

RESEARCH

Open Access



Towards distribution-based control of social networks

Dave McKenney* and Tony White

*Correspondence:
david.mckenney@carleton.ca
School of Computer Science,
Carleton University, 5302
Herzberg Laboratories, 1125
Colonel By Drive, Ottawa, ON
K1S5B6, Canada

Abstract

Background: Complex networks are found in many domains and the control of these networks is a research topic that continues to draw increasing attention. This paper proposes a method of network control that attempts to maintain a specified target distribution of the network state. In contrast to many existing network control research works, which focus exclusively on structural analysis of the network, this paper also accounts for user actions/behaviours within the network control problem.

Methods: This paper proposes and makes use of a novel distribution-based control method. The control approach is applied within a simulation of the real-valued voter model, which could have applications in problems such as the avoidance of consensus or extremism. The network control problem under consideration is investigated using various theoretical network types, including scale free, random, and small world.

Results: It is argued that a distribution-based control approach may be more appropriate for several types of social control problems, in which the exact state of the system is of less interest than the overall system behaviour. The preliminary results presented in this paper demonstrate that a standard reinforcement learning approach is capable of learning a control signal selection policy to prevent the network state distribution from straying far from a specified target distribution.

Conclusions: In summary, the results presented in this paper demonstrate the feasibility of a distribution-based control solution within the simulated problem. Additionally, several interesting questions arise from these results and are discussed as potential future work.

Keywords: Control, Opinions, Network control, Multi-agent systems

Background

Complex networks, including social, communication, and even financial networks are constantly increasing in prevalence. As the behaviour of these networked systems can have important consequences, interest in the development of methods to control them to either achieve a goal state or avoid undesirable states is also increasing. A significant amount of research has been dedicated to the structural analysis of complex networks, especially how structure relates to what is known as the full state controllability of a system [1]. Much of this existing work, though, has ignored the behavioural aspect of these systems and has only answered questions relating to the identification of control structures. However, it has been observed that ignoring behaviour can lead to naive control

solutions [2]. In addition to this, the control target within these works is generally a single point within the vector space model of the system state.

In many cases, especially scenarios involving crowding, flocking and consensus, the overall state and behaviour of the system is of more interest than reaching some exact network state specification. This work proposes distribution-based control as a method to control these types of systems, where we are interested in both the identification of a control node set (structural) and the generation of control signals (behavioural) to maintain some state distribution within the network. Using a distribution as a control target, we can define a more general control target within the system when compared to a point-based approach. For example, a normal distribution is used as a target in the work presented here, which could simultaneously address the problems of avoiding consensus and extremism of opinion within a social network. Another possible example would be the use of an exponential/gamma distribution of some influence measure on a social system to limit the percentage of the population that is capable of significantly influencing a large portion of others, which could help to limit the rate of change of the system's state. The essential requirement of distribution-based control is the ability to measure distance between distributions within a particular time interval.

This paper makes several contributions. First, we introduce the problem of distribution-based network control, which is an evolution of the more general network control problem (NCP) formalized by [3]. Second, we describe the general architecture for a distribution-based control system. Third, we present preliminary results demonstrating the application of distribution-based control within the real-valued voter model, which has been used previously in network control research. These findings demonstrate the feasibility of using a basic learning technique (reinforcement learning) to develop successful control strategies for the distribution-based control problem. The results also bring to light several interesting questions related to network control in general, which are identified as areas for future work.

The remainder of this paper is outlined as follows. "[Related work](#)" discusses existing research within the network control domain, especially that which is related to the full state controllability of networked systems, and identifies particular deficiencies within the existing research that this work aims to address. It also briefly discusses a previously defined problem based on the real-valued voter model, which is related to the problem studied in this work. "[Distribution-based control](#)" describes the general architecture of a distribution-based control system and discusses the formulation of the distribution control problem that is used within this work. The experimental model that we attempt to control and the learning algorithm used to develop a control policy are outlined in "[Experimental model](#)" and "[Learning a control policy](#)", respectively. Results demonstrating the efficacy of the learned control policies across three different types of theoretical networks are included in "[Results](#)". Finally, the paper concludes with a discussion of future work directions and a summary of the main conclusions in "[Future work](#)" and "[Conclusions](#)".

Related work

Network control

There is a significant amount of existing research relating to the control of complex networks. A large proportion of this work relates to the analysis of network controllability

from the perspective of full state controllability. A system, such as a complex network, is said to be fully state controllable if it is possible to move the system from any initial state x to any other possible state y in finite time [4]. The work of [1] provided an in-depth analysis of the full state controllability of linear time-invariant systems, proposing algorithms for the identification of a minimal set of control nodes. Using the structural controllability formulation of [5], [6] built upon the work of [1] by identifying structural properties that require additional control inputs. The work of [1], which was limited to directed networks, has also been generalized by [7] to produce an algorithm to identify a minimal set of control nodes within networks with arbitrary structure.

One of the main criticisms of these structural control theory works is that they do not account for individual dynamics within the system. As indicated by [8], this means that applying the structural control framework to any system in which individual dynamics are required to satisfactorily model the system would produce spurious, naive or misleading results. This problem had also been previously recognized by the work of [2], which found that including any individual dynamics within the system results in a network being controllable with only a single control input.

Another criticism of work relating to full state controllability analysis is that, in many scenarios, the requirement of full state controllability is unnecessarily strong. This is true in many network control problems, where the goal may not be to move the system between any two arbitrary states, but instead to avoid the system moving into one of a set of undesired states. As full state controllability only requires that the system can be moved between two states in finite time, there are several limitations from a practical perspective as well. The first of these limitations is that, in some applications, the system may need to be moved to/from some state in a limited amount of time. In addition to this, full state controllability does not account for potentially negative and catastrophic states that may be encountered when moving between any two states, which may have significant impacts on the performance of a control system in practice.

Finally, much of the existing network control research focuses on the structural problem of identifying which nodes to use as controllers within the network. Significantly less work has been devoted to developing algorithms for the selection of the control signals that will be used as inputs to these controllers to achieve network control. Recent work, such as that of [9] and [3], has simultaneously considered the problem of control agent selection along with the generation of control signals to achieve control of complex network systems. Including the behavioural aspect of control within these works has demonstrated that control architectures selected using algorithms proposed in previous structural analysis work do not necessarily produce the most effective controllers. In fact, [9] found that using controller sets generated using the maximum matching principle of [1] produced inferior results when compared to several other control node selection heuristics.

The real-valued θ -consensus avoidance problem

As mentioned previously, the work presented here is an extension of the more general NCP formalized by [3]. As such, the distribution-based problem addressed here, which attempts to control the state distribution of a real-valued voter model simulation (details in "[Experimental model](#)"), is also inspired from a similar NCP problem used in [9] called

the real-valued θ -consensus avoidance problem (θ -CAP_{RV}). If the state of a node v at time t in the real-valued voter model simulation is represented by $s(v, t)$, and the set of all nodes in the network is denoted by V , a controller for the θ -CAP_{RV} problem attempts to maximize the utility function, U , in Eq. 1.

$$U_{\theta\text{-CAP}_{\text{RV}}}(t) = \begin{cases} 1, & \text{if } \frac{|\sum_{v \in V} s(v, t)|}{|V|} < \theta_G \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In other words, the controller attempts to avoid moving into a state where there is a disproportionate amount of positive or negative values in the system. In this work, we are dealing with a distribution target and are attempting to maintain both an average value and some level of spread across the states of the nodes within the system. This average and spread are determined by the mean and standard deviation of the specified target distribution.

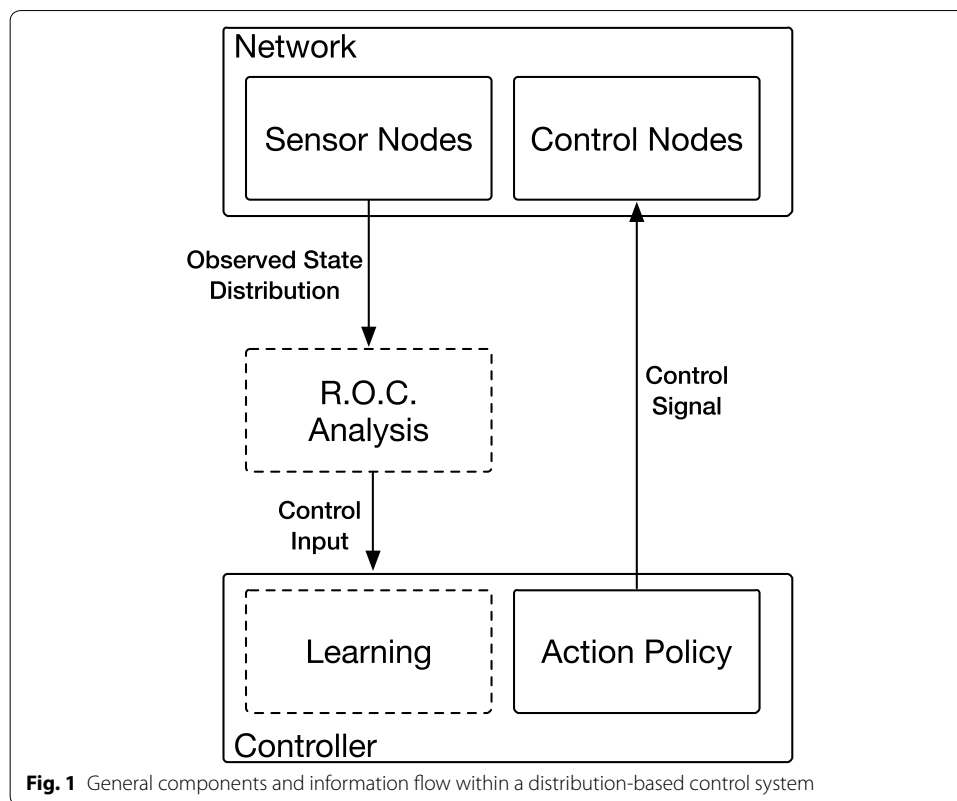
Distribution-based control

Within the work discussed in "[Related work](#)", the state of the network is generally represented by some vector capturing the state of each agent within the network. For example, within the work of [1] the goal of a control system would be to move the network state from one specific state vector to another (i.e. micro state control). In many scenarios, especially those involving flocking and crowding, we may be more interested in some overall property of the state of the system (i.e. macro state control). To address these scenarios, we propose the use of distribution-based control. The following subsection describes the components present in a general distribution-based control system. Following this, we formulate a general distribution control problem that is used in the experimental analysis presented later in this paper.

System components

There are several mandatory and optional components involved in a distribution-based control system. Figure 1 shows the basic components and information flow present in a basic distribution-based control system. A list of these components and a short description of each is included below:

- *Network* The network connecting entities within the system.
- *Sensor nodes* A set of nodes within the network which provides input regarding the network state to the controller.
- *Control nodes* A set of control nodes within the network, the state of which can be set at each time step. This set of nodes represents the interface which is used by the control system to affect the network state.
- *Target distribution* The defined ideal distribution of the system. In general, the control system attempts to keep the system's state distribution close to this target.
- *State distribution* A measure of the current distribution of the state, composed of the state value, $s(v, t)$, of each sensor node within the system. Both the state and target distributions can be represented by either a parameterized distribution (e.g. a normal distribution with specific mean and variance) or discretized to form a histogram.



- *Rate of change analysis (optional)* In the case of parameterized distributions, it is also possible to estimate the rate of change of the state distribution parameters over time (e.g. through the use of alpha–beta or Kalman filtering). This estimate can allow the ‘velocity’ of the system to be quantified, which could improve the performance of a control system by producing more accurate prediction of the future state of the system.
- *Controller* The controller is responsible for taking the state information as input and producing as output the control signals for each of the control nodes within the network. Within this work, reinforcement learning (see [10] for a thorough introduction) is used to generate a policy of signal selection based on the state distribution parameters.

Comparing distributions

As with control systems working with state vectors, to achieve distribution-based control, we require a method for comparing distributions. A method for distribution comparison is required for a number of reasons, such as determining if the controller has failed (i.e. in failure avoidance problems, see "[Failure avoidance control problem](#)"), determining a system’s velocity by comparing distributions at different time points, or determining how far away the system is from some other state. While there are a number of methods for comparing distributions, such as the Kullback–Leibler divergence and the total variation distance, we use the Hellinger distance here for the following reasons:

- It can be calculated easily on both continuous and discrete distribution types.
- It is bound between 0 and 1.
- It fulfils the properties of a metric.

In the continuous case, the Hellinger distance between two probability measures P and Q can be calculated as in Eq. 2, where f and g are the probability density functions of P and Q , respectively.

$$\begin{aligned} H^2(P, Q) &= \frac{1}{2} \int (\sqrt{f(x)} - \sqrt{g(x)})^2 dx \\ H^2(P, Q) &= 1 - \int \sqrt{f(x)g(x)} dx \\ H(P, Q) &= \sqrt{1 - \int \sqrt{f(x)g(x)} dx} \end{aligned} \quad (2)$$

For two discrete probability distributions P and Q defined over a common domain k , the Hellinger distance can be computed using Eq. 3.

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^k (\sqrt{p_i} - \sqrt{q_i})^2} \quad (3)$$

Failure avoidance control problem

Distribution-based control should be applicable to any utility-based network control problem. As the Hellinger distance, or any other distance measure used, allows the current state distribution to be quantitatively compared to some target, the utility of the system in relation to this distance can be measured at any time. Within this work, we focus solely on a failure avoidance type of problem, in which the control system attempts to keep the distance between the target state and measured state below some threshold value for as long as possible. A well known example of this problem from the domain of reinforcement learning is the pole-balancing problem, but more recently, the work of [9] and [3] has applied a failure avoidance approach to the problem of consensus avoidance in networks. In addition to consensus avoidance, there are a number of interesting failure avoidance problems that could be considered within social systems. For example, we may want to prevent the overall opinion or state in a system from changing too quickly¹, which may lead to panic (this is also applicable in economic systems). From an advertising perspective, we may want to avoid having the interest in a product or idea (as measured by mentions per unit time, for example) drop below some threshold. We may also want to prevent the disparity in some state value from growing too large between members of a system or group to minimize resentment, jealousy, spitefulness or general conflict.

To formulate a failure avoidance control problem using distributions, we must specify target values for distribution parameters and select a threshold value for the Hellinger distance between the measured state distribution and the target distribution. The control system must attempt to maintain the state distribution such that this Hellinger

¹ As we saw in 2008 during the financial crisis

threshold is not exceeded. If the Hellinger threshold is exceeded at any point, the controller is said to have failed in controlling the network. In this work, we define the target distribution to be $\mathcal{N}(0.0, 0.05)$. These values were selected because this type of distribution could be applied to various types of social problems where we wish for values to be centred around some state with a specific amount of variance. For example, this type of distribution could represent both problems of avoiding consensus and extremism within a system, as the state cannot converge to a single value, cannot bifurcate to extreme values, and cannot move significantly from the original mean.

Controllability analysis

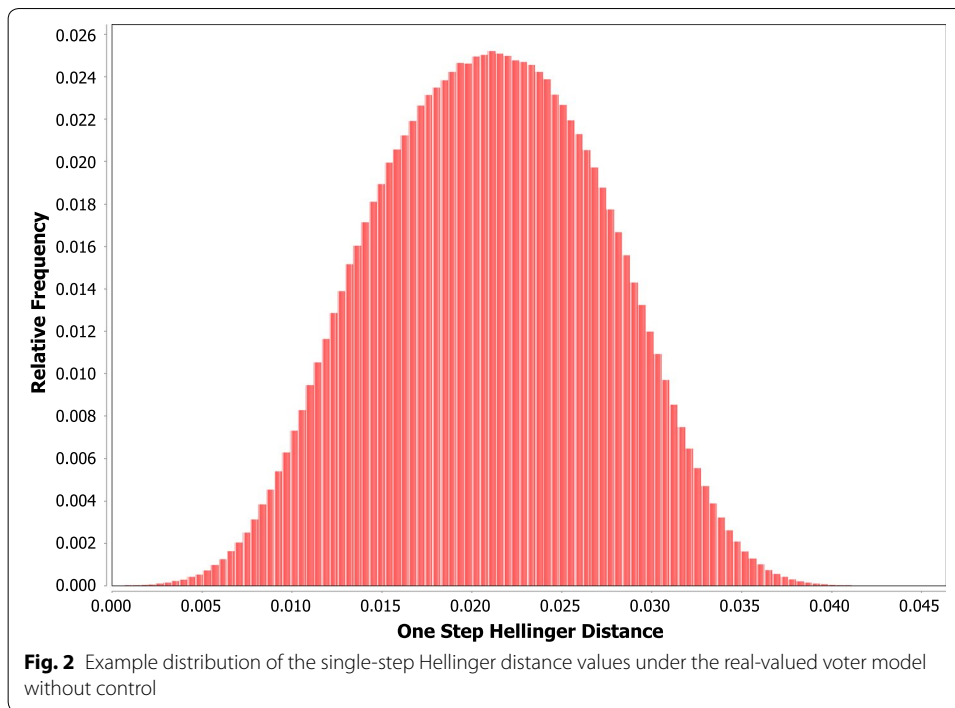
Within this work we are considering a failure avoidance problem inside a stochastic system. Assuming a model of this stochastic system is available, it may be possible to estimate the controllability of the system from a distribution-based control perspective. Ideally, a method similar to those developed within structural controllability research, which would be capable of estimating the overall controllability of a system when considering both the structural and behavioural components of a controller, is desired. By repeatedly simulating the model from a starting state (or many starting states) and measuring the Hellinger distance between subsequent states at different time steps, we can produce an estimate of the distribution of Hellinger distance values between the initial and resulting states for a time interval. This distribution can be treated as a measurement of the speed with which the system tends to change. As an example, consider Fig. 2, which shows the distribution of Hellinger distance values over a single time step within a simulation of the model used in this work when no control is applied. In this case, it would be unreasonable to expect to control the modelled system to a Hellinger threshold value of less than 0.03, as this threshold is exceeded in a single step in almost 10% of cases. Controlling the system to a threshold of 0.1 or 0.05, however, may be possible depending on how consistently the Hellinger distance moves over multiple steps and how significant of an effect the control system can exert on the system's distribution. Further development of this type of analysis could provide a tool for distribution-based control similar to those developed for full state controllability, allowing predictions to be made regarding the potential for effective control within a system.

Experimental model

To investigate the ability to control a system using the proposed distribution-based control approach, we formulate a problem using the real-valued voter model within the NCP framework described by [3]. The Network Control Problem definition requires the following components to be defined: a network, a diffusion model, a control system, and an objective function. The following subsections define each of the required components of the network control problem, including the objective function which uses a target distribution and the Hellinger distance measure to determine whether the system is still in an acceptable state.

Network

The network is represented by a graph $G = (V, E)$, where V is the set of nodes within the system and the set E represents the edges connecting nodes. The control problem



here is evaluated across three different theoretical network types. For each network type, 10 randomly generated networks of 100 agents each were considered. In all networks, each agent also included a link to itself. In addition to this, it is also ensured that each network consists of a single connected component. A description of each network type, as well as the parameters used in the generative models are described below. In the case of the random and small world networks, parameter values were selected to produce an average degree similar to those found in the scale free networks.

Random network

Each possible link between a pair of nodes, i and j , is included within the network with a probability $p = 0.031$ to produce an Erdős–Rényi random graph.

Scale free network

Links are formed between nodes based on the preferential attachment model described by [11].

Small world network

The small world networks were generated using the model of [12], with an average degree of 4 and a β value of 0.25.

Diffusion model

The specification of the network control problem from [3] defines the diffusion model with two parts: the sharing strategy and the learning strategy. In this work, we use a real-valued voter model to form the diffusion model. The voter model is commonly used to model the change of opinion within a group of networked individuals and has been

investigated in other network control research (i.e. [9]). Within this work, we consider a real-valued voter model in which each node's state is represented by a single value, bounded between -1.0 and 1.0 . Left uncontrolled, the voter model converges toward a single value over time. The equations representing the two strategies which define the real-valued voter model are included in Eqs. 4, 5 and 6. The sharing strategy ($Sh(v)$) of a node v within this model has each agent send its current state value ($s(v, t)$) to all of its neighbours, including itself, at each time step. Each shared piece of data a node v receives at a time step is stored in that node's information set I . The learning strategy ($L(v, I)$) for this model requires that, at each time step, every agent v move its state by an amount, *step* (0.01 is used as a constant here), toward one of its randomly selected neighbours' shared state values from the previous time step, as determined by Eq. 6.

$$Sh(v) = s(v, t) \quad (4)$$

$$L(v, I) = s(v, t) + (\text{step} \times \text{sign}) \quad (5)$$

$$\text{sign} = \begin{cases} -1 & \text{with probability } \frac{|\{s(u, t) < s(v, t) | s(u, t) \in I(v, t)\}|}{|I(v, t)|} \\ 1 & \text{with probability } \frac{|\{s(u, t) > s(v, t) | s(u, t) \in I(v, t)\}|}{|I(v, t)|} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Control system

The configuration of the control system specifies the set of nodes that the controller can set the state of to affect the overall network state. The results presented here consider many different possible configurations across the modelled networks. One of the main parameters of the configuration that is varied is the number of controllers, where we use either 3, 5 or 10 control nodes within the network. The set of controllers is determined using the FAR heuristic, as described by [9] and outlined in Algorithm 1. Starting from a seed node that is either included as input or selected randomly, this heuristic iteratively selects the next node such that it is the one with the largest shortest path to the current controller set. This has the effect of distributing the control nodes within the network in a way that maximizes the 'farness' between them.

Algorithm 1 Selection of control nodes using FAR heuristic.**Input:** D - matrix of network's all-pairs shortest paths**Input:** V_U - copy of set of network's nodes**Input:** N - size of control set**Input (optional):** V_I - a seed node from the network**Output:** V_C - set of control nodes, initially empty

```

function FARSELECT( $D, V_U, V_C, V_I$ )
  if  $V_I = null$  then
     $V_I \leftarrow$  randomly selected seed node from  $V_U$ 

   $V_C \leftarrow V_C \cup V_I$ 
   $V_U \leftarrow V_U \setminus V_C$ 
  while  $|V_C| < N$  do
     $v \leftarrow (v_j : \text{argmin}_i(D_{ij}) \geq \text{argmin}_i(D_{ik}), \forall k \in V_U, \forall i \in V_C)$ 
     $V_C \leftarrow V_C \cup v$ 
     $V_U \leftarrow V_U \setminus v$ 

```

The controller behaviour is learned using a reinforcement learning approach, as described in "[Learning a control policy](#)". As explained further in "[Learning a control policy](#)" to allow for more efficient execution of the learning and simulation process, a single signal (state value) is injected to all control nodes at each time step. By forcing the same signal to be used as input to each controller, the action space of the problem is made constant instead of growing exponentially relative to the number of controllers used. The value of the inserted signal is selected from a list consisting of values between -0.5 and 0.5 in 0.05 increments, allowing the controller to select from states within 10 standard deviations of the mean of the target distribution. This range was selected to ensure that the controller would be able to move the system in any direction that would be logically desirable.

Objective function

Within this work, we apply a failure avoidance approach within the distribution-based control problem. This requires both a target distribution and a Hellinger distance threshold to be specified. The target distribution we use here is a normal distribution with a mean of 0.0 and a standard deviation of 0.05 . As explained in "[Failure avoidance control problem](#)", this type of distribution could be applicable for a number of different types of social network control problems. With this target distribution and a specified Hellinger distance threshold, H_{\max} , the utility function of the overall network state can be defined using Eq. 7, where T and S represent the target and state distribution, respectively.

$$U(t) = \begin{cases} 1, & \text{if } H(T, S) < H_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The goal of the controller, then, is to maximize the utility over time. In other words, the controller must keep the distribution of the network's state within H_{\max} distance of

the specified target distribution. In the results presented here, the maximum length of a simulation is set at 50,000 steps, at which point it is said that the controller has successfully controlled the system.

Learning a control policy

To learn the control signal to insert into the network at any time step, we use reinforcement learning. More precisely, we use a gradient-descent SARSA [13] algorithm with a CMAC tiling [14] for function approximation of the real-valued distribution parameters. These are both commonly used solutions within the reinforcement domain. As was mentioned previously, the same signal is inserted into each controller to limit the size of the action space, which would otherwise grow exponentially with the number of controllers. As a comparison, using the single signal approach results in a constant sized action space of 21 actions, regardless of the number of control nodes, while the separate signal approach leads to an action space size of 9261 for three controllers and 4,084,101 for five controllers. The action set consisted of state values in the range of -0.5 to 0.5 in increments of 0.05 . The state space for the problem was represented by the difference between the state and target mean and standard deviation.

For each combination of network and controller set, up to 250 episodes were simulated for learning purposes, each starting from a randomly generated state within a Hellinger distance of 0.01 of the target distribution and ending if the distance between the state and target distribution exceeded the specified Hellinger threshold. Throughout training, the action policy was made progressively more greedy, which is necessary in many control applications due to the poor performance that can result from the selection of random actions. More specifically, a Boltzmann exploration policy was used with the temperature parameter of the Boltzmann distribution being halved after every 25 training episodes and reaching a final value of 0.0001 after 250 episodes. Based on preliminary experiments, low temperature values were necessary to ensure that the selection of random actions was not detrimental to the controller's performance. If the controller was capable of controlling the network for 50,000 steps in ten consecutive episodes, training was terminated early. Otherwise, all 250 episodes were used for learning the control policy.

After training was completed, the learned control policy was evaluated over a set of 250 episodes starting from pre-computed initial states, each of which was within a Hellinger distance of 0.01 of the target distribution. Each of these episodes is used to evaluate each network and control set combination to provide a consistent set of test scenarios. In all cases, the action selection policy used during this evaluation procedure was strictly greedy.

Results

To compare the performance of the learned control policy, we first require a baseline for comparison. To develop an estimate of the stability of the modelled system in the absence of intelligent control, we simulated the model within each network using three unintelligent control methods:

- *Null* No control signals are used.
- *Random* A single control signal randomly selected in the range of -0.5 to 0.5 is inserted to each control node at each time step.
- *Distribution* A single control signal is sampled from the target distribution and inserted to each control node at each time step.

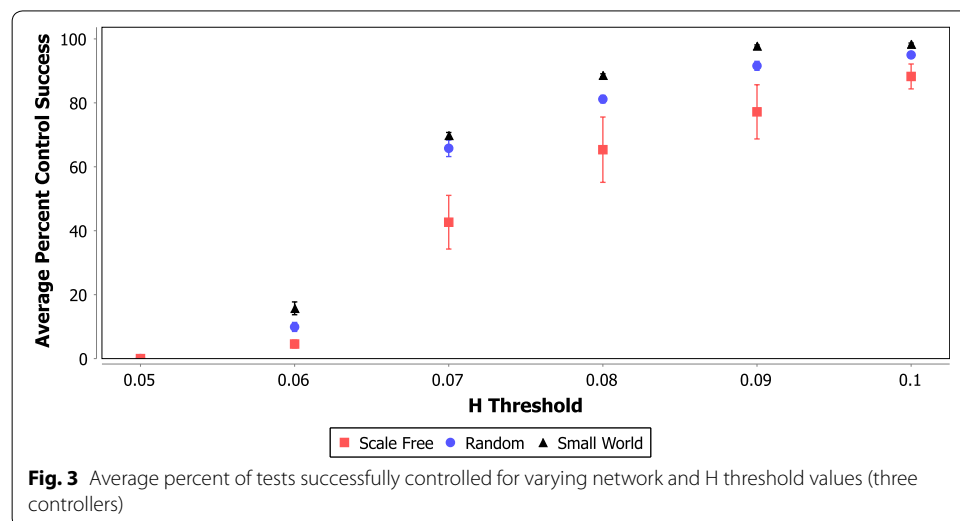
Table 1 shows the mean and standard deviation of the steps to failure across each of the three network types using these control methods. It should be noted that, for brevity, only the case of five control nodes and a Hellinger threshold of 0.1 is included here, as results were similar in other cases.

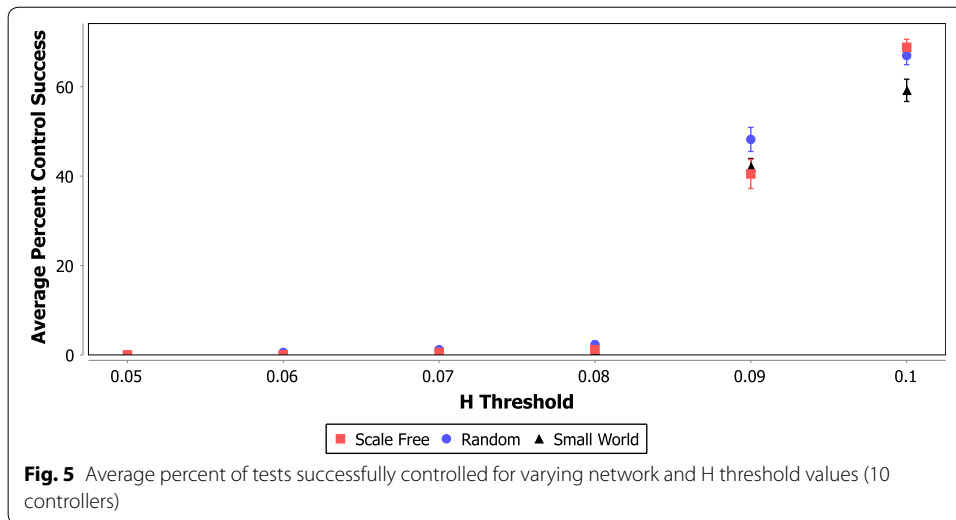
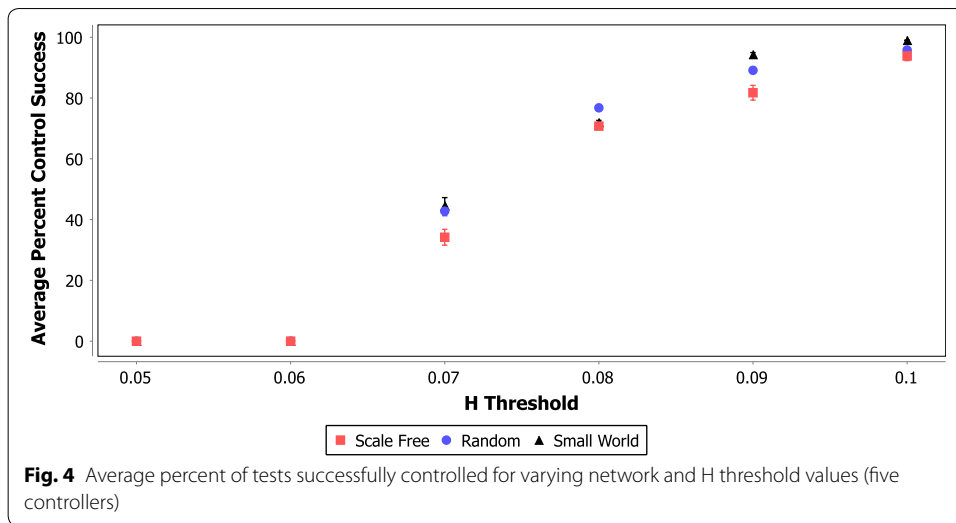
The data in this table demonstrate that in scenarios without control, or with only unintelligent control, the state distribution quickly moves away from the target distribution and the Hellinger distance threshold is exceeded. Due to the low mean steps to failure in Table 1, it should be no surprise that the percent of test cases that reached the 50,000 step success point was 0.0% for all three of these control strategies.

Figures 3, 4 and 5 show the average percent of all control tests that reached 50,000 rounds for each network type using a learned policy and 3, 5 and 10 controllers, respectively. These results include each possible set of controllers that can be selected from

Table 1 Average steps to failure over three network types using three unintelligent control strategies

Method	Network	Mean	SD
Null	Scale free	6.68	1.15
	Random	6.40	1.05
	Small world	5.50	0.85
Random	Scale free	6.36	1.15
	Random	6.05	1.04
	Small world	5.25	0.84
Distribution	Scale free	6.69	1.29
	Random	6.39	1.19
	Small world	5.60	1.01





each network instantiation using the FAR heuristic and their success in controlling each of the 250 test scenarios at each Hellinger threshold (resulting in a total of approximately 4.5 million data points). These figures demonstrate that the basic learning strategy described in "[Learning a control policy](#)" can produce an intelligent control policy that greatly increases the stability of the network system.

In scenarios with three or five controllers and a Hellinger threshold of 0.1, approximately 90% or more of the test scenarios are successfully controlled to the 50,000 round termination point. The case of ten controls has a lower probability of success, which is likely caused by the lack of precision in a single signal control approach. Essentially, with ten controllers the system may be moved too far in one direction, causing the system to fail. Another interesting result from these figures is that scale free networks appear to be more difficult to control than random or small world networks (this is more pronounced in the case of three controllers). Indeed, the summary presented for the three controller cases in Table 2 shows that the difference between the network classes was

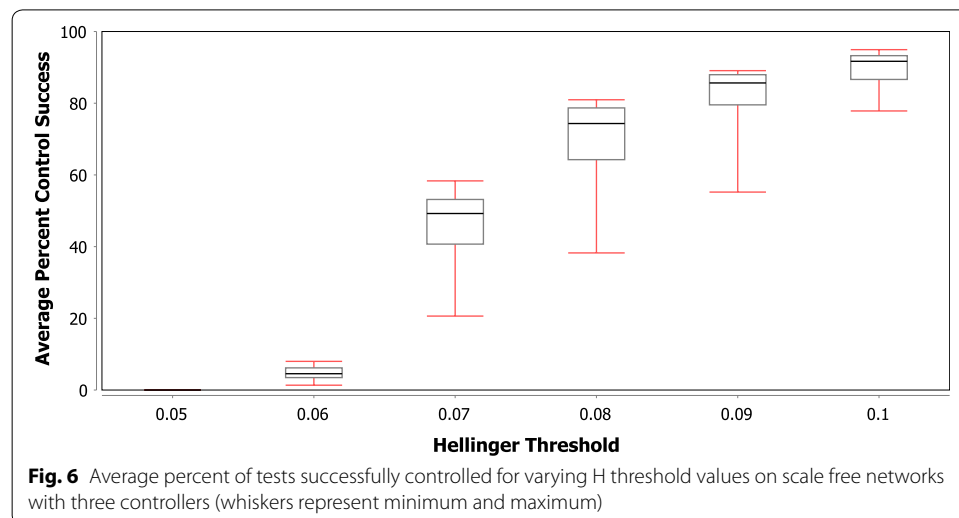
Table 2 Statistically significant difference in average number of successful tests for network class combinations (three controllers, two-tailed T test with $\alpha = 0.05$, X = Significant)

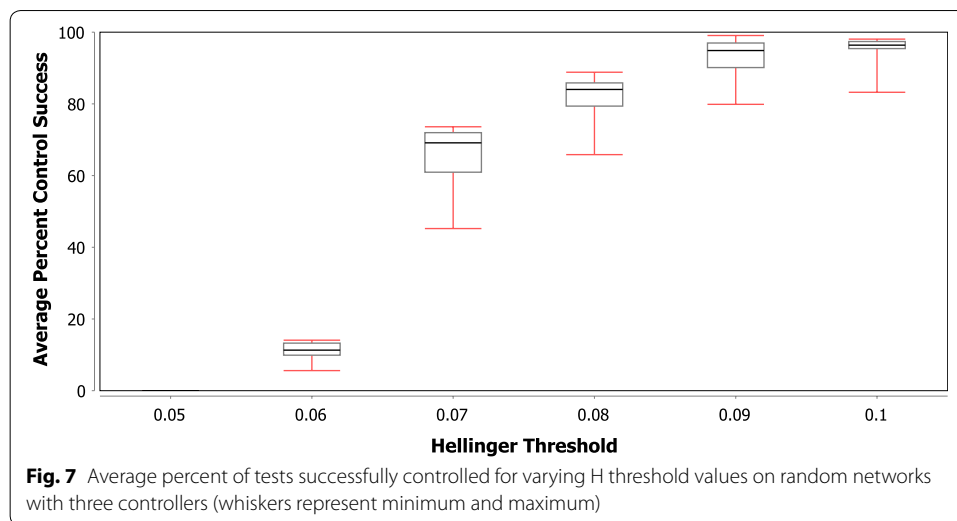
Networks		Hellinger threshold					
Class 1	Class 2	0.1	0.09	0.08	0.07	0.06	0.05
Scale free	Random	X	X	X	X	X	–
Scale free	Small world	X	X	X	X	X	–
Random	Small world	X	X	–	X	X	–

statistically significant under a two-tailed T test ($\alpha = 0.05$) under almost all Hellinger threshold and network combinations. While the data for the five and ten controller cases are not included here, the scale free networks generally showed a significant difference in all cases where some degree of control was possible.

Future work should attempt to determine whether this difference in control performance is due to scale free networks being inherently more difficult to control, or due to the fact that larger variation in node properties in scale free networks requires more specific control sets to be selected (again, these results are aggregated across each possible set of controllers).

While the previous discussion compared the control performance across three network types, the box-and-whisker plots in Figs. 6 and 7 show the variance in control performance across the different scale free and random network instantiations, respectively. A number of interesting points can be taken from this figure. First, while the boxes in the plots show that most of the networks are generally clustered in a fairly tight range, the minimum value in the scale free plot represents an extreme outlier, which indicates that one of these scale free network instantiations is significantly harder to control than the others. While the variance is not quite as high in the random network case, there is still a nearly 30% difference in control success between the best and worst random network under a Hellinger threshold of 0.07. Secondly, in both network types, there appears to be a phase shift between threshold levels of 0.07 and 0.06 where the overall controllability seems to move from ‘possible to control’ to ‘nearly impossible to control’. Finally,





the variance in controllability seems to be lowest in both the upper and lower threshold cases. The increase in variance among the middle threshold values further indicates the difference in controllability that may be observed across networks of the same class and could even represent a difference in the ‘phase shift’ threshold value for the different network instantiations.

As networks from each class are generated using the same production algorithm, they should have similar structural properties, and thus similar expected controllability. The differences in success rates in some cases, however, are shown to be more than 40%. When the single-step H distributions (see "[Controllability analysis](#)" for a brief explanation of what these distributions represent) for the best/worst scale free network were compared, the values were found using maximum likelihood estimation to be $\mathcal{N}(0.0179, 0.0061)$ and $\mathcal{N}(0.0174, 0.0061)$, respectively. The percent difference between the means of these distributions is only 2.8%, which would not be expected to cause an overall difference in control success probability of greater than 40%. When this analysis was extended to include 10 steps, the distribution parameter estimates changed to $\mathcal{N}(0.168, 0.025)$ and $\mathcal{N}(0.160, 0.025)$, which still only represents a 4.9% difference. In addition to the small difference in distribution parameters, of the two networks, the one with the largest ‘velocity’ of state change is the one that has higher control performance. The cause of the difference in control success, then, should not be exclusively that the model behaves differently without control within these networks, but must involve a difference in how control signals move throughout the networks. Again, future work should attempt to determine what is different between these networks that leads to such disparity in controller success. If certain network or control set properties can be identified as causing this disparity, then improved network stability could be achieved through either improved controller selection or modifications to the network structure. These network and control set properties could be determined by identifying correlations between control success and various network properties that differ between the networks and control sets (e.g. average path lengths, shortest paths, centrality).

Future work

There are a number of different areas in which this work will be expanded in the future. First, as mentioned in the previous section, the current results raise some interesting questions relating to network controllability. The results demonstrated that some networks, even those created using the same generative model, seem to be easier to control than others. Comparing different properties of these networks could help determine what type of properties result in networks that are more or less difficult to control. Algorithms from previous structural control analysis research could be applied to these same networks to determine if they predict the same increase in control difficulty. This comparison could either support or refute existing criticisms of the structural control analysis approach. Specifically, this could provide evidence to help determine whether ignoring the behavioural aspect of control leads to inaccurate conclusions regarding the practical controllability of networks.

In addition to comparing the overall controllability of different networks, the control sets that can be selected within a network could also be compared. Data produced through simulation of control systems using different control sets could help determine what properties are present/missing in successful/unsuccessful controller sets. Analysis of these data could lead to improved algorithms and heuristics for the selection of control nodes within a network control system.

Finally, the controllability analysis briefly discussed in "[Controllability analysis](#)" could be a useful tool in theoretically analysing networks and controllers. The current state of this analysis work only considers the expected distance the state distribution can move in some specified number of steps. Including a theoretical measure representing the ability of a control system to affect this distribution, however, could allow for a probabilistic analysis to determine the expectation of the system's controllability. This type of analysis could be used to compare possible controllers or possible network changes which could be implemented to form systems that are easier to control or less likely to fail.

Conclusion

This paper introduced the problem of distribution-based control as an alternative to existing approaches to complex network control which typically addresses the problem of full state controllability. When applying distribution-based control, we are no longer concerned with the exact state of the set of nodes within a system, but instead are attempting to maintain some distribution of state. This is important when considering many types of social network control problems, especially those involving crowding, opinion and influence. Within these types of problems, we are generally more concerned with the overall behaviour of the system or an aggregate measure (i.e. the distribution) of the state than an exact specification of the opinion or level of influence of each system participant. This paper has also continued the effort to investigate the behavioural component of network control, which has not previously been investigated in as much depth as the structural component.

To investigate the use of distribution-based control, a control system was implemented to prevent the distribution of state values in a real-valued voter model simulation from straying away from a specified mean and standard deviation. The experimental results demonstrated that it was possible to learn a control signal selection policy to successfully

maintain the desired network state distribution in a large percentage of cases, especially when compared to the 100% failure rate realized without intelligent control. These results also identified a number of important questions that should be addressed in future work, which were summarized in "[Future work](#)".

Author's contributions

Both authors contributed equally to the ideas and content within this paper. The original draft of this paper was prepared by DM, with edits and improvements suggested by TW. Both authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The data (networks, simulation output, etc.) are not currently stored on a publicly available repository, but the authors are willing to move the data to a publicly available location before publication.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

Not applicable.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 13 February 2017 Accepted: 13 February 2018

Published online: 01 March 2018

References

1. Liu Y-Y, Slotine J-J, Barabási A-L. Controllability of complex networks. *Nature*. 2011;473(7346):167–73.
2. Cowan NJ, Chastain EJ, Vilhena DA, Freudenberg JS, Bergstrom CT. Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PLoS ONE*. 2012;7(6):38398.
3. Runka A, White T. Towards intelligent control of influence diffusion in social networks. *Soc Netw Anal Min*. 2015;5(1):9.
4. Paraskevopoulos P. Modern control engineering. Boca Raton: CRC Press; 2001.
5. Lin CT. Structural controllability. *IEEE Trans Autom Control*. 1974;19(3):201–8.
6. Ruths J, Ruths D. Control profiles of complex networks. *Science*. 2014;343(6177):1373–6.
7. Yuan Z, Zhao C, Di Z, Wang W-X, Lai Y-C. Exact controllability of complex networks. *Nat Commun*. 2013;4:2447.
8. Zhao C, Wang W-X, Liu Y-Y, Slotine J-J. Intrinsic dynamics induce global symmetry in network controllability. *Sci Rep*. 2015;5:8422.
9. Runka A. On the control of opinion in social networks. Ph. D. Thesis, Carleton University, 2016.
10. Sutton R, Barto A. Reinforcement learning: an introduction. Cambridge: MIT Press; 1998.
11. Barabási A, Albert R. Emergence of scaling in random networks. *Science*. 1999;286(5439):509.
12. Watts D, Strogatz S. Collective dynamics of 'small-world' networks. *Nature*. 1998;393(6684):440–2.
13. Rummery GA, Niranjan M. On-line Q-learning using connectionist systems. Technical report, University of Cambridge, Department of Engineering, 1994.
14. Albus JS. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *J Dyn Syst Meas Control*. 1975;97(3):220–7.