

# Influencing Social Networks: An Optimal Control Study

**Daan Bloembergen**<sup>1</sup> and **Bijan Ranjbar-Sahraei**<sup>2</sup> and **Haitham Bou Ammar**<sup>3</sup> and  
**Karl Tuyls**<sup>4</sup> and **Gerhard Weiss**<sup>5</sup>

**Abstract.** We study the evolution of cooperation in social networks, aiming in particular at ways of influencing the behavior in such networks using methods and techniques from optimal control theory. This is of importance to many scenarios where politicians or policy makers strive to push consensus on some topic that may seem sub-optimal from individuals' perspectives. To this end, we employ the Continuous Action Iterated Prisoner's Dilemma (CAIPD) as model for the interactions in a social network. This model describes how neighboring nodes influence each other, and in effect determines how different strategies may spread through the network. We extend this model, incorporating a mechanism for external influence on the behavior of individual nodes. Next we prove reachability of an arbitrary network-wide agreement using the Lyapunov's Direct Method. Based on the theory of Linear-Quadratic Trackers we propose a step-wise iterative control algorithm, and show the effectiveness of the proposed controller in various Small World and Scale Free social networks.

## 1 INTRODUCTION

Modeling the evolution of cooperation in social networks has recently attracted much attention, aiming to understand how individuals work together and influence each other in such settings [7, 15]. In particular, researchers have focussed on how cooperation can be sustained in a population of agents despite its high cost. This is of interest to many real-world settings in which individual selfishness might hinder the acceptance of overall more efficient strategic choices. For example, policy makers may aim to convince companies to switch to a new green technology that yields better results in the long run, but requires an investment from each company initially, making them hesitant to be the first to switch over.

This strategic dilemma in choosing between the selfish rationality of defection and the social welfare of cooperation is aptly captured by the widely-adopted game of Prisoner's Dilemma [1]. In the Prisoners Dilemma model, defection is the best response against any opponent strategy. This makes mutual defection the single Nash equilibrium in the game. However, mutual cooperation would yield a higher reward to all players, and as such much research has been devoted to determining incentivising structures that promote cooperation. For example, cooperation can be promoted by punishing defectors as

<sup>1</sup> Dept. of Knowledge Engineering, Maastricht University, The Netherlands,  
email: daan.bloembergen@maastrichtuniversity.nl

<sup>2</sup> Dept. of Knowledge Engineering, Maastricht University, The Netherlands,  
email: b.ranjbarsahraei@maastrichtuniversity.nl

<sup>3</sup> Computer and Information Science Dept., Grasp Lab, University of Pennsylvania, United States, email: haithamb@seas.upenn.edu

<sup>4</sup> Dept. of Computer Science, University of Liverpool, United Kingdom,  
email: k.tuyls@liverpool.ac.uk

<sup>5</sup> Dept. of Knowledge Engineering, Maastricht University, The Netherlands,  
email: gerhard.weiss@maastrichtuniversity.nl

in [3, 16], or by pre-setting "cooperation committed" individuals for a given cost as in [6]. Both incentives increase the willingness to cooperate in scenarios where defection is individually the rational choice.

In parallel to this line of research, the control theory community has also developed strong approaches for the analysis of various types of multi-agent systems. For example, Liu et. al [10] study the controllability of social networks, by finding and controlling so-called "driving nodes", or Ren and Beard [14] study dynamical consensus seeking in multi-agent systems.

Very limited number of works exist which take advantage of optimal control theory for influencing the behaviors in social networks. In [5] optimal control theory is used to derive microscopic control laws using a macroscopic cost function. These microscopic control laws are then used by individuals to optimize their trajectories. In [17], a nonlinear dynamical model for time evolution of "friendliness levels" in the network is adopted as the main framework, and it is shown that any agent in the network is able to reach an arbitrary final state by perturbing its own neighborhood. However, both these works assume individuals that are able to compute the control signals and intentionally want to change their state in the network. In contrast to this, in this paper we deal with simple spontaneous individuals who follow the basic social interaction rules and are influenced by external signals. Moreover, to the best of our knowledge, this is the first work on influencing the evolution of cooperation using optimal control theory.

The Continuous Action Iterated Prisoner's Dilemma (CAIPD) model, proposed in [13], is adopted as the main framework for capturing the evolutionary and control behavior of social networks due to its generalization capabilities (see Section 2.1). This paper contributes by: (1) extending the CAIPD model to formally incorporate external influence, (2) proving reachability of arbitrary agreement under the introduced external controller, (3) developing an algorithmic technique capable of handling the time varying nature of the social network while balancing control effort and convergence speed, and (4) studying the performance of the proposed algorithm using empirical simulations that highlight the influence of both the network and control structure on the resulting system dynamics.

## 2 BACKGROUND

This section details CAIPD, adopted in this work, as well as selected topics from dynamical systems and control needed for the remainder of the paper.

### 2.1 The Model

In CAIPD [13],  $N$  individuals are positioned on  $N$  vertices  $v_i \in \mathcal{V}$  for  $i = \{1, \dots, N\}$  of a graph  $\mathbb{G} = (\mathcal{V}, \mathcal{W})$ . The symmetric  $N \times N$

adjacency matrix  $\mathcal{W} = [w_{ij}]$ , with  $w_{ij} \in \{0, 1\}$ , describes the  $i^{th}$  to  $j^{th}$  player connections with all  $w_{ii} = 0$ . In contrast to other models, CAIPD allows for a continuous degree of cooperation rather than a binary choice, captured by each player's state  $x_i$  ranging from  $x_i = 0$  for pure defection to  $x_i = 1$  for pure cooperation. A player pays a cost  $cx_i$  while the opponent receives a benefit  $bx_i$ , with  $b > c$ . This way a defector (i.e.,  $x_i = 0$ ) pays no cost and distributes no benefits. Accordingly, the fitness of player  $i$  can be calculated as  $f_i = -\deg[v_i]cx_i + b \sum_{j=1}^N w_{ij}x_j$ , where  $\deg[v_i]$  denotes the number of neighbors of  $v_i$ . CAIPD assumes rational players that adopt imitation dynamics, and copy their neighbors strategy proportional to fitness. Player  $i$  adopts its  $j^{th}$  neighbor's strategy with strength of  $p_{ij} = w_{ij} \cdot \text{sigmoid}(\beta(f_j - f_i))$ , where  $\beta$  determines how selective adoption is towards fitter strategies.

A network with a state  $\mathbf{x}$  and topology  $\mathbb{G}$  is defined as  $\mathbb{G}_{\mathbf{x}} = (\mathbb{G}, \mathbf{x})$  with  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ . The network  $\mathbb{G}_{\mathbf{x}}$  can then be regarded as a dynamical system, where  $\mathbf{x}$  evolves according to the nonlinear mapping  $\dot{\mathbf{x}} = [h_1(\mathbf{x}), \dots, h_N(\mathbf{x})]^T$ , with  $h_i(\mathbf{x})$  denoting the dynamics of the  $i^{th}$  individual in  $\mathbb{G}_{\mathbf{x}}$ . Precisely,  $h_i(\mathbf{x}) = \frac{1}{\deg[v_i]} \left[ \sum_{j=1}^N p_{ij} (x_j(t) - x_i(t)) \right]$ . This dynamical system can be re-written in a standard form by introducing the Laplacian of  $\mathbb{G}$ ,  $\mathcal{L}(\cdot)$  as

$$\dot{\mathbf{x}}(t) = -\mathcal{L}[\mathbf{x}(t)] \mathbf{x}(t), \quad (1)$$

where

$$\mathcal{L}_{ij} = \begin{cases} -p_{ij}/\deg[v_i] & \text{if } i \neq j \\ \sum_{j=1}^N p_{ij}/\deg[v_i] & \text{if } i = j \end{cases} \quad (2)$$

In CAIPD, fitness updates are performed at each time step  $t$ . To induce more realistic behaviors, as well as allowing for optimal control incorporation, we introduce the concept of dwell time  $\tau$  as proposed in [8], and integrated this into CAIPD. This allows to rewrite the model of [13] as piece-wise time invariant:

$$\dot{\mathbf{x}}(t) = -\mathcal{L}_k \mathbf{x}(t), \quad (3)$$

where  $\mathcal{L}_k = \mathcal{L}(\mathbf{x}(k\tau))$  for  $k\tau < t < (k+1)\tau$  and  $k = 1, 2, \dots$ . Clearly, as  $\tau \rightarrow 0$ , the system in Equation 3 collapses to that introduced in [13], while for  $\tau \rightarrow \infty$  a static consensus model, as proposed in [4], is attained.

## 2.2 Lyapunov's Direct Method

Generally, a nonlinear dynamical system can be represented by  $\dot{\mathbf{x}} = \mathbf{f}(t; \mathbf{x}, \mathbf{u})$ , where state variables  $\mathbf{x}$  change according to nonlinear mappings  $f_i$  for  $i \in \{1, \dots, N\}$ . In the special case of linear and time invariant (LTI) systems, the system collapses to

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (4)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$  and  $\mathbf{u} = [u_1, u_2, \dots, u_q]^T$  represent state and input variables, respectively, while  $\mathbf{A}$  and  $\mathbf{B}$  denote the transition and control matrices. In control design, the aim is to determine a feed-back controller capable of driving the system to a reference state. Detailed discussion of such techniques are beyond the scope of this paper<sup>6</sup>. Here, the interest is in stability and convergence analysis of dynamical systems. Stability can be studied in the vicinity of equilibria (i.e., points where  $\dot{\mathbf{x}} = 0$ ). To quantify such neighborhoods, an open ball,  $\mathcal{B}(\bar{\mathbf{x}}, \epsilon)$ , centred at  $\bar{\mathbf{x}}$  with a radius  $\epsilon$ , (i.e., the set  $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \bar{\mathbf{x}}\| < \epsilon\}$ , where  $\|\cdot\|$  represents the  $L_2$ -norm) is defined. Lyapunov stability can then be stated as:

<sup>6</sup> Interested readers are referred to [9, 11] for a thorough study of control theory.

**Definition 1 (Lyapunov Stability)** An equilibrium point  $\mathbf{x}_e$  of a nonlinear system is said to be Lyapunov stable, if for all  $\epsilon > 0$  there exists a  $\delta > 0$  such that:  $\bar{\mathbf{x}} \in \mathcal{B}(\mathbf{x}_e, \delta) \implies \mathbf{f}(t; \bar{\mathbf{x}}, \mathbf{0}) \in \mathcal{B}(\mathbf{x}_e, \epsilon)$  for all  $t \geq 0$ .

In **Lyapunov's direct method**, the rate of change in a potential function of the system can be used to verify the Lyapunov stability. Namely, a Lyapunov function is defined as:  $V(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}$  such that  $V(\mathbf{x}) \geq 0$  with equality if and only if  $\mathbf{x} = 0$ . The system is asymptotically stable in the sense of Lyapunov when  $\frac{d}{dt} V(\mathbf{x}) \leq 0$  with equality if and only if  $\mathbf{x} = 0$ .

## 2.3 Optimal Control Design

In Linear Quadratic Tracking (LQT), the goal is to control a dynamical system in the form of (4) to follow a reference signal  $\bar{\mathbf{y}}$  defined by:

$$\dot{\mathbf{z}} = \mathbf{F}\mathbf{z}, \quad \bar{\mathbf{y}} = \mathbf{H}\mathbf{z}$$

where  $\mathbf{z}$  is the internal state vector,  $\bar{\mathbf{y}}$  is the output of the system, and  $\mathbf{z}(t_0) = \mathbf{z}_0$  is the initial state. To solve such a problem, the following augmented system is defined so as to capture the dynamical behavior of both the original and the tracker dynamics:

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\mathbf{u}$$

with

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \quad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

and  $\tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}]^T$ . To capture the incurred tracking error, the following cost function is defined:

$$\mathcal{J} = \int_{t_0}^T \left[ \tilde{\mathbf{x}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right] dt \quad (5)$$

with

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & -\mathbf{QH} \\ -\mathbf{H}^T \mathbf{Q} & \mathbf{H}^T \mathbf{QH} \end{bmatrix}$$

being the augmented state cost matrix, and  $\mathbf{Q}$  and  $\mathbf{R}$  are being the state and input cost matrices, respectively. The goal is then to determine  $\mathbf{u} = \mathbf{u}^*$  such that the cost function of Equation 5 is minimized. The optimal control law can be attained using  $\mathbf{u}^*(t) = -\mathbf{R}^{-1} \tilde{\mathbf{B}}^T \tilde{\mathbf{P}}(t) \tilde{\mathbf{x}}(t)$ , where  $\tilde{\mathbf{P}}(t)$  is the solution to the Riccati differential equation:

$$\begin{aligned} \dot{\tilde{\mathbf{P}}}(t) &= \tilde{\mathbf{A}}^T \tilde{\mathbf{P}}(t) + \tilde{\mathbf{P}}(t) \tilde{\mathbf{A}} - \tilde{\mathbf{P}}(t) \tilde{\mathbf{B}}^T \mathbf{R}^{-1} \mathbf{B} \tilde{\mathbf{P}}(t) + \tilde{\mathbf{Q}} \\ \tilde{\mathbf{P}}(T) &= \mathbf{0} \end{aligned}$$

The controller is typically simplified by partitioning  $\tilde{\mathbf{P}}(t)$  in terms of the original systems:

$$\tilde{\mathbf{P}}(t) = \begin{bmatrix} \mathbf{P}(t) & \mathbf{P}_{12}(t) \\ \mathbf{P}_{12}^T & \mathbf{P}_{22}(t) \end{bmatrix}$$

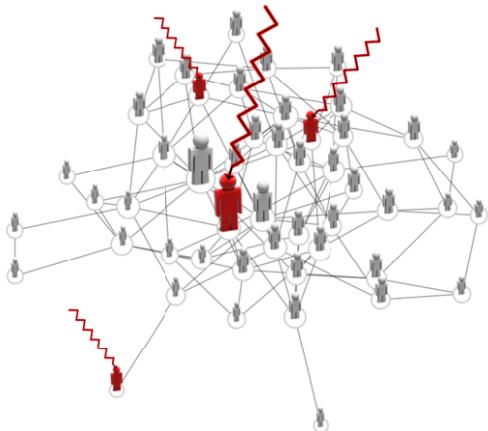
leading to:

$$\mathbf{u}^*(t) = \mathbf{K}_1(t)\mathbf{x}(t) + \mathbf{K}_2(t)\mathbf{z}(t), \quad (6)$$

$$\begin{cases} \mathbf{K}_1(t) = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}(t) \\ \mathbf{K}_2(t) = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}_{12}(t) \end{cases}$$

with the following partitioned Riccati equations:

$$\begin{aligned} \dot{\mathbf{P}}(t) &= -\mathbf{P}(t)\mathbf{A} - \mathbf{A}^T \mathbf{P}(t) + \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}(t) - \mathbf{Q} \\ \dot{\mathbf{P}}_{12}(t) &= \mathbf{P}_{12}(t)\mathbf{F} - \mathbf{A}^T \mathbf{P}_{12}(t) + \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}_{12}(t) + \mathbf{QH} \\ \mathbf{P}(T) &= \mathbf{P}_{12}(T) = \mathbf{0} \end{aligned} \quad (7)$$



**Figure 1:** Network under external influence; the controlled individuals are shown in red and the zigzag arrows denote the control signals.

### 3 REACHABILITY ANALYSIS

To influence the evolution in a network following the CAIPD model,  $L \leq N$  control signals,  $u_1, u_2, \dots, u_L$ , affecting the behavior of  $L$  controlled individuals,  $x_1, x_2, \dots, x_L$ , are introduced. These signals can be generated from any external source such as news outlets, government regulations, or even distributed leaders outside the network. Formally, considering the CAIPD model  $\dot{\mathbf{x}} = -\mathcal{L}_k \mathbf{x}$  as in (3) the external influence is incorporated as

$$\dot{\mathbf{x}} = -\mathcal{L}_k \mathbf{x} + \mathbf{B} \mathbf{u} \quad (8)$$

where  $\mathbf{u} = [u_1, u_2, \dots, u_L]$  is a vector of control signals, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_{L \times L} \\ \mathbf{0}_{(N-L) \times L} \end{bmatrix}$$

is the input matrix. A schematic of a network under external influence is shown in Figure 1, where the control signals are signified using red zigzag arrows.

In this work, the main aim is to reach network-wide agreement of the form  $\mathbf{x}_f = x^* \mathbf{1}$ , with  $x^*$  representing the cooperation level. Reachability of  $\mathbf{x}_f$  at time  $t_0$  is defined as:

**Definition 2 (Reachability)** A state  $\mathbf{x}_f$  is reachable at time  $t_0$ , if there exists a control input  $\mathbf{u}_r(t)$  such that  $\mathbf{x}_f = \lim_{t \rightarrow \infty} \mathbf{x}(t; t_0, \mathbf{x}_0, \mathbf{u}_r(t))$ , where  $\mathbf{x}_0 = \mathbf{x}(t_0)$ .

Based on the above, the following theorem showing the reachability of any feasible agreement (i.e.  $0 \leq x^* \leq 1$ ) assuming a single controlled individual is presented.

**Theorem 1 (Reachability of Agreements)** For the CAIPD model with external influence in form of (8), any agreement  $0 \leq x^* \leq 1$  is reachable at  $t_0$  for arbitrary  $\mathbf{x}_0$  by influencing a single controlled individual  $x_c$  using the control input

$$u_r = \begin{cases} -\epsilon \cdot \text{sgn}(e) + \mathbf{B}^\top \mathcal{L}_k \mathbf{x} & \text{IF } e \neq 0 \\ \mathbf{B}^\top \mathcal{L}_k \mathbf{x} & \text{IF } e = 0 \end{cases} \quad (9)$$

with  $\epsilon > 0$  and error defined as  $e = x_c - x^*$ . Then

$$\lim_{t \rightarrow \infty} \mathbf{x}(t; t_0, \mathbf{x}_0, u_r(t)) = x^* \mathbf{1}.$$

**Proof:** We split the control process into two phases. The first is driving the network toward the manifold  $e = 0$  such that the controlled node reaches the agreement value (i.e.,  $x_c \rightarrow x^*$ ). In the

second, the goal is keeping the system on that manifold (i.e.,  $e = 0$ ), by ensuring that  $\dot{e} = 0$ . Consider the Lyapunov function candidate  $V = 0.5e^2$ . It can be easily verified that  $V \geq 0$ , except for the case of having the controlled individual reaching an agreement (i.e.,  $e = 0$ ), where  $V = 0$ . The derivative of the candidate function is:

$$\dot{V} = e \dot{e} = e \dot{x}_c = e (-\mathbf{B}^\top \mathcal{L}_k \mathbf{x} + u)$$

Replacing the control input  $u_r$  of (9) for  $e \neq 0$  in the above leads to:

$$\dot{V} = e (-\epsilon \cdot \text{sgn}(e)) = -\epsilon |e| \quad (10)$$

where  $|\cdot|$  denotes the absolute value of a scalar. Therefore, according to the Lyapunov's direct method, if  $\epsilon > 0$  then in finite time  $V$  will attain a value of zero, and thus  $e = 0$ . This concludes the first phase of the control process.

In the second phase, the network should be ensured to stay on the manifold  $e = 0$ . The derivative of the error signal is computed as:

$$\dot{e} = \dot{x}_c = -\mathbf{B}^\top \mathcal{L}_k \mathbf{x} + u \quad (11)$$

By replacing the control input  $u_r$  of (9) for  $e = 0$  in (11),  $\dot{e} = \dot{x}_c = 0$ , and thus guaranteeing that the system stays on the manifold  $e = 0$ . Without loss of generality, assume that the controlled individual is the first individual in CAIPD model. Then, the network can be represented in following form:

$$\frac{d}{dt} \begin{bmatrix} x_c \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \mathcal{L}_{k_{21}} & \mathcal{L}_{k_{22}} & \dots & \mathcal{L}_{k_{2N}} \\ \mathcal{L}_{k_{31}} & \mathcal{L}_{k_{32}} & \dots & \mathcal{L}_{k_{3N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{k_{N1}} & \mathcal{L}_{k_{N2}} & \dots & \mathcal{L}_{k_{NN}} \end{bmatrix} \begin{bmatrix} x^* \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \quad (12)$$

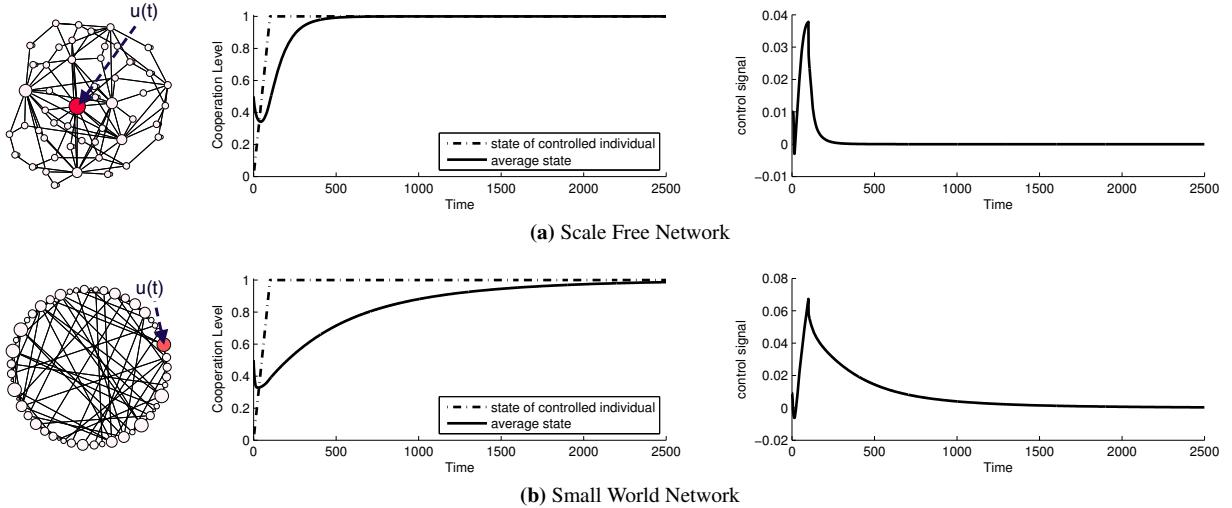
Using the CAIPD's model properties, the new state space matrix is diagonally dominant, with non-positive diagonal elements. Furthermore, in the graph associated with this network, the only node without incoming links is that associated with the controlled individual. Therefore, through the second phase of control, there exists a spanning tree in the network where the controlled individual is its root. Based on [12], the network eventually achieves an agreement. This implies that  $\lim_{t \rightarrow \infty} x_i(t) = \lim_{t \rightarrow \infty} x_c(t) = x^*$ , thus concluding the proof.  $\square$

The results of applying the proposed controller in (9) and choosing  $x^* = 1$  on two sample networks are shown in Figure 2. In both networks, the Lyapunov functions converge to zero around Time = 150, after which the system evolves on the  $e = 0$  manifold until all individuals reach pure cooperation (i.e. Time = 600 and Time = 2500 for Scale Free and Small World networks, respectively.).

Clearly, from Theorem 1, any arbitrary agreement can be reached by using the control signal of Equation (9). Although successful, two problems arise. Firstly, the aforementioned method makes use of only one controlled node and ignores the multidimensional control scenario. Secondly, no systematic procedure for choosing the proper value of  $\epsilon$ , affecting both control effort and speed of convergence, is derivable. Next, a solution to the above problems, based on optimal control theory, is presented.

### 4 ITERATIVE CONTROL ALGORITHM

In this section, the previous technique is extended to the case where several nodes of the social network are externally influenced. In particular, the general scenario in which neither direct online measurements of the state values nor of the real system's dwell time  $\tau$  are



**Figure 2:** Reachability of pure cooperation using a single controlled individual. Both networks contain 50 nodes, with an average node degree of 4; the small world network is generated using a 0.5 rewiring probability.

available, is considered. Therefore, an optimal control policy  $\mathbf{u}^*$  is designed based on the initial configuration of the system and the *evaluation* dwell time  $\tau_{\text{eval}}$ , which can be thought of as the controller's estimation of the real dwell time  $\tau_{\text{real}}$ . This control policy aims at driving the system towards an arbitrary agreement  $x^*$  in finite time  $T$ . The dynamics of the estimated state vector  $\hat{\mathbf{x}}$  can be written as:

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = -\tilde{\mathcal{L}}_j \hat{\mathbf{x}}(t) + \mathbf{B} \mathbf{u}_j(t) \\ \hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0), \end{cases} \quad (13)$$

for  $t_{j-1} < t \leq t_j$ , where  $j = 1, 2, \dots, \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$  and  $t_j = j \cdot \tau_{\text{eval}}$ .

$\tilde{\mathcal{L}}_j$  is a fixed Laplacian matrix computed according to  $\hat{\mathbf{x}}(t_{j-1})$  and  $\mathbf{B}$  is the input matrix as defined in (3). The cost function used to compute the optimal control policy in the  $j^{\text{th}}$  time period is defined as:

$$\mathcal{J}_j = \int_{t_j}^T [\tilde{\mathbf{x}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{u}_j^\top \mathbf{R} \mathbf{u}_j] dt \quad (14)$$

where  $\tilde{\mathbf{Q}}$  and  $\mathbf{R}$  are the same as in Equation (5) with  $\mathbf{H} = \mathbf{I}_{N \times N}$ , and  $\tilde{\mathbf{x}} = [\hat{\mathbf{x}}, x^* \mathbf{1}]^\top$  is the augmented state vector. The goal is then to determine  $\mathbf{u}_j^*(t)$  such that the cost function of (14) is minimized for each  $j$ . Following (14), the optimal control law is:

$$\mathbf{u}_j^*(t) = \mathbf{K}_{j_1}(t) \hat{\mathbf{x}}(t) + \mathbf{K}_{j_2}(t) x^* \mathbf{1}, \quad (15)$$

$$\begin{cases} \mathbf{K}_{j_1}(t) = -\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_j(t) \\ \mathbf{K}_{j_2}(t) = -\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_{j_{12}}(t) \end{cases} \quad (16)$$

$\mathbf{K}_{j_1}$  and  $\mathbf{K}_{j_2}$  are computed by backward integrating the following Riccati equations:

$$\begin{aligned} \dot{\mathbf{P}}_j(t) &= \mathbf{P}_j(t) \mathcal{L}_j + \mathcal{L}_j^\top \mathbf{P}_j(t) + \mathbf{P}_j(t) \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_j(t) - \mathbf{Q} \\ \dot{\mathbf{P}}_{j_{12}}(t) &= \mathcal{L}_j^\top \mathbf{P}_{j_{12}}(t) + \mathbf{P}_j(t) \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_{j_{12}}(t) + \mathbf{Q} \mathbf{H} \\ \mathbf{P}_j(T) &= \mathbf{P}_{j_{12}}(T) = 0 \end{aligned} \quad (17)$$

The details of the proposed controller are given in Algorithm 1. Given the initial states, network topology and fixed parameters, the controller provides control dynamics  $\mathbf{K}_{j_1}(t)$  and  $\mathbf{K}_{j_2}(t)$  for  $j = 1, 2, \dots, \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$ , which can be used to generate the control input as

$$\mathbf{u}^*(t) = \mathbf{K}_{j_1}(t) \hat{\mathbf{x}}(t) + \mathbf{K}_{j_2}(t) x^* \mathbf{1}, \quad (18)$$

for controlling the real network (8) for  $t_{j-1} < t < t_j$  and every  $j$ .

---

#### Algorithm 1 Step-wise control

---

**Input:**  $\mathbb{G}, \mathbf{x}(t_0), \mathbf{B}, \mathbf{Q}, \mathbf{R}, \tau_{\text{eval}}, T, \delta$

- 1: Initialize  $\mathbf{K}_1, \mathbf{K}_2$
- 2:  $\hat{\mathbf{x}}(t_0) \leftarrow \mathbf{x}(t_0), j \leftarrow 1$
- 3: **while**  $j \leq \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$  **do**
- 4:   Compute  $\mathcal{L}_j$  using  $\hat{\mathbf{x}}(t_{j-1})$  and  $\mathbb{G}$  according to (2)
- 5:    $\mathbf{P}_j, \mathbf{P}_{j_{12}} \leftarrow \mathbf{0}$
- 6:    $t' \leftarrow T$
- 7:   **while**  $t' > (j-1) \cdot \tau_{\text{eval}}$  **do** {backwards integration}
- 8:     Update  $\mathbf{P}_j(t'), \mathbf{P}_{j_{12}}(t')$  using (17)
- 9:     Calculate  $\mathbf{K}_{j_1}(t'), \mathbf{K}_{j_2}(t')$  using (16)
- 10:     $t' \leftarrow t' - \delta t$
- 11: **end while**
- 12:   Calculate  $\mathbf{u}_j^*(t)$  using (18)
- 13:   Simulate the CAIPD model with  $\mathcal{L}_j$  and  $\mathbf{u}_j^*$  for  $t_{j-1} < t \leq t_j$  using (13) and store  $\hat{\mathbf{x}}(t_j)$
- 14:    $j \leftarrow j + 1$
- 15: **end while**

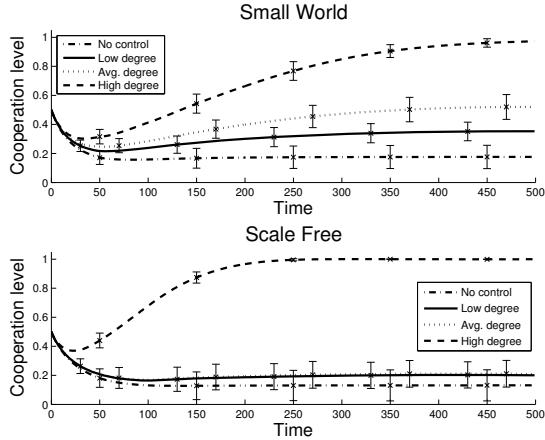
**Output:**  $\mathbf{K}_1, \mathbf{K}_2$

---

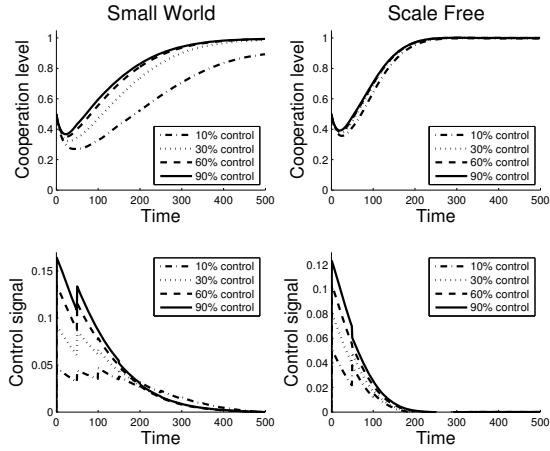
## 5 NUMERICAL VERIFICATION

The proposed step-wise controller is numerically evaluated on a number of networks with varying properties. In particular, the Barabási-Albert *scale-free* [2] and Watts-Strogatz *small world* networks [18], both exhibiting many properties of real-world systems, are adopted. For all experiments reported in this section, the controller is evaluated on 100 randomly generated networks in order to ensure statistical significance. In all experiments, networks with an average node degree of 4 are used; the small world networks are generated using a 0.5 rewiring probability. For the optimal controller,  $\mathbf{Q} = \mathbf{I} \times 0.001$  and  $\mathbf{R} = \mathbf{I} \times 25$  were used in the cost function of Equation (5) and (14) to ensure smooth control signals<sup>7</sup>. Unless stated otherwise, the network is assumed to exhibit piece-wise linear dynamics of  $\tau_{\text{real}} = 50$  (i.e. the real dwell time of the system). The controller step size was set to  $\tau_{\text{eval}} = 50$ . The final agreement goal was set to pure cooperation  $x^* = 1$  for all individuals.

<sup>7</sup> Please note that we acquired similar results for various  $\mathbf{Q}$  and  $\mathbf{R}$  settings



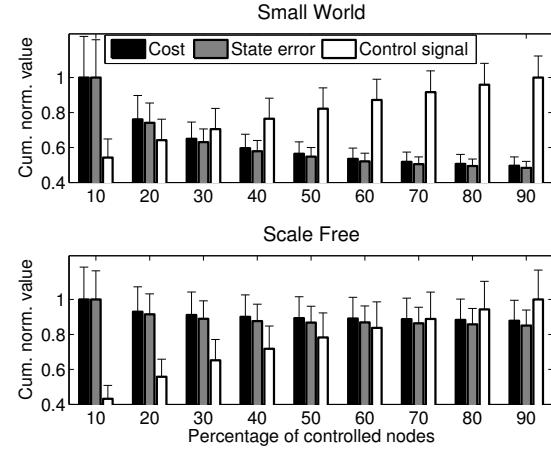
**Figure 3:** Comparing heuristic controllers with different sets of controlled nodes. In each case, 20% of the nodes are controlled.



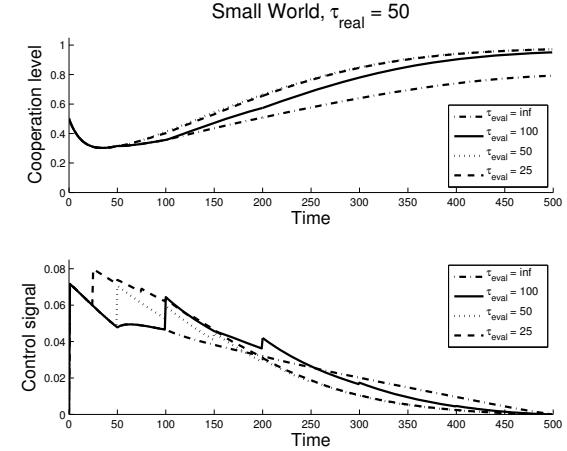
**Figure 4:** Comparing the influence of the percentage of controlled nodes on performance and control input.

In the first set of experiments the investigation considered which nodes need to be controlled. Firstly, the performance of the proposed control algorithm when influencing 20% of the nodes with either the lowest, average, or highest degrees was studied. Figure 3 shows the results for both Small World and Scale Free networks, leading to the following two conclusions: (1) controlling high degree nodes improve convergence to the cooperative state, and (2) this effect is strongest for Scale Free networks. Intuitively, this can be explained by the fact that high degree nodes allow the control input to spread quickly over the network. Moreover, in Scale Free graphs few high degree nodes are involved in the majority of all connections (so-called *hubs*), which explains why these are of key importance in such networks.

In the second set of experiments, the goal was to study the effect of the number of controlled nodes, while keeping their type fixed (i.e., highest degree), on the overall performance. Figure 4 shows both the average network state over time and the corresponding total control input for different percentages of controlled nodes. It is clear that increasing the number of controlled nodes improves convergence, in particular for Small World networks. For Scale Free graphs this effect is almost negligible; again this can be intuitively explained by the scale free degree distribution exhibited in such networks. Moreover, it can be observed that the total control input increases, although not proportionally: controlling more nodes means that individually they need less input.



**Figure 5:** Comparing the influence of the percentage of controlled nodes on cost, state error and control input.

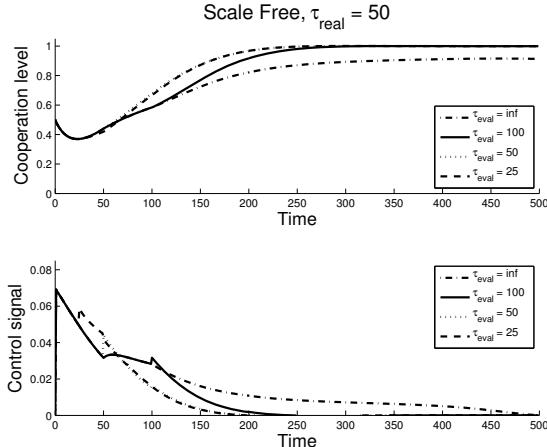


**Figure 6:** Influence of controller step size  $\tau_{\text{eval}}$  on performance and control input.

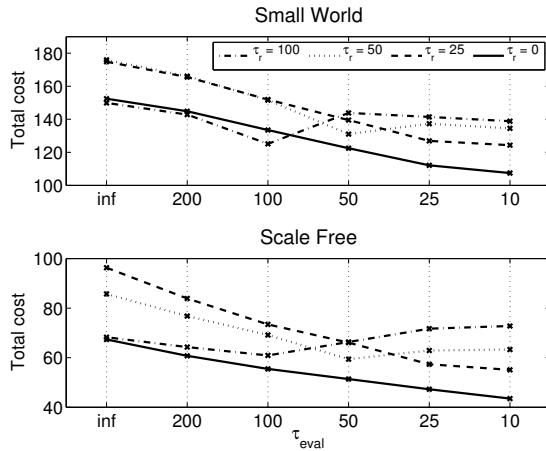
Figure 5 summarises more extensive experiments for a range of percentages of controlled nodes, showing the effect on the cost function, state error, and control input. All measures are normalised for presentation purposes. These results again show the relative insensitivity of Scale Free graphs to the number of controlled nodes. Moreover, it is clear that for Small World networks the benefit of more controlled nodes diminishes as their percentage grows. Depending on the cost function parameters  $Q$  and  $R$  this gives rise to a trade-off between decreasing state error on the one hand and increasing control input on the other.

In this set of experiments, the number and type of controlled nodes are kept fixed (the 20% highest degree nodes), while the influence of modifying the controller's steps size  $\tau_{\text{eval}}$  is considered. Figures 6 and 7 show the average network state and control input over time for different values of  $\tau_{\text{eval}}$  when  $\tau_{\text{real}} = 50$ , both for Small World and Scale Free networks, respectively. Here,  $\tau_{\text{eval}} = \inf$  means the controller assumes a fixed linear system, e.g. it never updates its estimate of the Laplacian matrix. Clearly, decreasing  $\tau_{\text{eval}}$  improves convergence, while also resulting in a smoother control signal. When  $\tau_{\text{eval}} < \tau_{\text{real}}$  the convergence does not change anymore, however the total control effort might increase as the controller overestimates the dynamics of the real system.

Finally, Figure 8 summarises a more extensive range of experiments with varying  $\tau_{\text{real}}$  and  $\tau_{\text{eval}}$ . Several observations can be made from these results. Firstly, the curve showing  $\tau_{\text{real}} = 0$ , meaning that



**Figure 7:** Influence of controller step size  $\tau_{\text{eval}}$  on performance and control input.



**Figure 8:** Influence of controller step size  $\tau_{\text{eval}}$  on the total cost, for systems with different real step sizes, indicated with  $\tau_r$  in this figure.

the network is continuously changing, shows that a smaller step size for the controller indeed leads to lower total cost. A similar conclusion can be drawn from the case of  $\tau_{\text{real}} = 25$ . In contrast, for larger  $\tau_{\text{real}}$  a (local) minimum can be observed when the controller step size  $\tau_{\text{eval}}$  exactly matches  $\tau_{\text{real}}$ , after which the total cost rises again. This is due to the overestimation of the system dynamics leading to higher initial control effort than actually required, as also noted before in the discussion of Figures 6 and 7. Finally, it is interesting to observe that faster changing networks (i.e. decreasing  $\tau_{\text{real}}$ ) tend to yield lower total cost, in particular when the controller step size is reasonably small as well. In such cases, the inherent dynamics of the network help the evolution of cooperation, although some initial external control is still required for convergence to the full cooperative state, as seen before in Figure 3.

## 6 CONCLUSION

In this paper we have studied the evolution of cooperation in social networks, focusing on means of controlling this evolution to achieve network-wide cooperation in the continuous-action iterated prisoner's dilemma (CAIPD) model. This model, introduced in previous work, has already been shown to provide insights into the sustainability of cooperation in complex networks. However, convergence to pure cooperation is not guaranteed and depends highly on the network structure. Building on this work, the main contributions

of this paper are threefold. Firstly, the CAIPD model has been extended to allow for external influence on arbitrary nodes. Secondly, reachability of network-wide agreement on an arbitrary cooperation level has been proven. Thirdly, a step-wise iterative control algorithm aiming at minimizing the control effort and state error over time has been proposed. Finally, the performance of this algorithm has been empirically evaluated on various Small World and Scale Free social networks. Studying the (optimal) control of social networks is relevant for many real-world settings. For example, politicians may try to convince particular well-connected individuals of their ideas, hoping those individuals will then spread their ideas through their network. Similarly, the government might provide tax deductions to companies that switch to sustainable production, hoping that their competitors follow automatically due to market dynamics. As such, studying the control of social networks has broad applicability, and many directions for future work can be taken. Of particular interest would be to automatically identify the key nodes that should be controlled to minimize cost or convergence time.

## REFERENCES

- [1] Robert Axelrod and William D. Hamilton, ‘The evolution of cooperation’, *Science*, **211**, 1390–6, (1981).
- [2] Albert-László Barabási and Réka Albert, ‘Emergence of scaling in random networks’, *Science*, **286**(5439), 509–12, (October 1999).
- [3] Robert Boyd, Herbert Gintis, and Samuel Bowles, ‘Coordinated punishment of defectors sustains cooperation and can proliferate when rare’, *Science*, **328**(5978), 617–620, (2010).
- [4] Morris H. DeGroot, ‘Reaching a consensus’, *Journal of the American Statistical Association*, **69**(345), 118–121, (1974).
- [5] Greg Foderaro, Silvia Ferrari, and Thomas A Wettergren, ‘Distributed optimal control for multi-agent trajectory optimization’, *Automatica*, (2013).
- [6] The Anh Han, Luís Moniz Pereira, Francisco C. Santos, and Tom Lenaerts, ‘Good agreements make good friends’, *Scientific reports*, **3**, (2013).
- [7] Lisa-Maria Hofmann, Nilanjan Chakraborty, and Katia Sycara, ‘The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study’, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, 685–692, (2011).
- [8] Ali Jadbabaie, Jie Lin, and A. Stephen Morse, ‘Coordination of groups of mobile autonomous agents using nearest neighbor rules’, *Automatic Control, IEEE Transactions on*, **48**(6), 988–1001, (2003).
- [9] *The Control Handbook*, ed., W. Levine, CRC Press, 1996.
- [10] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási, ‘Controllability of complex networks’, *Nature*, **473**(7346), 167–173, (2011).
- [11] Richard M. Murray, Zexiang Li, and S. Shankar Sastry, *A mathematical introduction to robotic manipulation*, CRC press, 1994.
- [12] Bijan Ranjbar-Sahraei, Haitham Bou Ammar, Daan Bloembergen, Karl Tuyls, and Gerhard Weiss, ‘Theory of cooperation in complex social networks’, in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-14)*, (2014).
- [13] Bijan Ranjbar-Sahraei, Haitham Bou-Ammar, Daan Bloembergen, Karl Tuyls, and Gerhard Weiss, ‘Evolution of Cooperation in Arbitrary Complex Networks’, in *13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, (2014).
- [14] Wei Ren and Randal W. Beard, ‘Consensus seeking in multiagent systems under dynamically changing interaction topologies’, *Automatic Control, IEEE Transactions on*, **50**(5), 655–661, (2005).
- [15] Francisco C. Santos and Jorge M. Pacheco, ‘Scale-Free Networks Provide a Unifying Framework for the Emergence of Cooperation’, *Physical Review Letters*, **95**(9), 1–4, (August 2005).
- [16] Karl Sigmund, Christoph Hauert, and Martin A. Nowak, ‘Reward and punishment’, *Proceedings of the National Academy of Sciences*, **98**(19), 10757–10762, (2001).
- [17] Tyler H Summers and Iman Shames, ‘Active influence in dynamical models of structural balance in social networks’, *EPL (Europhysics Letters)*, **103**(1), 18001, (2013).
- [18] Duncan J. Watts and Steven H. Strogatz, ‘Collective dynamics of ‘small-world’ networks’, *nature*, **393**(6684), 440–442, (1998).