Complexity of Control on Finite Automata

Jean-Charles Delvenne and Vincent D. Blondel

Abstract—**We consider control questions for finite automata viewed as input/output systems. In particular, we find estimates of the minimal number of states of an automaton able to control a given automaton. We prove that, on average, feedback closed-loop control automata do not have fewer states than open-loop control automata when the control objective is to steer the controlled automaton to a target state. We compare our approach to other ways of formalizing of formalizing analogous control objectives.**

Index Terms—**Complexity, feedback, finite automata, open loop, random instance.**

I. Introduction

In CONTROL theory, feedback is used to obtain a desired behavior from a system. A variety of arguments have been developed to explain why feedback is useful. Most arguments rely on the notions of uncertainty and lack of information: feedback is particularly useful when the system we want to control is not perfectly known, or when the signals between the system and its environment suffer from noise. In this paper, we study another possible interest of feedback, namely, that feedback may result in a reduction of the controller size or complexity.

We assume that the system we wish to control is perfectly known, and that we would like to steer it from a given initial state to a target state in the simplest possible way. Does the availability of feedback lead to smaller, less complex controllers? And how can complexity be measured?

The following motivating example is adapted from an example appearing in Egerstedt and Brockett [1]. Imagine that one must guide a driver from one particular crossroad to another crossroad in a city. One possible sequence of instructions may be: Turn right onto Regent St, Turn left onto Oxford St, Turn right after 3.3 km.

Another possible description would be to enumerate what direction to follow at every crossroad encountered: Straight, Straight, Right, Straight, Straight, Left, Straight, Straight, Straight, Straight, Straight, Right.

What is the difference between these two descriptions of the same path? The first description needs observations on the environment to be effective: street names, milestones. On the other hand, the second description needs more (but simpler) instructions.

We may see a city as a graph whose vertices are crossroads and edges are streets. We start from an initial vertex and at every step we follow the edge whose input label corresponds to the given input, and we read the output label attached to the edge. In the above example, the edge input labels are Straight, Left or Right, while the output label could for instance be the names of the streets. Such a graph is a particular example of an (input/output) automaton. An (input/output) automaton is a discrete-time system with a finite state space and finite input and output alphabets. At every time step an input letter is read, the state is changed accordingly, and an output letter is produced. These automata may be viewed as approximations of continuous state space systems, where space, inputs and outputs are quantized by finitely many values.

In the context of this paper, we will consider the objective of driving an automaton from an initial to a target state, and the complexity of an automaton will be identified with its number of states. An automaton can be controlled by another automaton, either in open-loop or in feedback. In order to steer an automaton from an initial to a target state, feedback control automata can in some cases be much less complex than open-loop control automata. In the main result of this paper, we prove that, even though this may be the case for particular automata, it is not true on average. More precisely, we prove in our main result that the average number of states needed to drive an n-state automaton from one state to another is

- in the order of $\ln n$ for an open-loop control automaton;
- between the order of $\ln n / \ln \ln n$ and the order of $\ln n$ for a feedback control automaton.

Thus we see that although it is not proved that open-loop and feedback have the same order of complexity, little room is left for improvement. Measuring the output is of little use on a system with a random structure. However it is easy to find examples with particular structures on which feedback leads to controllers of very low complexity. We suggest that it could also be the case for classes of automata, such as quantized linear systems for example.

Our results are in contrast with those presented by Egerstedt and Brockett [1], who also compare the complexity of open-loop versus feedback controllers, and reach the opposite conclusion. The reason for these apparently contradictory results is that the model used in [1] is very different from ours: it uses a powerful variant of automata as controllers, while we consider the simplest kind of finite automata. Moreover, their results hold under some assumptions on the structure of the automaton to be controlled, while we study random, unstructured automata.

More specifically, the results presented in [1] apply to a sophisticated variant of automata called free-running feedback automata. Essentially, these are automata that do not read an input letter at every time step, and for which an input letter contains a complete description of the feedback law to be applied to the system automaton. Thus the input alphabet may be quite large.

Manuscript received November 5, 2004; revised August 8, 2005. Recommended by Guest Editor M. Egerstedt. J.-C. Delvenne is a Research Fellow with the Belgian National Research Fund (FNRS). This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian State, Prime Minister's Office for Science, Technology, and Culture. The scientific responsibility rests with its authors.

The authors are with the Division of Applied Mathematics, the Université Catholique de Louvain, 4, B-1348 Louvain-la-Neuve, Belgium (e-mail: delvenne@inma.ucl.ac.be; blondel@inma.ucl.ac.be).

Digital Object Identifier 10.1109/TAC.2006.876948

0018-9286/$20.00 © 2006 IEEE

The complexity of control is the length of the input word realizing the objective, multiplied by the logarithm of the input alphabet cardinality. Then, for such devices, with hypotheses on the structure of the system automaton, a strategy based on a mixture of open-loop and feedback is proved to be less complex than pure open-loop.

The work presented here is also in the spirit of [2]–[4] (and many others) where systems have a continuous state space, but discrete time, discrete inputs and discrete outputs. Those articles study the amount of information needed to control the system; their motivation is typically remote control, where the information has to travel across a communication channel with finite capacity. We pursue similar goals but with quite different methods.

Before we can control a system, we must sometimes identify it. In 1956, Moore [5] proposed algorithms to identify a black-box automaton, on which we can test inputs and observe the corresponding outputs. Then Trakhtenbrot, Barzdin and Korshunov developed a probabilistic point of view and showed that automata are much easier to identify on average than in the worst case. They also designed efficient algorithms able to identify "most" automata; see [6]. We follow a similar framework, but explore one step beyond: supposing that the automaton is correctly identified, how can we control it in the least complex way?

A formalism of control on finite automata has been initiated in the '80s by Ramadge and Wonham [7]–[9], under the name of *supervisory control of discrete event systems*. Although our formalism does not fit into theirs, we briefly describe it for the sake of comparison.

The automaton to be controlled, called the *plant*, is ruled by a partial transition function. Every transition is labeled by one symbol from an alphabet Σ. This symbol can be seen as both the input and the output symbol of the transition (while we allow two different symbols for input and output in our model). This automaton is viewed as generating a language, which is the set of all sequences of symbols output by all possible sequences of transitions from the initial state. This language is non-trivial because the transition function is not defined everywhere.

The controller, called *supervisor*, takes as input a symbol of the alphabet Σ (output by the last transition of the plant), then undergoes a partially defined transition, and outputs an *enabling rule*, which is a subset of symbols allowed for the next transition of the plant. The plant then chooses nondeterministically one of those allowed symbols and performs the corresponding transition, if defined.

A typical goal of supervisory control is, given a plant and a language over the alphabet of the plant, to design a supervisor that forces the plant to generate exactly the given language. Conditions for such a supervisor to exist and algorithms to construct it can be devised.

The difference with our formalism lies both in the precise nature of the objects considered and the goal we are pursuing. The objects we consider are automata with both an input and an output alphabet, and the controller is an object of the very same nature as the system to be controlled. Our goal is to make the automaton reach a certain state, not to generate a certain language.

The theory of supervisory control has been extended to other kinds of discrete event systems (e.g., Petri nets); see [10] for more detail.

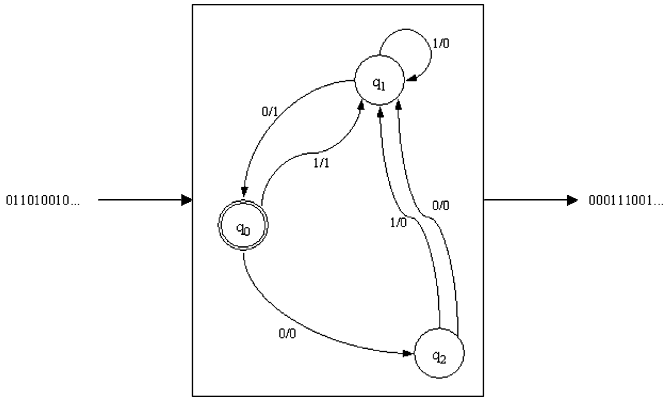

Fig. 1. Example of automaton. The vertices are the states, and the edges have labels of the form x/y, where x is an input symbol and y an output symbol. The initial state is q_0. Upon the input of the word 011010010, the automaton passes through the successive states q_0, q_2, q_1, q_1, q_0, q_1, q_0, q_2, q_1, q_0 and outputs the word 000111001.

Our main theorem is stated in Section II, along with preliminary definitions. The theorem is proved in Sections III, IV, V and VI. Conclusions and ideas for future work are presented in the last section.

II. Formulation and Results

In this section, we give precise definitions of automaton, control, and complexity of control. Then we describe our main result.

Definition: An *automaton* is a sextuple $(Q, q, X, Y, \delta, \gamma)$, where

- Q is a finite set, the elements of which are called the *states*;
- X is a finite set, called the *input alphabet*;
- $q \in Q$ is the *initial state*;
- Y is a finite set, called the *output alphabet*;
- $\delta \in Q^{Q\times X}$ is the *transition function*;
- $\gamma \in Y^{Q\times X}$ is the *output function*.

Such automata are also called *transducers* or *Mealy machines*. We set

$$\delta(q, x_0x_1\ldots x_k) = \delta(\delta(\ldots\delta(\delta(q,x_0),x_1),\ldots),x_k)$$

for any $x_0x_1\ldots x_k$ in X^*. (Recall that X^* denotes the set of finite words on the alphabet X.)

An automaton can be seen as an input/output dynamical system. Given an input word $x_0x_1\ldots x_n$, the automaton starts from the initial state q, moves to state $q' = \delta(q,x_0)$ and emits the output letter $y_0 = \gamma(q,x_0)$. Then it reads the input letter x_1, moves to state $q'' = \delta(q',x_1)$ and emits the output letter $y_1 = \delta(q',x_1)$, etc. Finally, the output word corresponding to $x_0x_1\ldots x_n$ is given by $y_0y_1\ldots y_n$ (see Fig. 1 for an example).

If X has only one symbol then we say that the automaton is *inputless*, because the input contains no information.

Note that we only consider deterministic automata. This implies that the dynamics of the system is perfectly known, as well as the initial state and the output function. We emphasize that our aim is to compare feedback and open-loop in terms of complexity, not in terms of robustness with respect to noise or uncertainty. We leave the case when noise, lack of information or nondeterminism is present for future research.

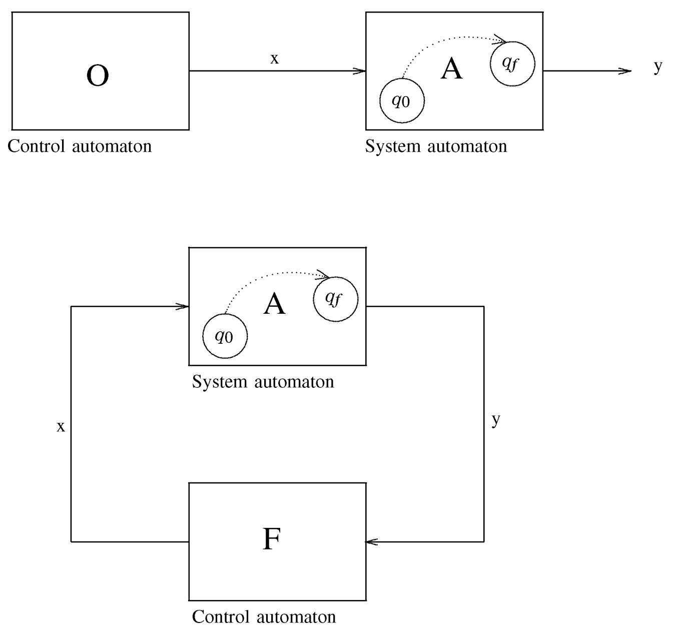

Fig. 2. We want to drive the system automaton A from q_0 to q_f. (Top) Open-loop strategy. Note that no input has been drawn towards O, since only one input word is possible, up to the length. (Bottom) Feedback strategy. The control automaton F is supposed "to play first": the first value of $x_0 \in X$ must be given as input to A, leading to an output $y_0 \in Y$, leading to a new output x_1 of F, etc.

Suppose we have an automaton and we would like to drive it from one state to another state, by feeding the automaton with an appropriate word. This is what we call the reachability problem.

Definition 2: The reachability problem is as follows.

- Instance: An automaton A (called the *system automaton*) and a state of A (the *target state*).
- Problem: Find a word (the *control word*) of X^* that makes A go from the initial state through the target state at least once.

The control word can be chosen to be the output of another automaton. We compare two kinds of strategies in order to solve a reachability problem: open-loop and feedback.

- *The open-loop strategy:* An inputless automaton O (the *control* automaton) is connected to the input of the target automaton, as indicated on the top of Fig. 2.
- *The feedback strategy:* An automaton F (the *control* automaton) is connected in feedback (bottom of Fig. 2). We can see it as a game played between the control automaton and the system automaton; the goal of the former is to force the latter to enter the target state. The game-theoretic interpretation is explored, in a larger context, in Chapter II of [6]. Let us mention a detail: for the process to be completely specified, the first input feeding the system automaton must be specified, too.

Thus, an *open-loop solution* for an instance of the reachability problem is an inputless automaton that makes the system automaton evolve and reach the target state, when connected in open-loop as described above. A *feedback solution* is composed of a control automaton and a symbol of its output alphabet, that makes the system automaton reach the target state when the control automaton is connected in feedback with the system automaton and gives the specified symbol as first output.

Our goal is to find for both strategies an estimate of the number of states of the control automaton needed to steer a given system automaton A. We would like also to know whether open-loop control is more complex than feedback control. In other words, we ask the following question: does the possibility of measurement on the system lead to less complex controllers?

For the sake of simplicity, we will focus on automata with binary input and binary output alphabets. However our results have immediate extension to alphabets of arbitrary fixed cardinality.

Given a solvable instance of the reachability problem, we define the *open-loop complexity* of the instance as the number of states of the smallest open-loop solution. The *feedback complexity* of the instance is the number of states of the smallest feedback solution.

Why to define complexity as the number of states? Since the input and output alphabets have fixed sizes, the number of states is enough to determine how complex an automaton is. Moreover, the number of states is shown later in this section to be approximately equal to the number of bits needed to give a full description of the automaton.

This "complexity" is not to be confused with computational complexity, which is the least amount of time or memory needed by a computer to answer a problem. Here checking solvability of a given instance of the reachability problem can be done in polynomial time, since it is enough to check the existence of a path from the initial state to the target state. Let us consider the problem to check whether the open-loop (or feedback) complexity of a given instance of the reachability problem is smaller than a given integer p. This problem is in the class NP, since checking that a given control automaton of less than p states is an open-loop (or feedback) solution can be done in polynomial time. We do not know how much time is needed to compute the open-loop (or feedback) complexity of a given instance of the reachability problem. A brute force approach, trying all possible controllers of increasing complexity to find one that solves the problem, takes a more than exponential time in the worst case. Indeed, the complexity of an n-state instance is at most n, and there are more than exponentially many automata of size n, as shown below.

Note that if we allow the sizes of the alphabets to vary with n, then the conclusions might change as well. For instance, if the output alphabet is the set of states of the system automaton ($Y = Q$), with the identity as output function ($\gamma(y) = y$), then for any solvable instance of the reachability problem there is a one-state feedback solution. Indeed, take a path linking the initial state to the target state. The feedback automaton gives, for any state on this path, the next input in order to remain on the path. This example suggests the existence of a trade-off between the size of the output alphabet and the number of states of the control automaton. Note nevertheless that if the sizes of the alphabets are not fixed, a better measure of complexity of the control automaton should take into account the sizes of the alphabets as well as the number of states.

A possible motivation for considering only fixed size alphabets, beyond simplicity, is the situation where the finite alpha-

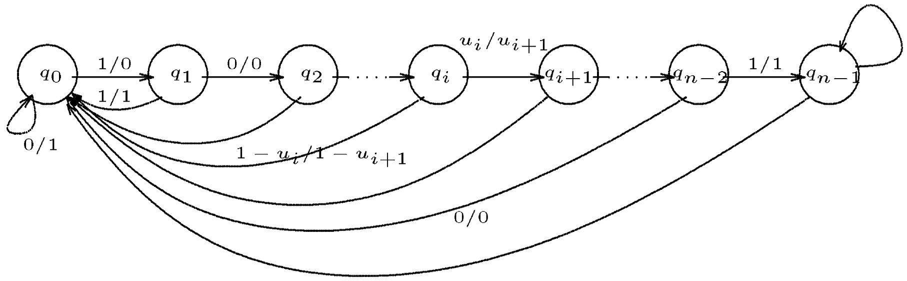

Fig. 3. An example of automaton for which the feedback strategy is much less complex then open-loop. The initial state is q_0, and we want to reach state q_{n-1}. The word $x_0x_1x_2\dots x_{n-2}$ is not eventually periodic. In the open-loop strategy, an $n-1$-state control automaton is needed. In the feedback strategy, a zero-state automaton is enough.

bets model the fact that data are transmitted through communication channels, imposing a finite flow of information, no matter how complex the system and the controller are.

The following example shows that feedback can be much less complex than open-loop.

Example 1: We want to solve the reachability problem for the n-state automaton represented on Fig. 3 and the target state q_{n-1}. We assume that the word $x_0x_1\dots x_{n-2}$ is not eventually periodic, except trivially. A word is *eventually periodic* if it is of the form uv^k, where u and v are subwords and k is a positive integer. An inputless automaton producing this word as output must therefore have at least $n-1$ states. And the complexity of the open-loop strategy for this instance is $n-1$.

But the particular form of the output function allows a one-state automaton put in feedback to solve the problem. Actually, even a zero-state automaton suffices! It is enough to connect the output of the automaton directly to the input.

As announced in the introduction, the gap of complexity between open-loop and feedback is quite small on average.

To make this statement precise, we must endow the set of n-state automata with a probability measure.

Let us compute the number of different n-state automata with input and output alphabets of cardinalities x and y respectively. From every state x edges must be drawn and for each of these there are n possible end states and y possible outputs. This leads to $(yn)^{xn}$ possible graphs. For any of them there is a choice of n possible initial states, and so there are finally $n(yn)^{xn}$ different automata. For simplicity, we suppose that the set of states is always $\{0, 1, \dots, n-1\}$. Note that giving a full description of an automaton takes $xn\log_2 yn + \log_2 n$ bits, which is in the order of $n\log_2 n$ when x and y fixed. So the number n of states is close to the amount of information needed to describe an automaton.

Of course, many different automata are *isomorphic*, i.e., they are identical up to a permutation of the states. And even more automata define the same input/output function on words.

However, keeping the formalism of [6], we shall do averages on the class of $n(yn)^{xn}$ different automata. For that purpose we endow this set with uniform probability measure. A random n-state instance of the reachability problem is composed of a random n-state automaton and a random state chosen uniformly among $\{0, 1, \dots, n-1\}$. From now we will also restrict ourselves to the case $x = y = 2$, as already mentioned.

Why the uniform distribution? This distribution reflects the fact that we have made no assumption on the origin or properties of the system automaton. The choice of the uniform distribution is consistent with the so-called "maximum entropy principle", stating that an a priori probability distribution on a family of objects is best chosen as the less informative distribution, given the constraints that we know to be satisfied. Here "less informative" is technically defined as "maximizing the Shannon entropy" see [11]. As we have no a priori knowledge on the system automaton, the uniform distribution is certainly the less informative. Choosing a simple distribution also makes the problem tractable. The uniform distribution is the one considered in [6].

Of course, we might argue that "real-life" systems are not all equally frequent or equally interesting, and most real-life situations would lead to a non-uniform distribution on system automata. For instance, the automaton might result from the quantization of a linear system. Depending on the method of quantization and the distribution of probability chosen for linear systems, we are likely to get a non-uniform distribution on finite automata. More generally, the quantization of any continuous discrete-time system should reflect the fact that "close" states have "close" images for the same input. Here again we expect that not all automata are equally represented. Similarly, if we quantize a continuous-time system with a small step of time, then the image of a state of the automaton is likely to be a "close" state.

When we say that a statement is true for "almost all n-state automata", we mean that it is true for "a proportion of n-state automata that tends towards 1 as n increases".

A *solvable* n-state instance of the reachability problem is made of an n-state system automaton and a target state that is reachable from the initial state. A random solvable instance is picked up by choosing at random an n-state automaton, and choosing at random a reachable state of the automaton. We now state the main result of this paper.

Theorem 1: Consider the uniform probability measure on n-state automata.

There are constants c_1, c_2 such that for any large enough n, the expected number of states of the smallest open-loop control automaton that solves a solvable n-state instance reachability problem is between $c_1\ln n$ and $c_2\ln n$.

There is a constant c_0 such that for any large enough n, the expected number of states of the smallest feedback control automaton that solves a solvable n-state instance reachability problem is between $c_0\ln n/\ln\ln n$ and $c_2\ln n$.

In other words, the expected complexity of open-loop is in $\Theta(\ln n)$, while the expected complexity of feedback is in

$\Omega(\ln n/\ln\ln n)$ and $\mathcal{O}(\ln n)$. Recall that $f(n) = \mathcal{O}(g(n))$, or $g(n) = \Omega(f(n))$, means that $|f(n)| \leq c|g(n)|$ for some $c > 0$ and all large enough n. If $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$ then we write $f(n) = \Theta(g(n))$.

Sections III, IV, V and VI are devoted to the proof of Theorem 1, and may be skipped in a first reading.

III. SOLVABLE INSTANCES OF THE REACHABILITY PROBLEM

It is clear that the expected complexity can be computed only over solvable instances. The following proposition essentially says that more than one third of the instances of the reachability problem are solvable. Recall that we consider only automata with binary input and output alphabets.

Proposition 1: For any $\alpha < 1/e = 0.36\ldots$ and any large enough n, a random n-state instance of the reachability problem is solvable with probability at least α. More precisely, for almost all n-state automata at least $\lceil \alpha n \rceil$ states are reachable from the initial state.

Proof: Let $Q = \{0, 1, \ldots, n-1\}$ the set of states of a random automaton.

A subset $R \subseteq Q$ is *invariant* if, starting from a state of R, it is impossible to leave R. In other words, no edge leaves R.

We are going to prove that in almost all n-state automata, there is no invariant subset of $\lfloor \alpha n \rfloor$ states or less, for $0 < \alpha < 1/e$. As the set of states that are reachable from the initial state is an invariant set, this is enough to prove the proposition.

The probability for any set R of r states to be invariant is $(r/n)^{2r}$, since $2r$ edges start from R, and each of them arrive in R with probability r/n.

For any $R \subseteq Q$, consider the random variable I_R that is equal to 1 if the set R is invariant and 0 otherwise. The expectation of I_R is the probability of the event $I_R = 1$, which is $(r/n)^{2r}$. The probability that there is a invariant set of size at most αn is equal to

$$\mathbb{P}\left(\sum_{\substack{R\subseteq Q\\ 0<\,\text{card}(R)\leq \alpha n}} I_R > 0\right)$$

where$\mathbb{P}$ denotes the probability of an event.

For any non-negative integer-valued random variable X, Markov's inequality

$$\mathbb{P}(X > 0) \leq \mathbb{E}X$$

holds, where $\mathbb{E}$ is the expectation. Indeed, $\mathbb{E}X = \sum_{i>0} i\mathbb{P}(X = i) \geq \sum_{i>0} \mathbb{P}(X = i) = \mathbb{P}(X > 0)$.

Here we take $X = \sum_{0<\,\text{card}(R)\leq \alpha n} I_R$. The probability that there is an invariant set of size at most αn is

$$\begin{aligned}\mathbb{P}(X > 0) &\leq \mathbb{E}X\\ &= \sum_{0<\,\text{card}(R)\leq \alpha n} \mathbb{E}I_R\\ &= \sum_{r=1}^{\lfloor \alpha n\rfloor} \binom{n}{r}\left(\frac{r}{n}\right)^{2r}.\end{aligned}$$

If we prove that this last quantity converges to 0 as $n \to \infty$, then we conclude from Markov's inequality that almost all automata of size n have have no invariant set of states of size at most αn. This argument is an example of the so-called *first moment method*.

It remains to prove that $\lim_{n\to\infty} \sum_{r=1}^{\lfloor \alpha n\rfloor} \binom{n}{r}(r/n)^{2r} = 0$. For every n and for every term of the series, the following relations hold:

$$\begin{aligned}\binom{n}{r}\left(\frac{r}{n}\right)^{2r} &\leq \frac{r^{2r}}{r!n^r}\\ &\leq \frac{e^r r^r}{n^r} \leq (e\alpha)^r.\end{aligned}$$

We used the inequalities $\binom{n}{r} \leq (n^r/r!)$ and $(r^r/r!) \leq \sum_{i\geq 0} r^i/i! = e^r$, and the fact that $r \leq \alpha n$.

Let us consider for every n the function

$$f_n : r \mapsto \begin{cases} \binom{n}{r}\left(\frac{r}{n}\right)^{2r} & \text{if } 0 < r \leq \alpha n;\\ 0 & \text{otherwise.}\end{cases}$$

We want to prove that $\lim_{n\to\infty} \sum_r f_n(r) = 0$. But the following properties are verified:

- The sequence $(f_n)_n$ converges pointwise to the constant function 0, since $f_n(r) \leq (e^r r^r/n^r)$.
- Every f_n is bounded by the function $g : r \mapsto (e\alpha)^r$, and g is summable: $\sum_{r\geq 0}(e\alpha)^r = (1/(1-e\alpha))$.

From Lebesgue's dominated convergence theorem, this is enough to conclude that the sequence $(\sum_r f_n(r))_n$ converges to $\sum_r 0 = 0$. ■

We don't know how far the critical ratio $1/e$ can be improved. It might be the case that all states are reachable for almost all automata. As far as we know this is an open problem.

IV. AN UPPER BBOUND ON COMPLEXITY

If a state of an n-state automaton is reachable from the initial state, then it is so with an input word of length at most n; hence the complexity of feedback and open loop are both bounded by n. On average much more can be said.

Theorem 2: There is a constant C such that for almost all n-state automata, if a state is reachable from another state, then it is reachable with an input word of length at most $C \ln n$.

We do not prove this result here; the interested reader is referred to Theorem 5.5 in [6].

Corollary 1: There is a constant C such that for almost all n-state instances of the reachability problem, there are feedback and open loop solutions of $C \ln n$ states or less. In particular, the expected complexity of a solvable instance is in $\mathcal{O}(\ln n)$ for both strategies.

V. LOWER BOUNDS ON COMPLEXITY

Let us a take a random instance of the reachability problem of size n, i.e., a random n-state automaton (with binary input and output alphabets) with a random target state. Take a random p-state inputless automaton (with binary output alphabet). We will denote $\mathcal{P}^{ol}(n, p)$ the probability that the random p-state control automaton solves the random n-state instance of the reachability problem in open-loop.

Now take a random p-state automaton (with binary input and output alphabets). We will denote $\mathcal{P}^f(n,p)$ the probability that the random p-state control automaton solves the random n-state instance in feedback.

In this section, we compute estimates of these quantities that allow us to give lower bounds on complexity.

First of all, a technical lemma:

Lemma 1: There is a constant $c > 0$ such that for any large enough p, any $n \geq p^3$ and $k = 2p\sqrt{n \ln n}$ we have

$$\frac{n^p e^{-k}}{\left(1 - \frac{k+p}{pn}\right)^{pn-k}} \leq c\frac{1}{n}.$$

Proof: Taking the logarithm:

$$\begin{aligned}
\ln \frac{n^p e^{-k}}{\left(1 - \frac{k+p}{pn}\right)^{pn-k}} &= -k + p\ln n \\
&\quad - (pn-k)\ln\left(1 - \frac{k+p}{pn}\right) \\
&= -k + p\ln n + (pn-k)\frac{k+p}{pn} \\
&\quad + \frac{1}{2}(pn-k)\left(\frac{k+p}{pn}\right)^2 \\
&\quad + \mathcal{O}\left((pn-k)\left(\frac{k+p}{pn}\right)^3\right).
\end{aligned}$$

We used the expansion $\ln(1+x) = x - x^2/2 + \mathcal{O}(x^3)$. A little calculation yields:

$$\begin{aligned}
\ln \frac{n^p e^{-k}}{\left(1 - \frac{k+p}{pn}\right)^{pn-k}} &= -k + p\ln n + k + p - \frac{k(k+p)}{pn} \\
&\quad + \frac{1}{2}\frac{(k+p)^2}{pn} - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} \\
&\quad + \mathcal{O}\left(pn\frac{k^3}{(pn)^3}\right) \\
&= p(\ln n + 1) + \frac{(p-k)(k+p)}{2pn} \\
&\quad - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\left(\frac{k^3}{(pn)^2}\right) \\
&= p(\ln n + 1) + \frac{p^2}{2pn} - \frac{k^2}{2pn} \\
&\quad - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} + \mathcal{O}\left(\frac{p\ln^{3/2} n}{\sqrt{n}}\right) \\
&= -p(\ln n - 1) + \frac{p}{2n} - \frac{1}{2}\frac{k(k+p)^2}{(pn)^2} \\
&\quad + \mathcal{O}\left(\frac{p\ln^{3/2} n}{\sqrt{n}}\right) \\
&\leq -p(\ln n - 1) + \frac{p}{2n} + \mathcal{O}\left(\frac{p\ln^{3/2} n}{\sqrt{n}}\right).
\end{aligned}$$

The last two terms of the last line are bounded by a constant, provided that $n \geq p^3$. We can also suppose $p \geq 2$. Thus, by exponentiation:

$$\frac{e^{-k}}{\left(1 - \frac{k+p}{pn}\right)^{pn-k}} \leq c\frac{1}{n}$$

for some $c > 0$. ■

Finally, a "Stirling-like" formula, taken from [12].

Lemma 2: For all $n \geq 1$,

$$e\left(\frac{n}{e}\right)^n \leq n! \leq en\left(\frac{n}{e}\right)^n.$$

Proof: From $\lfloor x \rfloor \leq x \leq \lceil x \rceil$, we have $\ln\lfloor x \rfloor \leq \ln x \leq \ln\lceil x \rceil$. Integrating between 1 and n, we find:

$$\sum_{i=1}^{n-1} \ln i \leq [x\ln x - x]_1^n \leq \sum_{i=2}^{n} \ln i$$

which is equivalent to

$$\ln(n-1)! \leq n\ln n - n + 1 \leq \ln n!.$$

The result follows almost immediately by exponentiation. ■

A. A Lower Bound for Open-Loop Complexity

We may obtain the following bound on $\mathcal{P}^{ol}(n,p)$:

Proposition 2: For any large enough p and any $n \geq p^3$, the probability $\mathcal{P}^{ol}(n,p)$ for a random inputless p-state automaton to solve a random n-state instance of the reachability problem in open-loop satisfies the following bound:

$$\mathcal{P}^{ol}(n,p) \leq 3p\sqrt{\frac{\ln n}{n}}.$$

Proof: We fix an arbitrary inputless control automaton of p states, and plug the output to the input of a random n-state automaton.

In a first step we give an upper bound on the probability for the path described in the random target automaton to pass through at least k of the n states. In a second step we use Lemma 1 to compute a bound on $\mathcal{P}^{ol}(n,p)$.

Step 1: To explore at least $k+1$ different states, we must explore at least k different edges. Each time a new edge is being explored, the end state of the edge is randomly chosen among the n states. If this new edge occurs after l steps (i.e., when the path already drawn has length l), and if less than k edges are explored at that moment, then at most $n - \lfloor l/p \rfloor$ states are allowed for the end state of the edge (in order not to loop before having seen k edges). This is because if the pair (state of the control automaton, state of the system automaton) is the same at two steps, then the system enters a loop, and because the succession of states of an inputless automaton is eventually periodic with period at most p. Thus at least $\lfloor l/p \rfloor$ states are "forbidden".

Moreover, the ith new edge is discovered after a length $l \geq i$. Thus the probability to explore at least k edges is less than the following product of k factors:

$$\overbrace{\frac{n}{n}\cdots\frac{n}{n}}^{p}\overbrace{\frac{n-1}{n}\cdots\frac{n-1}{n}}^{p}\cdots\cdots\overbrace{\frac{n-\lfloor k/p\rfloor}{n}\cdots\frac{n-\lfloor k/p\rfloor}{n}}^{\leq p} \tag{1}$$

where each factor—except maybe the last—is repeated p times. Dropping the last factor, this product is smaller than:

$$\begin{aligned}\left(\frac{n!}{(n-\lfloor k/p\rfloor)!n^{\lfloor k/p\rfloor}}\right)^p &\leq \left(n\frac{e^{-\lfloor k/p\rfloor}}{\left(1-\frac{1}{n}\left\lfloor\frac{k}{p}\right\rfloor\right)^{n-\lfloor k/p\rfloor}}\right)^p\\ &\leq n^p\frac{e^{-k'}}{\left(1-\frac{k'+p}{pn}\right)^{pn-k'}}\end{aligned}$$

where $k' = k-p$ (implying $k'/p < \lfloor k/p\rfloor \leq k'/p+1$). Lemma 2 has been used to derive the first inequality.

Step 2: Let k' be equal to $2p\sqrt{n\ln n}$. Then Lemma 1 ensures that

$$\frac{e^{-k'}}{\left(1-\frac{k'+p}{pn}\right)^{pn-k'}} \leq c\frac{1}{n}$$

for some $c > 0$, if p is large enough and $n \geq p^3$. Thus the probability to explore more than $2p\sqrt{n\ln n}+p+1$ states is at most c/n. The expected number of states visited by the control automaton in the system automaton is therefore at most

$$\left(1-c\frac{1}{n}\right)(2p\sqrt{n\ln n}+p+1)+c\frac{1}{n}n \leq 3p\sqrt{n\ln n}$$

for p large enough and $n \geq p^3$. As any of the n states may be chosen as the target state with equal probability, we conclude that $\mathcal{P}^{ol}(n,p) \leq 3p\sqrt{\ln n/n}$ for p large enough and $n \geq p^3$. ■

In fact, the proof shows that the result holds not only for a random control automaton but for any fixed control automaton.

We now derive a lower bound on open-loop complexity.

Proposition 3: There is a c such that for any large enough n the expected complexity of the open-loop strategy for a random n-state solvable instance of the reachability problem is at least $c\ln n$.

Proof: The behavior of an inputless automaton is completely characterized by the infinite word it produces as output. This output word is eventually periodic.

The number of inputless p-state automata generating different output sequences is $p2^p$. Indeed, p is the sum of lengths of the non-periodic part and the period, and there are p possibilities for the length of the period. This also includes the control automata of less than p states.

So, using Proposition 2, the probability that an n-state instance of the reachability problem is solved by at least one p-state open-loop automaton is (provided that $n \geq p^3$ and p is large enough) bounded above by

$$p2^p3p\sqrt{\frac{\ln}{n}}.$$

Assume that this quantity is lower than 1/6:

$$2^p3p\sqrt{\frac{\ln n}{n}} \leq \frac{1}{6}$$

which may be written as

$$3p^22^p \leq \frac{1}{6}\sqrt{\frac{n}{\ln n}}.$$

This is satisfied if

$$p \leq c_0\ln n$$

for some $c_0 > 0$. This condition is stronger than $n \geq p^3$, for p large enough.

Thanks to Proposition 1, we know that for n large enough, at least one third of the n-state instances of the reachability problem are solvable. If $p \leq c_0\ln n$, at most one sixth of the n-state instances are solved by open-loop control automata of p states or less.

Hence at least one half of the n-state solvable instances are not solved by any open-loop control automaton of at most p states. Consequently, the average complexity of an n-state solvable instance is at least $(c_0/2)\ln n$. So the result follows with $c = c_0/2$. ■

B. A Lower Bound for Feedback Complexity

We now adapt the arguments of the preceding subsection to the case of feedback.

Proposition 4: For any large enough p and for any $n \geq (2p)^3$, the probability $\mathcal{P}^f(n,p)$ for a random p-state automaton to solve in feedback a random n-state instance of the reachability problem satisfies the following bound:

$$\mathcal{P}^f(n,p) \leq 6p\sqrt{\frac{\ln n}{n}}.$$

Proof: First we fix an arbitrary feedback control automaton F.

The main idea of the proof is the same as in the proof of Proposition 2. There we used the fact that if the system automaton is twice in the same state while the open-loop control automaton is also in the same state, then the whole system falls into a loop and will not discover new states. The case of the feedback strategy is slightly more complicated. Indeed, knowing the states of the control and system automata is not sufficient to determine the evolution of the system.

Suppose that the system automaton A "has just played" (remember the game-theoretic terminology) and is now in state q, emitting a symbol y. The control automaton F is about to play and is in state s. Now we have enough information to determine the evolution of the system. If during two different steps, A is in state q, outputs the symbol y and F is in state s, then the whole system enters a loop. Since y can take two values, the system automaton and the control automaton can only be twice in the same states if we still hope to discover new states of A. Heuristically, we thus expect the same result as in Proposition 2, except that that p becomes $2p$. The rest of the proof rigorously establishes this fact.

Let the states of F be denoted by $s_1, \dots, s_p$. At every step of time, when F has just played, and for every state s_i, we define

- a_i as the number of states of A that have been reached exactly once while F was in state s_i;
- b_i as the number of states of A that have been reached exactly twice while F was in state s_i;
- r_i as $n-a_i-b_i$; as long as the whole system has not entered a loop, r_i is the number of states of A that have never been reached while F was in s_i.

At the beginning of the process, $r_i = n$ and $a_i = b_i = 0$ for all states s_i. At every step of time, if F is in s_i, then a_i, b_i, r_i must be updated. If a state of A is reached for the first time while F is in s_i, then a_i is increased ("$a_i := a_i + 1$") and r_i is decreased. If a state of A is reached for the second time while F is in s_i, then a_i is decreased and b_i is increased. If a state of A is reached for the third time or more, then we will not discover any new state of A anymore.

In the first two cases, the quantity $(a_i/2) + r_i$ decreases by 1/2.

We now want to estimate the probability that at least $k+1$ states of the system automaton are visited when F is connected in feedback to it. To explore at least $k+1$ different states of the system automaton, we must explore at least k different edges. Every time a new edge is being explored, the end state of the edge is chosen randomly among the n states. Suppose that after l steps (i.e., when the path already drawn has length l), less than k edges have been reached, the system automaton is in state $\tilde{q}$, F has just sent the output u and the edge starting from $\tilde{q}$ labelled by u has never been used: we must choose an end state $q = \delta_A(\tilde{q}, u)$ for this new edge, as well as an output value $y = \gamma_A(\tilde{q}, u)$ for this edge.

Suppose in addition that the current state of F is s_i. The end state of the new edge is to be chosen among three kinds of states.

- Those that have been reached once while F was in state s_i. There are a_i of them. If one of these states is chosen, then the probability for the trajectory to fall into a loop is at least 1/2, because one of the two possible values for $\gamma_A(\tilde{q}, u)$ is "forbidden" (remember we do not want to loop before having explored k edges).
- Those that have been reached (exactly) twice while F was in state s_i. There are b_i of them. They may not be chosen as the end state of the edge, since the whole system would loop.
- Those that have not been explored yet when F was in state s_i. There are r_i of them. They may be chosen without restriction.

So the probability to choose a "good" end state for a new edge (i.e., not to fall into a loop) is at most $(1/2)(a_i/n) + (r_i/n)$. Remember that $(1/2)a_i + r_i$ is equal to $n - (1/2)$ (number of times that F has been in state s_i).

Call t_i the number of occurrences of s_i while exploring a new edge, i.e., the number of times that F was in s_i when we took an edge of the path for the first time. If k edges at least have been explored, then $t_1 + \cdots + t_p \geq k$.

The probability of discovering at least k new edges, conditional to the repartition into $t_1, \dots, t_p$, is thus bounded above by

$$\prod_{1\leq i\leq p} \frac{n}{n}\frac{n-\frac{1}{2}}{n}\frac{n-\frac{2}{2}}{n}\cdots\frac{n-\frac{t_i}{2}}{n}. \tag{2}$$

The probability of discovering at least k new edges is this product, averaged over all possible repartitions into $t_1, \dots, t_p$.

On the other side, a product of this form, under the constraint $t_1 + \cdots + t_p \geq k$, takes its maximum value when $t_1, \dots, t_p$ are all equal to $\lfloor k/p \rfloor$ or $\lceil k/p \rceil$, as some elementary calculus can show.

Hence the probability of discovering at least k states in A is at most

$$\left(\frac{n}{n}\frac{n-\frac{1}{2}}{n}\frac{n-\frac{2}{2}}{n}\cdots\frac{n-\left\lceil\frac{k}{p}\right\rceil\frac{1}{2}}{n}\right)^p$$

tself bounded by

$$\left(\frac{n}{n}\frac{n}{n}\frac{n-1}{n}\frac{n-1}{n}\cdots\frac{n-\left\lceil\frac{k}{2p}\right\rceil}{n}\frac{n-\left\lceil\frac{k}{2p}\right\rceil}{n}\right)^p$$

where each factor is repeated twice.

Comparing with (1), we see that nothing has changed, except that p has become $2p$. So the end of the proof remains unchanged in its principle: the conclusion is that the probability that $\mathcal{P}^f(n,p)$ is less than $6p\sqrt{\ln n/n}$. ■

Again, the proof holds not only for a random feedback control automaton but for any feedback control automaton.

Thus:

Proposition 5: There exists a constant c such that for any large enough n, the expected number of states of the smallest feedback control automaton that solves a random n-state solvable instance of the reachability problem is greater than $c(\ln n/\ln\ln n)$.

Proof: The number of different p-state automata is $p(2p)^{2p}$ and there are two possible initial bits (for the initial output of the control automaton). So, using the preceding proposition, the probability that a random n-state instance of the reachability problem is solved by at least one p-state feedback automaton is, for p large enough and $n \geq (2p)^3$, bounded above by:

$$2p(2p)^{2p}6p\sqrt{\frac{\ln n}{n}}.$$

We want this quantity to be lower than 1/6:

$$2p(2p)^{2p}6p\sqrt{\frac{\ln n}{n}} \leq \frac{1}{6}.$$

This is equivalent to

$$3(2p)^{2p+2} \le \frac{1}{6}\sqrt{\frac{n}{\ln n}}$$

or

$$(2p+2)\ln(2p) \le \frac{1}{2}(\ln n - \ln\ln n) - \ln 18.$$

This is satisfied if

$$p \le \frac{1}{7}\frac{\ln n}{\ln\ln n}$$

for n large enough. Indeed, this condition implies

$$\begin{aligned}(2p+2)\ln(2p) &\le 3p\ln(7p)\\ &\le \frac{3}{7}\frac{\ln n}{\ln\ln n}(\ln\ln n - \ln\ln\ln n)\\ &= \frac{3}{7}\left(\ln n - \frac{\ln n \ln\ln\ln n}{\ln\ln n}\right)\\ &\le \frac{3}{7}(\ln n - \ln\ln n)\\ &\le \frac{1}{2}(\ln n - \ln\ln n) - \ln 18\end{aligned}$$

for any large enough p. This condition is stronger than $n \ge (2p)^3$.

Thanks to Proposition 1, we know that for n large enough, at least one third of the n-state instances of the reachability problem are solvable. If $\ln n/\ln\ln n$, it means that at least half of the solvable n-state instances are not solved by any p-state open-loop control automaton. Hence the expected complexity of an n-state instance is at least $(1/2)(1/7)\ln n$. So the result follows with $c = 1/14$. ■

VI. PROOF OF THE MAIN THEOREM

We are now able to prove Theorem 1.

Corollary 1 of Section IV proves that the complexity for both strategies is in $\mathcal{O}(\ln n)$. Proposition 3 of Section V proves that the open-loop complexity is in $\Omega(\ln n)$. Proposition 5 of Section V proves that the complexity of feedback is in $\Omega(\ln n/\ln\ln n)$. Hence the open-loop complexity is in $\Theta(\ln n)$ and the feedback complexity is in $\Omega(\ln n/\ln\ln n)$ and in $\mathcal{O}(\ln n)$.

VII. CONCLUSIONS

We have studied finite automata as input/output systems, and have analyzed the complexity needed to control them. It follows from our results that making measurements on the output of a system is not very useful for automata with a completely random structure. Of course, feedback cannot be more complex than open-loop, and we gave an example where it is actually much less complex.

Thus, we are lead to the conclusion that feedback is useful to reduce complexity only when the automaton to be controlled has a particular structure. Future work could be devoted to finding classes of particular systems for which feedback is particularly interesting. For instance, it would be natural to see what happens if the automaton results from the quantization of a linear system. The introduction of some noise or nondeterminism in the model is also a logical direction of research.

Finally, it can be argued that we did not really address the introductory example (describing a path in a city), since in that case the size of the output alphabet (number of streets, for instance) is not fixed, but depends on the size of the automaton (number of crossroads). For what kind of dependence between sizes of input alphabet, output alphabet, and number of states does Theorem 1 remain valid? We have seen for instance that if we allow output alphabets of the size of the system automaton, then one-state feedback controllers are always enough.

ACKNOWLEDGMENT

The authors would like to thank P. Collins, A. Megretski, S. Zampieri, P. Delsarte, J. Bricmont, P. Koiran, and P. Kůrka for their comments and insights. They are also grateful to the anonymous reviewers and to the editorial staff for precious remarks and suggestions.

REFERENCES

[1] M. Egerstedt and R. Brockett, “Feedback can reduce the specification complexity of motor programs,” *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 213–223, Feb. 2003.

[2] R. Brockett and D. Liberzon, “Quantized feedback stabilization of linear systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 7, pp. 1279–1289, Jul. 2000.

[3] D. Delchamps, “Stabilizing a linear system with quantized state feedback,” *IEEE Trans. Autom. Control*, vol. 35, no. 8, pp. 916–924, Aug. 1990.

[4] F. Fagnani and S. Zampieri, “Quantized stabilization of linear systems: complexity versus performance,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1534–1548, Sep. 2004.

[5] E. Moore, “Gedanken-experiments on sequential machines,” in *Automata Studies*, C. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1956.

[6] B. Trakhtenbrot and Y. Barzdin, *Finite Automata, Behavior and Synthesis*. Amsterdam, The Netherlands: North-Holland, 1973.

[7] P. J. Ramadge and W. M. Wonham, “Supervisory control of a class of discrete-event processes,” *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[8] ——, “The control of discrete event systems,” *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[9] W. M. Wonham and P. J. Ramadge, “On the supremal controllable sublanguage of a given language,” *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 637–659, May 1987.

[10] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, ser. The Kluwer International Series in Discrete Event Dynamic Systems. Boston, MA: Kluwer, 1999, vol. 11.

[11] R. D. Rosenkrantz, Ed., *E.T. Jaynes: Papers on Probability, Statistics and Statistical Physics*. Dordrecht, The Netherlands: D. Reidel, 1983.

[12] M. Dietzfelbinger and M. Kunde, “A case against using stirling’s formula (unless you really need it),” *Bull. Eur. Assoc. Comput. Sci.*, vol. 80, pp. 153–158, 2003.

Jean-Charles Delvenne received the Ph.D. degree in applied mathematics from the Université Catholique de Louvain, Belgium.

He is currently a Postdoctoral Visitor at the California Institute of Technology, Pasadena. His research interests are mainly in systems and control theory, theoretical computer science, and information theory.

Vincent D. Blondel received the M.Sc. in pure mathematics from Imperial College, London, U.K., in 1988, and the Ph.D. degree in applied mathematics from the Université Catholique de Louvain, Belgium, in 1992.

He is a Professor of Applied Mathematics at the Université Catholique de Louvain, where he is also Head of the Department of Mathematical Engineering. He has held postdoctoral research positions at Oxford University, Oxford, U.K., the Royal Institute of Technology, Stokholm, Sweden, and INRIA Rocquencourt, Paris, France. He has been an Invited Professor at the Ecole Normale Supéerieure, Lyon, France, and at the University of Paris VII, Paris, France. During the academic year 2005–2006, he was an Invited Professor and Fulbright Scholar at the Massachusetts Institute of Technology, Cambridge. His major current research interests lie in several areas of mathematical control theory and theoretical computer science. He has published four books and about 60 scientific papers in these areas.

Dr. Blondel was a recipient of the Prize of Mathematics of the Belgian Royal Academy of Science, and has been awarded the SIAM Prize for control and systems theory.