates on this program as $\dot{x} = f(x, k_1(h(x)))$ until $\xi_1(h(x)) = 1$. At this point the next pair is read and $\dot{x} = f(x, k_2(h(x)))$ until $\xi_2(h(x)) = 1$, and so on[1].

Now, a number of results have been derived for such (and similar) systems, driven by strings of symbolic inputs, i.e. when the control and interrupt sets are finite. For example, in [3], the set of reachable states was characterized, while [7] investigates optimal control aspects of such systems. In [5], [9], [11], the connection between MDLs and robotics was investigated. However, in this paper we continue the development begun in [2], where the control programs are viewed as having an information theoretic content. In other words, they can be coded more or less effectively. Within this context, one can ask questions concerning minimum complexity programs, given a particular control task.

But, in order to effectively code symbols, drawn from a finite alphabet, one must be able to establish a probability distribution over the alphabet. If such a distribution is available then Shannon's celebrated source coding theorem [13] tells us that the minimal expected code length $l$ satisfies

$$\mathcal{H}(\mathcal{A}) \leq l \leq \mathcal{H}(\mathcal{A}) + 1, \tag{2}$$

where $\mathcal{A}$ is the alphabet, and where the *entropy* is given by

$$\mathcal{H}(\mathcal{A}) = \sum_{a \in \mathcal{A}} p(a) \log_2 \frac{1}{p(a)}, \tag{3}$$

where $p(a)$ is the probability of drawing the symbol $a$ from $\mathcal{A}$. The main problem that we will study in this paper is how to produce an empirical probability distribution over the set of modes, given a string of input-output data.

The outline of this paper is as follows: In Section II we define the problem at hand and show how this can be addressed within the MDL framework. In Section III we show how to find mode sequences that contain the smallest number of distinct modes, followed by a description of how these types of sequences can be modified for further complexity reductions, in Section IV. In Section V we give an example where the control programs are obtained from data generated by 10 roaming ants.

## II. MOTION DESCRIPTION LANGUAGES

The concept of a *motion alphabet* has been proposed recently in the literature as a finite set of symbols representing different control actions that, when applied to a specific system, define segments of motion [4], [8], [11]. A MDL is thus given by a set of strings that represent such idealized motions, and particular choices of MDLs become meaningful only when the language is defined relative to the physical device that is to be controlled.

If we assume that the input and output spaces ($U$ and $Y$ respectively) in Equation (1) are finite, which can be justified by the fact that all physical sensors and actuators have finite range and resolution, the set of all possible

This work was sponsored by the National Science Foundation through the program ECS NSF-CAREER award (grant # 0237971).

[1] Note that the interrupts can also be time-triggered, but this can easily be incorporated by a simple augmentation of the state space.

modes $\Sigma_{total} = U^Y \times \{0,1\}^Y$ is finite as well. We can moreover adopt the point of view that a data point is measured only when the output or input change values, i.e. when a new output or input value is encountered. This corresponds to a so called *Lebesgue sampling*, in the sense of [1], which allows us to only study input-output strings of finite length, as was the case in [2]. Under this sampling policy, we can define a mapping $\delta : \mathbb{R}^N \times U \to \mathbb{R}^N$ as $x_{p+1} = \delta(x_p, k(h(x_p)))$, given the control law $k : Y \to U$, with a new time update occurring whenever a new output or input value is encountered. For such a system, given the input string $(k_1, \xi_1), \ldots, (k_z, \xi_z) \in \Sigma^*$ where $\Sigma \subseteq \Sigma_{total}$, the evolution is given by

$$\begin{cases} x(q+1) = \delta(x(q), k_{l(q)}(y(q))), \quad y(q) = h(x(q)) \\ l(q+1) = l(q) + \xi_{l(q)}(y(q)). \end{cases} \tag{4}$$

Now, given a mode sequence of control-interrupt pairs $\sigma \in \Sigma^\star$ (here $\Sigma^\star$ denotes the set of all mode strings over $\Sigma$), we are interested in how many bits we need in order to specify $\sigma$. If no probability distribution over $\Sigma$ is available, this number is given by the *description length* defined in [12]:

$$\mathcal{D}(\sigma, \Sigma) = |\sigma| \log_2(card(\Sigma)),$$

where $|\sigma|$ denotes the length of $\sigma$, i.e. the total number of modes in the string. This measure gives us the number of bits required for describing the sequence in the "worst" case, i.e. when all the modes in $\Sigma$ are equally likely. However, if we can establish a probability distribution $p$ over $\Sigma$, the use of optimal codes can, in light of Equation (2), reduce the number of bits needed, which leads us to the following definition:

**Definition (Specification Complexity):** *Given a finite alphabet $\Sigma$ and a probability distribution $p$ over $\Sigma$. We say that a word $\sigma \in \Sigma^*$ has specification complexity*

$$\mathcal{S}(\sigma, \Sigma) = |\sigma| \mathcal{H}(\Sigma).$$

An initial attempt at establishing a probability distribution over $\Sigma \subseteq U^Y \times \{0,1\}^Y$ was given in [2]. In that work, the main idea was to recover modes (and hence also the empirical probability distribution) from empirical data. For example, supposing that the mode string $\sigma = \sigma_1\sigma_2\sigma_1\sigma_3$ was obtained, then we can let $\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$, and the corresponding probabilities become $p(\sigma_1) = 1/2,\ p(\sigma_2) = 1/4,\ p(\sigma_3) = 1/4$. In such a case where we let $\Sigma$ be built up entirely from the modes in the sequence $\sigma$, the empirical specification complexity depends solely on $\sigma$:

$$\mathcal{S}^e(\sigma) = |\sigma|\mathcal{H}^e(\sigma) = -\sum_{i=1}^{M(\sigma)} \lambda_i(\sigma) \log_2 \frac{\lambda_i(\sigma)}{|\sigma|}, \tag{5}$$

where $M(\sigma)$ is the number of distinct modes in $\sigma$, $\lambda_i(\sigma)$ is the number of occurrences of mode $\sigma_i$ in $\sigma$, and where we use superscript $e$ to stress the fact that the probability distribution is obtained from empirical data.

Based on these initial considerations, the main problem, from which this work draws its motivation, is as follows:

**Problem (Minimum Specification Complexity):** *Given an input-output string $S = (y(1), u(1)), (y(2), u(2)), \ldots, (y(n), u(n)) \in (Y \times U)^n$, find the minimum specification complexity mode string $\sigma \in \Sigma^*_{total}$ that is consistent with the data. In other words, find $\sigma$ that solves*

$$\mathrm{P}(\Sigma_{total}, \boldsymbol{y}, \boldsymbol{u}) : \begin{cases} \min_{\sigma \in \Sigma^*_{total}} \mathcal{S}^e(\sigma) \\ \text{subject to } \forall q \in \{1, \ldots, n\} \\ \begin{cases} \sigma_{l(q)} = (k_{l(q)}, \xi_{l(q)}) \in \Sigma_{total} \\ k_{l(q)}(y(q)) = u(q) \\ \xi_{l(q)}(y(q)) = 0 \Rightarrow l(q+1) = l(q), \end{cases} \end{cases}$$

*where the last two constraints ensure consistency of $\sigma$ with the data $S$, and where $\boldsymbol{y} = (y(1), \ldots, y(n))$, $\boldsymbol{u} = (u(1), \ldots, u(n))$ give the empirical data string.*

Note that this is slightly different than the formulation in Equation (4) since we now use $\sigma_{l(q)}$ to denote a particular member in $U^Y \times \{0,1\}^Y$ instead of the $l(q)$-th element in $\sigma$.

Unfortunately, this problem turns out to be very hard to address directly. However, the easily established property

$$0 \le \mathcal{H}^e(\sigma) \le \log_2(M(\sigma)),\ \forall \sigma \in \Sigma^\star_{total}$$

allows us to focus our efforts on a more tractable problem. Here, the last inequality is reached when all the $M(\sigma)$ distinct modes of $\sigma$ are equally likely.

As a consequence, we have $\mathcal{S}^e(\sigma) \le |\sigma| \log_2(M(\sigma))$ and thus it seems like a worth-while endeavor, if we want to find low-complexity mode sequences, to try to minimize either the length of the mode sequence $|\sigma|$ or the number of distinct modes $M(\sigma)$. In fact, minimization of $|\sigma|$ was done in [2], while the minimization of $M(\sigma)$ is the main pursuit in this paper:

**Problem (Minimum Distinct Modes):**
*Given an input-output string $S = (y(1), u(1)), (y(2), u(2)), \ldots, (y(n), u(n)) \in (Y \times U)^n$, find $\sigma$ that solves*

$$\mathrm{Q}(\Sigma_{total}, \boldsymbol{y}, \boldsymbol{u}) : \begin{cases} \min_{\sigma \in \Sigma^*_{total}} M(\sigma) \\ \text{subject to } \forall q \in \{1, \ldots, n\} \\ \begin{cases} \sigma_{l(q)} = (k_{l(q)}, \xi_{l(q)}) \in \Sigma_{total} \\ k_{l(q)}(y(q)) = u(q) \\ \xi_{l(q)}(y(q)) = 0 \Rightarrow l(q+1) = l(q). \end{cases} \end{cases}$$

## III. THE ALWAYS INTERRUPT SEQUENCE

**Definition (Always Interrupt Sequence):** *We will refer to any mode string $\sigma = \sigma_1 \ldots \sigma_n \in \Sigma^*_{total}$ such that*

$$\begin{cases} M(\sigma) \triangleq card\{\sigma_i \mid \sigma_i \in \sigma\} \\ \qquad\quad \triangleq card\{l(q) \mid q = 1, \ldots, n\} \qquad (6) \\ \qquad\quad = \max_{y \in Y}(card\{u \mid (y,u) \in S\}) \\ \xi_{l(q)}(y(q)) = 1, \quad q = 1, \ldots, n \qquad (7) \end{cases}$$

*as an Always Interrupt Sequence (AIS)*[2].
Here, Equation (6) means that the total number of distinct modes $M(\sigma)$ used in the AIS is equal to the maximum number of different input values $u$ associated with an output value $y$ in the sense that $(u, y)$ appears as an input-output pair in the data string. One direct consequence of Equation (7) is that the length of an AIS is equal to the length $n$ of the input-output string it is consistent with.

**Existence:** *Given an input-output string $S \in (Y \times U)^n$ there always exist an Always Interrupt Sequence consistent with the data.*

*Proof:* The consistency of a mode string with the data $S$ is ensured by the two conditions :

$$\forall q \in \{1, \ldots, n\}, \quad \left\{ \begin{array}{l} k_{l(q)}(y(q)) = u(q) \\ \xi_{l(q)}(y(q)) = 0 \Rightarrow l(q+1) = l(q) \end{array} \right.$$

Let $M$ denote $\max_{y \in Y}(card\{u \mid (y, u) \in S\})$. For every $y \in Y$, there exist $m \leq M$ distinct values of $u$ such that $(y, u) \in S$. One possible way to construct an AIS consistent with the data is to associate one distinct mode from the $M$ available modes with each of the different values of $u$, i.e.

$$\forall (i, j) \text{ such that } y(i) = y(j),\ u(i) \neq u(j) \Rightarrow l(i) \neq l(j)$$

By doing so for every value of $y$ encountered in $S$, we ensure that

$$\forall (i, j), \quad \left\{ \begin{array}{l} l(i) = l(j) \\ y(i) = y(j) \end{array} \right. \quad \Rightarrow u(i) = u(j)$$

so that the first condition is met. The second condition is always met since by definition of the AIS, $\xi_{l(q)}(y(q)) = 1,\ q = 1, \ldots, n$.
Hence we have constructed an AIS that is consistent with the data $S$. ∎

One important fact should be noted here. In the proof, we proposed one particular AIS but there are many different ways to construct an AIS consistent with the data.

**Example.** Given the following input-output string

| | |
|---|---|
| $y$ | 0 0 1 2 2 0 1 1 0 1 2 2 1 2 1 0 2 0 2 1 |
| $u$ | 4 2 1 2 3 0 3 3 1 1 4 4 0 2 3 4 4 0 1 0 |

We have $Y = \{0, 1, 2\}$ and:

$$\begin{aligned} card\{u \mid (0, u) \in S\} &= card\{4, 2, 0, 1, 4, 0\} &= 4 \\ card\{u \mid (1, u) \in S\} &= card\{1, 3, 3, 1, 0, 3, 0\} &= 3 \\ card\{u \mid (2, u) \in S\} &= card\{2, 3, 4, 4, 2, 4, 1\} &= 4 \end{aligned}$$

so that an AIS will use $M = \max\{4, 3, 4\} = 4$ modes here.

As seen in the previous proof for existence, one way to build an AIS is to, for each $y \in Y$, establish an injective mapping between $U$ and $U^Y$. For example, we can use:

| mode | $y = 0$ | $y = 1$ | $y = 2$ |
|---|---|---|---|
| 1 | $k_1(0) = 4$ | $k_1(1) = 1$ | $k_1(2) = 2$ |
| 2 | $k_2(0) = 2$ | $k_2(1) = 3$ | $k_2(2) = 3$ |
| 3 | $k_3(0) = 0$ | $k_3(1) = 0$ | $k_3(2) = 4$ |
| 4 | $k_4(0) = 1$ | | $k_4(2) = 1$ |

Thus we get the following $l$-string:

| | |
|---|---|
| $y$ | 0 0 1 2 2 0 1 1 0 1 2 2 1 2 1 0 2 0 2 1 |
| $u$ | 4 2 1 2 3 0 3 3 1 1 4 4 0 2 3 4 4 0 1 0 |
| $l$ | 1 2 1 1 2 3 2 2 4 1 3 3 3 1 2 1 3 3 4 3 |

and the corresponding mode sequence is $\sigma = \sigma_1\sigma_2\sigma_1\sigma_1\sigma_2\sigma_3\sigma_2\sigma_2\sigma_4\sigma_1\sigma_3\sigma_3\sigma_3\sigma_1\sigma_2\sigma_1\sigma_3\sigma_3\sigma_4\sigma_3$.

**Theorem:** *Any mode string consistent with a given input-output string S is such that its number of modes is greater than or equal to $M = \max_{y \in Y}(card\{u(q) \mid (y, u(q)) \in S,\ q \in \{1, \ldots, n\}\})$.*

*Proof:* Suppose that there exists a mode string $\sigma$ consistent with the data using only $m < M$ modes. Consider the value of $y \in Y$ such that $card\{u(q) \mid (y, u(q)) \in S,\ q \in \{1, \ldots, n\}\} = M$ and label it $y_M$. In other words, there exist $M$ different values of $u \in U$ such that $(y_M, u) \in S$. As $m < M$ there must exist two couples $(y_M, u(i))$ and $(y_M, u(j))$ in $S$ with $u(i) \neq u(j)$ that are associated with the same mode, say $\sigma_x = \sigma_{l(i)} = \sigma_{l(j)}$. As the mode string is supposed to be consistent, we can write $k_x(y_M) = u(i)$ for the first couple and $k_x(y_M) = u(j)$ for the second one. But as $u(i) \neq u(j)$ we have a contradiction.
Consequently, any mode string consistent with a given input-output string S must use at least $M$ modes. ∎

**Corollary.** *Any AIS consistent with the data is a solution to the problem* Q$(\Sigma_{total}, \boldsymbol{y}, \boldsymbol{u})$.

*Proof:* To be consistent with the data, a mode string must use at least $M$ modes. An AIS consistent with the data uses exactly $M$ modes. Thus it solves $Q(\Sigma_{total}, \mathbf{y}, \mathbf{u})$. ∎

Since there is a large number of AIS consistent with the data, one problem would be to pick the one that minimizes $\mathcal{S}^e(\sigma)$. However, as will be seen in the next section, there are potentially better ways of obtaining low complexity programs by abandoning the AIS structure.

## IV. Modified Mode Sequences

### A. The Sometimes Interrupt Sequence

Here we introduce a method that reduces the length of a given AIS. The idea is to modify the interrupt function $\xi$ of each mode and make it be equal to zero (i.e. no mode change) whenever possible. Ideally, a

[2] Note that given a finite set $C$, by $card(C)$ we understand the number of different elements in $C$, e.g. $card(\{c_1, c_2, c_1, c_3\}) = 3$

sequence like $\sigma = \sigma_1\sigma_1\sigma_2\sigma_2\sigma_2\sigma_1\sigma_1\sigma_1\sigma_2\sigma_2$ could then be reduced to $\sigma' = \sigma_1\sigma_2\sigma_1\sigma_2$. This method, if plausible, would not add any new modes. The resulting sequence would still use exactly $M$ distinct modes and would thus be another solution to $Q(\Sigma_{total}, \mathbf{y}, \mathbf{u})$. But as the $\xi$ functions are modified, the resulting mode sequence is no longer an AIS. In this matter, the resulting sequence will be referred to as a *Sometimes Interrupt Sequence* (SIS).

**Algorithm (Sometimes Interrupt Sequence):** *Given an AIS $\sigma = \sigma_{l(1)}, \ldots, \sigma_{l(n)}$ consistent with an input-output sequence $S = (y(1), u(1)), \ldots, (y(n), u(n))$, we construct the associated SIS by :*

1) *keeping $\xi_x(y(q)) = 1$ for all mode $\sigma_x$ whenever $\exists q \in \{1, \ldots, n\}$ such that $l(q) = x$ and $l(q+1) \neq x$,*
2) *changing all the other values of $\xi$ to zero, for all modes.*

**Theorem.** *The SIS derived from an AIS consistent with the data is consistent with the data.*

*Proof:* We recall here again the two conditions for consistency :

$$\forall q \in \{1, \ldots, n\}, \quad \begin{cases} k_{l(q)}(y(q)) = u(q) \\ \xi_{l(q)}(y(q)) = 0 \Rightarrow l(q+1) = l(q). \end{cases}$$

The modifications of the AIS mode sequence only concern the $\xi$ functions, i.e. the interrupts. Thus, we just have to prove that the modified sequence does not violate the second consistency condition.

Suppose we have a case where $\xi_{l(q)}(y(q)) = 0$ and $l(q+1) \neq l(q)$. This is impossible as it contradicts the first step in the construction of the SIS. Thus the second condition for consistency is always met and the SIS derived from an AIS which is consistent with the data is consistent with the data as well. ■

**Example.** Consider the following input-output string and the given AIS mode sequence $\sigma$ (or equivalently the $l$ string) which is consistent with this data.

| $y$ | 1 | 0 | 2 | 0 | 2 | 2 | 1 | 2 | 2 | 0 | 1 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u$ | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| $l$ | 1 | →2 | 2 | →1 | 1 | 1 | →2 | 2 | →1 | →2 | 2 | 2 |

Now construct the associated SIS :

1) The arrows in the table show us whenever $l(q) \neq l(q+1)$, i.e. whenever we need to keep $\xi_{l(q)}(y(q)) = 1$. Here, we need to keep $\xi_1(1) = 1$, $\xi_1(2) = 1$ and $\xi_2(2) = 1$.
2) So we can set $\xi_1(0) = 0$, $\xi_2(0) = 0$ and $\xi_2(1) = 0$.

Consequently, the mode switches happening at $q = 2, 4, 7, 10$ and 11 have been suppressed and the above mode string has been reduced from $\sigma = \sigma_1\sigma_2\sigma_2\sigma_1\sigma_1\sigma_1\sigma_2\sigma_2\sigma_1\sigma_2\sigma_2\sigma_2$ with length $N = 12$ to $\sigma = \sigma_1\sigma_2\sigma_1\sigma_1\sigma_2\sigma_1\sigma_2$ with length 7.

It can be easily shown that the action of removing one element from a mode string $\sigma$ strictly reduces its specification complexity $\mathcal{S}^e(\sigma)$. The SIS is thus a mode sequence with lower complexity than the AIS it is derived from.

### *B. The Minimum Entropy AIS*

Here we present a particular AIS that can be obtained using the following algorithm:

**Algorithm (Minimum Entropy AIS):**
*For all distinct y,*

- *sort the different (y,u) by decreasing r((y,u)), where r((y,u)) denotes the number of occurrences of the pair (y,u) in S.*
- *associate mode 1 with the most recurrent (y,u)*
- *associate mode 2 with the second most recurrent (y,u)*

⋮ ⋮ ⋮ ⋮

- *associate mode M with the least recurrent (y,u)*

*The resulting AIS is called a Minimum Entropy AIS.*

Let us here note that there can be more than one Minimum Entropy AIS for a given input-output string $S$. This is the case whenever $\exists(y, u_1, u_2)$ with $u_1 \neq u_2$ and $r((y, u_1)) = r((y, u_2)) > 1$ .

**Example:** Given the following input-output sequence $S$:

| $y$ | 1 | 0 | 2 | 0 | 2 | 2 | 1 | 2 | 0 | 0 | 1 | 2 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u$ | 0 | 0 | 1 | 2 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 2 |

where $Y = \{0, 1, 2\}$. We have :

$$\text{for y=0,} \quad \begin{cases} r((0,0)) = 2 \\ r((0,2)) = 2 \end{cases} \Rightarrow \begin{cases} (0,0) \text{ is given mode } 1 \\ (0,2) \text{ is given mode } 2 \end{cases}$$

$$\text{for y=1,} \quad \begin{cases} r((1,1)) = 1 \\ r((1,0)) = 2 \end{cases} \Rightarrow \begin{cases} (1,0) \text{ is given mode } 1 \\ (1,1) \text{ is given mode } 2 \end{cases}$$

$$\text{for y=2,} \quad \begin{cases} r((2,0)) = 3 \\ r((2,1)) = 2 \\ r((2,2)) = 1 \end{cases} \Rightarrow \begin{cases} (2,0) \text{ is given mode } 1 \\ (2,1) \text{ is given mode } 2 \\ (2,2) \text{ is given mode } 3 \end{cases}$$

so that $\sigma = \sigma_1\sigma_1\sigma_2\sigma_2\sigma_1\sigma_2\sigma_2\sigma_1\sigma_2\sigma_1\sigma_1\sigma_1\sigma_3$ is a Minimum Entropy AIS for $S$. And because $r((0,0)) = r((0,2))$, $S$ has exactly two Minimum Entropy AIS. The other one is obtained when $(0,2)$ is given mode 1 and $(0,0)$ is given mode 2: we get $\sigma = \sigma_1\sigma_2\sigma_2\sigma_1\sigma_1\sigma_2\sigma_2\sigma_1\sigma_1\sigma_2\sigma_1\sigma_1\sigma_3$.

**Lemma 1:** *Given an input-output sequence $S = (y(1), u(1)) \ldots ((y(n), u(n))$ and a corresponding AIS $\sigma = \sigma_{l(1)} \ldots \sigma_{l(n)}$. Assume $\sigma_i, \sigma_j \in \sigma$ are such that $\lambda_i \leq \lambda_j$ and let $\rho_j < \rho_i \leq \lambda_i$. Then, by changing $\rho_i$ of the $\sigma_i$s in $\sigma$ to $\sigma_j$s and by changing $\rho_j$ of the $\sigma_j$s in $\sigma$ to $\sigma_i$s, the entropy will be strictly reduced.*

*Proof:* the difference in entropy[3] between the sequence $\sigma$ and the modified sequence $\sigma'$ is:

$$\begin{aligned} H(\sigma) - H(\sigma') &= -\frac{\lambda_i}{n}\ln(\frac{\lambda_i}{n}) + \frac{\lambda_i - \rho_i + \rho_j}{n}\ln(\frac{\lambda_i - \rho_i + \rho_j}{n}) \\ &\quad - \frac{\lambda_j}{n}\ln(\frac{\lambda_j}{n}) + \frac{\lambda_j - \rho_j + \rho_i}{n}\ln(\frac{\lambda_j - \rho_j + \rho_i}{n}) \\ &= f(\frac{\lambda_i}{n}) - f(\frac{\lambda_j + \rho_i - \rho_j}{n}) \end{aligned}$$

where $f(x) = -x\ln(x) + (x - \frac{\rho_i - \rho_j}{n})\ln(x - \frac{\rho_i - \rho_j}{n})$.
$\forall x, f'(x) = \ln\left(\frac{x - \frac{\rho_i - \rho_j}{n}}{x}\right) < 0$ since $\rho_i > \rho_j$.
Thus,

$$\left\{ \begin{array}{l} \lambda_i \leq \lambda_j \\ \rho_i > \rho_j \end{array} \right. \Rightarrow \frac{\lambda_i}{n} < \frac{\lambda_j + \rho_i - \rho_j}{n} \Rightarrow H(\sigma) - H(\sigma') > 0,$$

which concludes the proof. ■

**Theorem:** *Among all the AIS of a given input-output string S, the Minimum Entropy AIS is the one that minimizes the entropy.*

*Proof:* The number of AIS is finite. Thus the entropy reaches a minimum on the set of AIS. We will prove by contradiction that this minimum is reached for the Minimum Entropy AIS.
Assume that there exists an AIS that is not a Minimum Entropy AIS for which the minimum entropy is reached. For such a sequence, assume there exists a pair $(y, u)$ that is associated with more than one mode, i.e. $\exists(p, q)$ such that $(y, u) = (y(p), u(p)) = (y(q), u(q))$ and $l(p) \neq l(q)$. Without loss of generality, let $\lambda_{l(p)} \leq \lambda_{l(q)}$. We can always change the mode associated with $(y(p), u(p))$ from mode $\sigma_{l(p)}$ to mode $\sigma_{l(q)}$. The result is an AIS with strictly lower entropy, according to Lemma 1, with $\lambda_i = \lambda_l(p)$, $\lambda_j = \lambda_l(q), \rho_i = 1, \rho_j = 0$. This would contradict the fact that this sequence gives the minimum entropy over the set of AIS. Thus every occurrence of the pair $(y, u)$ must be associated with the same mode.
This being known, the only way in which an AIS satisfying this property can avoid being a Minimum Entropy AIS is the existence of two modes $\sigma_i$ and $\sigma_j$ such that

$$\left\{ \begin{array}{l} \lambda_i \leq \lambda_j \\ \exists(y, u_k, u_l) \text{ such that } \left\{ \begin{array}{l} (y, u_k) \text{ is given mode } i \\ (y, u_l) \text{ is given mode } j \\ r((y, u_k)) > r((y, u_l)) \end{array} \right. \end{array} \right.$$

If we switch the modes associated with $(y, u_k)$ and $(y, u_l)$ i.e. if we associate $(y, u_k)$ with mode $\sigma_j$ and $(y, u_l)$ with mode $\sigma_i$, then, according to Lemma 1, with $\lambda_i \leq \lambda_j, \rho_i = r((y, u_k)), \rho_j = r((y, u_l)) < \rho_i$, the resulting AIS will have a strictly lower entropy, which is a contradiction. ■

We have thus produced two methods that allow us to reduce the complexity of an AIS, either by changing the interrupt structure or by carefully designing the mode that a given input-output pair is associated with. In the following section, we will see an example where these two methods are combined in the sense that a Minimum Entropy AIS is used as a basis for producing a low-complexity SIS.

## V. WHAT ARE THE ANTS DOING?

In this section we consider an example where ten ants *(Aphaenogaster cockerelli)* are placed in a tank with a camera mounted on top, as seen in Figure 1. A 52 second movie is shot from which the Cartesian coordinates, $x$ and $y$, and the orientation, $\theta$, of every ant is calculated every 33ms using a vision-based tracking software. This experimental setup is provided by Tucker Balch and Frank Dellaert at the Georgia Institute of Technology Borg Lab[4] [10].

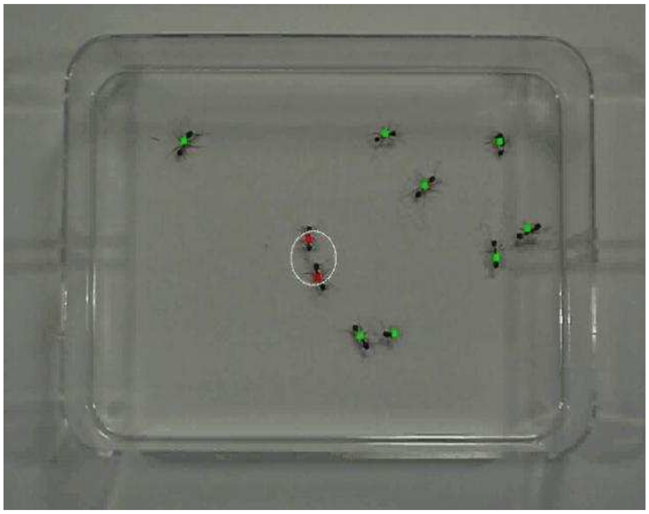

Fig. 1. Ten ants are moving around in a tank. The circle around two ants means that they are ”docking”, or exchanging information.

From this experimental data, an input-output string is constructed for each ant $i$ as follows : at each sample time $k$,

- the input $u(k)$ is given by $(u_1(k), u_2(k))$ where $u_1(k)$ is the quantized angular velocity and $u_2(k)$ is the quantized translational velocity of the ant $i$ at time $k$.
- the output $y(k)$ is given by $(y_1(k), y_2(k), y_3(k))$ where $y_1(k)$ is the quantized angle to the closest obstacle, $y_2(k)$ is the quantized distance to the closest obstacle, and $y_3(k)$ is the quantized angle to the closest goal of the ant $i$ at time $k$.

An *obstacle* is either a point on the tank wall or an *already visited* ant within the visual scope of ant $i$, and a *goal* is an ant that has not been visited recently.

In this example, we choose to quantize $u_1(k), u_2(k), y_1(k), y_2(k)$ and $y_3(k)$ using 8 possible

[3] Note that, for convenience when taking derivative, we will use $\ln$ instead of $\log_2$ in the expression of entropy. Entropy is thus expressed in nats instead of bits.

[4] http://borg.cc.gatech.edu

TABLE I

| ant# | $\|\sigma\|$ |  | $M(\sigma)$ |  | $\mathcal{H}^e(\sigma)$ |  | $\mathcal{S}^e(\sigma)$ |  |
|---|---|---|---|---|---|---|---|---|
|  | $\sigma_1$ | $\sigma_2$ | $\sigma_1$ | $\sigma_2$ | $\sigma_1$ | $\sigma_2$ | $\sigma_1$ | $\sigma_2$ |
| 1 | 21* | 57 | 21 | 5* | 4.4 | 1.4 | 92 | 82 |
| 2 | 34* | 66 | 34 | 5* | 5.1 | 1.5 | 172 | 99 |
| 3 | 25* | 68 | 25 | 6* | 4.6 | 2.0 | 116 | 139 |
| 4 | 33* | 64 | 33 | 6* | 5.0 | 1.8 | 166 | 116 |
| 5 | 20* | 65 | 20 | 6* | 4.3 | 1.9 | 86 | 121 |
| 6 | 26* | 73 | 26 | 6* | 4.7 | 1.8 | 122 | 133 |
| 7 | 33* | 71 | 33 | 6* | 5.0 | 2.0 | 166 | 145 |
| 8 | 19* | 74 | 19 | 7* | 4.2 | 2.2 | 80 | 166 |
| 9 | 25* | 71 | 25 | 10* | 4.6 | 2.4 | 116 | 169 |
| 10 | 23* | 60 | 23 | 4* | 4.5 | 1.7 | 104 | 102 |

values for each. Thus $u(k)$ and $y(k)$ can respectively take $64$ and $512$ different values. For each ant, a mode sequence $\sigma_1$ with the shortest length , in the sense of [2], and the SIS associated with a Minimum Entropy AIS $\sigma_2$ have been computed from the input-output string of length $n = 106$. Results including string length, number of distinct modes, entropy and specification complexity of these two sequences for each of the ten ants are given in Table I.

In Table I, results marked with a star are optimal. For $\sigma_1$, it is the length $|\sigma|$ that is minimized and for $\sigma_2$, it is the number of distinct modes $M(\sigma)$. It should however be noted that the length of $\sigma_2$ is in fact less than $n = 106$ as the mode sequence is not an AIS but a SIS. The minimum length sequence $\sigma_1$ has been constructed using the dynamic programming algorithm given in [2], in which every element of the mode sequence is a new mode. Consequently, $|\sigma_1| = M(\sigma_1)$.Moreover, the entropy of $\sigma_1$ is exactly equal to $\log_2(|\sigma_1|)$ as every mode is used only once in the sequence. The entropy of $\sigma_2$ is always smaller because the number of distinct modes is minimized and the modes are not equally recurrent in $\sigma_2$. Among all SIS calculated from an AIS, $\sigma_2$ has a low entropy because it is the SIS calculated from the Minimum Entropy AIS[5].
Finally, the specification complexity is smaller with $\sigma_1$ for five of the ten ants, and smaller with $\sigma_2$ for the five others. On the average, there is a little advantage for $\sigma_2$, with a total of 1152 bits compared to 1220 bits for $\sigma_1$.
An efficient way to ensure a low complexity coding would be to estimate both sequences for each ant and pick the one with lowest specification complexity. In our example, the total number of bits needed to encode the ten mode sequences using this coding strategy is 1064 bits.

## VI. CONCLUSIONS

In this paper, we present a numerically tractable solution to the problem of recovering modes from empirical data. Given a string of input-output pairs, the string with the smallest number of distinct modes that is consistent with the data is characterized algorithmically through the notion of an Always Interrupt Sequence. This has implications for how to generate multi-modal control laws by observing real systems, but also for the way the control programs should be coded. The algorithms for obtaining AIS and slightly modified derivatives of such strings can be thought of as providing a description of what modes are useful for solving a particular task, from which an empirical probability distribution over the set of modes can be obtained. This probability distribution can be put to work when coding the control programs, since a more common mode should be coded using fewer bits than an uncommon one. This work has thus a number of potential applications from teleoperated robotics, control over communication constrained networks, to minimum attention control.

### *Acknowledgments*

The authors are grateful to Tucker Balch and Frank Dellaert for providing access to their bio-tracking data.

## REFERENCES

[1] K.J. Åström and B.M. Bernhardsson. Comparison of Riemann and Lebesgue Sampling for First Order Stochastic Systems. In *IEEE Conference on Decision and Control*, pp. 2011–2016, Las Vegas, NV, Dec. 2002.

[2] A. Austin and M. Egerstedt. Mode Reconstruction for Source Coding and Multi-Modal Control. *Hybrid Systems: Computation and Control*, Springer-Verlag, Prague, The Czech Republic, Apr. 2003.

[3] A. Bicchi, A. Marigo, and B. Piccoli. Encoding Steering Control with Symbols. *IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003.

[4] R.W. Brockett. On the Computer Control of Movement. In the *Proceedings of the 1988 IEEE Conference on Robotics and Automation*, pp. 534–540, New York, April 1988.

[5] M. Egerstedt. Motion Description Languages for Multi-Modal Control in Robotics. In *Control Problems in Robotics, Springer Tracts in Advanced Robotics* , (A. Bicchi, H. Cristensen and D. Prattichizzo Eds.), Springer-Verlag, pp. 75-90, Las Vegas, NV, Dec. 2002.

[6] M. Egerstedt and R.W. Brockett. *Feedback Can Reduce the Specification Complexity of Motor Programs*. IEEE Transactions on Automatic Control, Vol. 48, No. 2, pp. 213–223, Feb. 2003.

[7] E. Frazzoli. Explicit Solutions for Optimal Maneuver-Based Motion Planning. *IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003.

[8] D. Hristu and S. Andersson. Directed Graphs and Motion Description Languages for Robot Navigation and Control. *Proceedings of the IEEE Conference on Robotics and Automation*, May. 2002.

[9] D. Hristu-Varsakelis, M. Egerstedt, and P.S. Krishnaprasad. On The Structural Complexity of the Motion Description Language MDLe. *IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003.

[10] Z. Khan, T. Balch and F. Dellaert. *An MCMC-based Particle Filter for Tracking Multiple Interacting Targets*, Technical Report number GIT-GVU-03-35 October 2003

[11] V. Manikonda, P.S. Krishnaprasad, and J. Hendler. Languages, Behaviors, Hybrid Architectures and Motion Control. In *Mathematical Control Theory*, Eds. Willems and Baillieul, pp. 199–226, Springer-Verlag, 1998.

[12] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, World Scientific Series in Computer Science, Vol. 15, River Edge, NJ, 1989.

[13] C. E. Shannon. Mathematical Theory of Communication. *Bell Syst. Tech. J.*, Vol. 27, pp. 379-423, July and pp. 623-656, October, 1948.

[5] However, it is not necessary true that this is the SIS with minimum entropy. A few examples of SIS calculated from arbitrary AIS with lower entropy than $\sigma_2$ have indeed been found.