2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Control of Decisions in Stochastic Multi-Agent Systems

Elisa Gaetan[1,3], Laura Giarré[1], Simona Sacone[2], Paolo Falcone[1]

Abstract— **In this paper, we consider a multi-agent system, where part of the agents interact with a system that can be externally controlled. Such a system can be thought as a super-agent or the environment the multi-agents operate in. We model the agents' decision process as a Markov Chain and the externally controlled system as a linear dynamical system. We formulate the problem of controlling the probability that a set of agents make a specific decision as a Model Predictive Control problem. Such a control problem formulation is completed by a controllability analysis of the system. Simulation results from a small-scale example reveal the potential of the considered modeling and control framework for applications where the decisions of a set of agents are to be influenced to achieve a desired objective.**

I. Introduction

Exploiting multi-agent models, preferably stochastic, to gain a deeper understanding of the communications and interconnections between agents within a network has become increasingly significant. This topic is of main interest to companies, that seek strategies to influence their customers [1], and to political factions, which aim to shape the preferences of voters [2]. Beyond these examples, a more profound comprehension of the interactions between agents could lead to improvements in other fields, such as transportation [3] and smart systems [4]. In a network of agents, the opinion can be influenced by external factors or by the action of one or multiple agents (malicious or benevolent). The problem of modeling such interactions or studying how to force network agents to change their decisions has been addressed in the past.

Within the literature, we can broadly categorize such research into two distinct approaches. On one side, there are papers that primarily strive to describe and understand the interactions between agents. These works predominantly concentrate on studying and learning the complexity of these interactions, aiming to gain insights into the underlying dynamics and decision-making processes. On the other side, we find works that, while recognizing and acknowledging the existence of these interactions, aim at controlling the network, towards specific objectives or desired states. These two approaches, though interconnected, represent distinct research directions within the field of agent-based systems and social network dynamics.

Belonging to the first approach, in [5] the authors acknowledge the presence of direct influences within social networks and aim to estimate these influences, from partial observations. In [6], the authors concentrate on addressing the issue of modeling interactions within a social network by utilizing Markovian models. Specifically, they assume that sharing opinions might induce a transformation in the agents' decision-making processes. Moving to [7], the authors extend upon the insights drawn from the aforementioned work. In their approach, they combine the concepts of attraction and repulsion between agents, thus accounting for interactions that alter the conventional transition rates between states.

Conversely, some works have concentrated on the identification of one or more agents (or nodes) to control and guide the entire network toward a desired state. In [8], the authors develop an algorithm for the selection of control nodes and the design of control inputs to steer a network towards a specific target state. Another approach can be found in [9], where the authors propose a dynamical model that is under the influence of a leader agent. They assume that the leader has the capacity to influence the other surrounding agents in the network.

In the present work, our approach aligns with the second category of papers, utilizing some insights from the first category. Thus, we propose a framework that leans more towards the *control* of opinions, rather than the modeling of interactions. Our framework involves the integration of an opinion dynamics model, specifically a Discrete Time Markov Chain, with a general dynamical model in which the control input is applied. The first model describes the evolution of the state of one or more *target agents* within a network. The other model characterizes the evolution of a specific agent, referred to as *controller agent*, which belongs to the network but we assume it to be controllable and immune to the influences of the other agents, similar to the approach outlined in [10]. However, in this configuration, the controller agent has the capacity to influence the states of the target agents through interactions.

Our approach can be compared to a more conventional framework related to Markov Decision Processes (MDP) [11], which is employed to study the decision-making process of a stochastic agent. In this framework, the optimal policy, which represents the sequence of actions to undertake, is determined by maximizing the cumulative reward. Within the context of MDP, the agent has the capacity to make various choices to accomplish a specific goal. This can be

*This work is partially supported by the EU Next-Generation (PNRR) within the Italian National Ph.D. Program in Autonomous Systems (DAuSy) and by the Mobility National Center, SPOKE 6.

[1]E. Gaetan, L. Giarré and P. Falcone are with the Engineering Department 'Enzo Ferrari', University of Modena and Reggio Emilia, Modena, Italy. {elisa.gaetan, laura.giarre, paolo.falcone}@unimore.it

[2]S. Sacone is with the Department of Informatics, Bioengineering, Robotics and Systems Engineering 'DIBRIS', University of Genova, Italy. simona.sacone@unige.it

[3]E. Gaetan is also with the Italian National PH.D. DAUSY, Politecnico di Bari, Bari, Italy. e.gaetan@phd.poliba.it

viewed as a form of control action executed by the agent on its own decision-making process. Instead, our framework is based on the assumption that there exists an external input capable of manipulating the states of the agents, which is an outer contribution that can affect the system.

This paper is organized as follows. Section II presents a motivating example for our framework, while Section III establishes the mathematical basis of our model. Section IV states how the controller is derived, and Section V presents the simulation results. Finally, conclusions are drawn in Section VI.

II. Illustrative Example

The next example illustrates our objectives and intentions, along with a possible application setup.

Example 1: A population of consumers is partitioned in groups G_1 and G_2, based on, e.g., common attitudes or habits, and there is no interaction between them. The individuals of G_1 and G_2 are asked to make a decision and purchase a new t-shirt to replace their own. A correlation between the colors of the old and new t-shirts is found to hold with a known probability. Furthermore, the probability of individuals in each group owning t-shirts of different colors, such as red, yellow, or blue is known.

A t-shirt firm wants to improve the sales of blue t-shirts and look for sales, sponsorship, and advertisement policies to convince consumers in G_1 and G_2. However, G_1 and G_2 are affected by the price differently and the individuals of each group are influenced by the decisions made by the individuals of the other group.

Our (the firm's) objective is to decide the price of the t-shirts to control the probability that members of G_1 and G_2 purchase a new blue t-shirt.

In the rest of the paper, we build a modeling and control framework that can be deployed in applications where, similarly to Example 1, the behavior (alternatively, decisions or opinions) of a set of stochastic and interacting agents is affected, to some extent, by the evolution of a process that can be externally controlled, e.g., the environment the agents operate in.

III. Mathematical Model

Similarly to [6], we employ a discrete-time version of a Markov Chain (DTMC) to describe the decision-making process of a stochastic agent r. The DTMC is described by the equation

$$\Pi^r(k+1) = (Q^{(r)})^T \Pi^r(k) \tag{1}$$

with initial condition $\Pi^r(0) = \Pi^r(0)$. $\Pi^r(k) \in \mathbb{R}^{m\times 1}$ is the state probability vector at time instant k such that $\Pi^r(k)^T\mathbf{1} = 1$ (where $\mathbf{1}$ is a column vector with all the entries equal to 1) and m corresponds to the number of discrete states the agent r can evolve to. $Q^{(r)}$ is the transition probability matrix and must be row-stochastic ($Q^{(r)}\mathbf{1} = \mathbf{1}$), with all the entries in $[0,1]$.

A set of $n \geq 1$ Markovian agents evolving accordingly to (1) can be considered as a Markovian network [6]. The evolution of the network is given by

$$\Pi(k+1) = Q^T \Pi(k), \tag{2}$$

where $\Pi(k)$ is the probability vector of the network. To distinguish between the single case, we call the states of the network *configurations*. Specifically, a single network configuration is given by the combination of the states of all the agents. Assuming that each agent can achieve m states and denoting with $\mathcal{S}^r$ the set of states of the agent r, the set of configurations $\mathcal{S}$ of the network has dimension $p = m^n$ and it is generated as

$$\mathcal{S} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^r \times \cdots \times \mathcal{S}^n, \tag{3}$$

with $\times$ the Cartesian product.

If the agents are independent, then the matrix $Q \in \mathrm{R}^{p\times p}$ is simply given by

$$Q = Q^{(1)} \otimes \cdots \otimes Q^{(r)} \otimes \cdots \otimes Q^{(n)}, \tag{4}$$

with $\otimes$ the Kronecker product. Conversely, if they are not independent, their interactions can influence one another, and this influence is captured by the structure of the matrix Q. In the literature (see [6], [7]), the interactions among agents are modeled through a modification of the transition matrix, leading to the equation

$$\Pi(k+1) = \tilde{Q}^T \Pi(k) = (Q + \Omega(\bar{\boldsymbol{\alpha}}))^T \Pi(k). \tag{5}$$

In (5), $\Omega(\bar{\boldsymbol{\alpha}})$ models the interactions described by the parameter vector $\bar{\boldsymbol{\alpha}}$. As mentioned in Section I, while the focus of [6] is primarily adjusting the values of $\bar{\boldsymbol{\alpha}}$ based on the network structure, our objective is to control $\Pi(k)$, given a desired probability vector Π^{des} to achieve.

Our aim is to integrate the Markov model of the evolution of one (1) or more (2) agents (target agents), with a model that describes the evolution of the agent that we can control (controller agent), assuming it to be independent with respect to the target agents, as in [10]. In this work, we assume that the controller agent is the *environment* the target agents are placed in. The model $\mathcal{W}$ describing the evolution of the target agents' opinions, together with the dynamics of the environment, is

$$\Pi(k+1) = Q^T \Pi(k) + \mu(k, x(k)), \tag{6a}$$

$$x(k+1) = f(x(k), u(k)). \tag{6b}$$

In $\mathcal{W}$, (6a) represents the evolution of the target agents, and (6b) the evolution of the environment. In particular, $\Pi(k)$ describes the evolution of the state probabilities of the target agents, based on their intrinsic behavior, given by Q. The influence from the controller agent, under the assumption that it is independent of the target agents, is given by the bias vector $\mu(k, x(k))$. Lastly, the controller agent state dynamics is given by the function $f(x(k), u(k))$. This function could be either linear or nonlinear (based on the scenario), and it is controlled by the control input $u(k)$. Moreover, in (6a), $\mu(k, x(k))$ can be seen as the control input directly applied on the DTMC.

In this first step, we want to keep the model $\mathcal{W}$ linear. Therefore, we make two assumptions:

1) the influence $\mu(k, x(k))$ that the environment exerts on the target agents is linear w.r.t. $x(k)$, so that $\mu(k, x(k)) = Hx(k)$, and
2) the evolution of the environment is ruled by a linear discrete system $x(k+1) = Ax(k) + Bu(k)$.

Consequently, the system $\mathcal{W}$ becomes

$$\underbrace{\begin{bmatrix} \Pi(k+1) \\ x(k+1) \end{bmatrix}}_{x_a(k+1)} = \underbrace{\begin{bmatrix} Q^T & H \\ 0 & A \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \Pi(k) \\ x(k) \end{bmatrix}}_{x_a(k)} + \underbrace{\begin{bmatrix} 0 \\ B \end{bmatrix}}_{\tilde{B}} u(k), \quad (7)$$

where $x_a(k)$ is the augmented state, and $\tilde{A}$ and $\tilde{B}$ are the augmented matrices of the system $\mathcal{W}$, under the aforementioned assumptions.

Before advancing with the design of the controller, we undertake a preliminary study on the reachability of the system.

A. Reachability of DTMC

In control theory [12], a discrete linear system is said to be *reachable* if the rank of the reachability matrix is full.

In (7), the dynamics of DTMC are under the influence of the state of the environment denoted as $x(k)$, which can be seen as the control input of the DTMC itself. Let us assume that the evolution of the environment's state is fully reachable. Therefore, we have

$$rank(\mathcal{R}_E) = rank([B, AB, \ldots, A^{m-1}B]) = m. \quad (8)$$

Notably, by construction, the environment has the same order as the DTMC. This implies that, regardless of the initial state $x(0) = x_0$, it is feasible to attain any desired state for the environment. Consequently, there are no restrictions on the control input applied to the DTMC, and our primary concern is to demonstrate the reachability of the DTMC portion of the model $\mathcal{W}$.

Consider the reachability matrix of the DTMC[1] in (7), where $\Pi(k), x(k) \in \mathbb{R}^{m \times 1}$ and $H \in \mathbb{R}^{m \times m}$ is non-singular,

$$\mathcal{R}_{MC} = [H,\ Q^T H,\ (Q^T)^2 H,\ \ldots,\ (Q^T)^{m-1} H]. \quad (9)$$

If $rank(\mathcal{R}_{MC}) = m$, then the system is reachable. By analyzing (9), it can be observed that the system is reachable (since H is non-singular), and the image of $\mathcal{R}_{MC}$ is

$$\mathcal{I}m(\mathcal{R}_{MC}) = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m\}, \quad (10)$$

where $\boldsymbol{e}_i$ is the i^{th} vector of the orthonormal basis.

Given a final state Π_k and initial conditions Π_0, we want to investigate if it is possible to achieve Π_k through a feasible input sequence.

Remark 1: Henceforth, we will abbreviate the forced evolution component of the DTMC dynamics $Hx(k)$ as $\mu(k)$, as we assumed that the environment is fully reachable and H is non-singular.

[1]The subsystem given by $\Pi(k+1) = Q^T \Pi(k) + Hx(k)$.

This problem is solvable *iff*

$$\Pi_k - (Q^T)^k \Pi_0 = \mathcal{R}_{MC}^k\, \mathcal{U}_k, \quad (11)$$

where $\mathcal{U}_k = [\mu(k-1), \ldots, \mu(0)]^T$ and $\mathcal{R}_{MC}^k$ is the reachability matrix at step k. Eq. (11) has a solution, namely (at least) one input sequence, *iff*

$$\Pi_k - (Q^T)^k \Pi_0 \in \mathcal{I}m(\mathcal{R}_{MC}^k). \quad (12)$$

The vector $\Pi_k - (Q^T)^k \Pi_0$ can be rewritten as a linear combination of the orthonormal basis vectors. Thus, considering (10), the condition (12) is satisfied.

Nonetheless, it is important to note that, in our scenario, the mere knowledge of system controllability is insufficient. Indeed, the control input $\mu(k)$ applied in (6a) should not alter the structural properties of a Markov Chain. For this reason, the constraint $\mu(k)^T \mathbf{1} = 0\ \forall k$ must be enforced for the input sequence (for the derivation and proof of this constraint, please refer to Section IV-A). However, it is worth noting that we can exploit the fact that the target state to be reached must adhere to the specific constraint $\Pi_k^T \mathbf{1} = 1$, since Π_k is a probability vector. In the following, we prove that it is possible to find a feasible input sequence $\mathcal{U}_k$, given the desired probability Π_k, the initial condition Π_0 and the row-stochastic transition matrix Q.

Theorem 1: Consider the linear system given by

$$\Pi_{k+1} = Q^T \Pi_k + \mu_k, \quad (13)$$

where Π_k is a vector of probabilities and Q is a row-stochastic matrix. If this system is reachable for a given initial condition Π_0 and a final state vector Π_k, then the input sequence $\mathcal{U}_k = [\mu(k-1), \ldots, \mu(0)]^T$ satisfies the constraint:

$$\mathcal{U}_k^T \mathbf{1} = \mathbf{0}. \quad (14)$$

Proof: Consider the probability evolution (11). By definition, Π_k and Π_0 are probability vectors and therefore

$$\Pi_k^T \mathbf{1} = 1\ ,\ \Pi_0^T \mathbf{1} = 1. \quad (15)$$

Thus

$$\begin{aligned} (\Pi_k)^T \mathbf{1} &= ((Q^T)^k \Pi_0 + \mathcal{R}_k\, \mathcal{U}_k)^T \mathbf{1} \\ 1 &= \underbrace{((Q^T)^k \Pi_0)^T \mathbf{1}}_{a} + \underbrace{(\mathcal{R}_k\, \mathcal{U}_k)^T \mathbf{1}}_{b}, \end{aligned} \quad (16)$$

where $\mathcal{R}_k$ is the reachability matrix at step k. For sake of simplicity, let us analyze the terms *a* and *b* of (16) separately. For the term a, the following holds

$$\begin{aligned} &\Pi_0^T ((Q^T)^k)^T \mathbf{1} = \Pi_0^T (Q^T (Q^T)^{k-1})^T \mathbf{1} = \\ &\Pi_0^T ((Q^T)^{k-1})^T \underbrace{Q\mathbf{1}}_{=\mathbf{1}} = \cdots = \Pi_0^T \mathbf{1} = 1. \end{aligned} \quad (17)$$

In the term *b*, $(\mathcal{R}_k\, \mathcal{U}_k)^T \mathbf{1} = \mathcal{U}_k^T \mathcal{R}_k^T \mathbf{1}$. $\mathcal{R}_k^T \mathbf{1}$ can be written as

$$\mathcal{R}_k^T \mathbf{1} = \begin{bmatrix} I \\ Q \\ Q^2 \\ \ldots \\ Q^{k-1} \end{bmatrix} \mathbf{1} = \begin{bmatrix} I\mathbf{1} \\ Q\mathbf{1} \\ Q(Q\mathbf{1}) \\ \ldots \\ Q^{k-2}(Q\mathbf{1}) \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \ldots \\ \mathbf{1} \end{bmatrix}. \quad (18)$$

Thus, by combining (17) and (18), the following holds

$$1 = \underbrace{1}_{a} + \underbrace{\mathcal{U}_k^T \mathbf{1}}_{b} \tag{19}$$

and finally $\mathcal{U}_k^T \mathbf{1} = 0$. ■

Hence, the DTMC is reachable with a feasible input sequence, under the assumption of full reachability of the environment. However, it is important to note that this assumption may not be feasible in all scenarios, and this could impact the overall reachability of the system.

In the next section, we will show how to derive the control policy u for (7), with the specific goal of controlling the evolution of the DTMC component.

IV. Model Predictive Controller

In this section, we formulate a MPC problem of reference tracking for model (7), assuming the state is fully observable. As in standard MPC control theory [13], we consider a quadratic cost to be minimized

$$J = x_a(N)^T S x_a(N) + \sum_{i=0}^{N-1} \Big(\widetilde{x}_a(k+i)^T Q_{err} \widetilde{x}_a(k+i) + u(k+i)^T R u(k+i) \Big) \tag{20}$$

where

- $\widetilde{x}_a(k+i) = x_a(k+i) - x_a^{ref}(k+i)$ is the tracking error and Q_{err} is the corresponding weighting matrix;
- $u(k+i)$ is the control input and R is the corresponding weighting matrix;
- $x_a(N)$ is the last state and S is the corresponding weighting matrix;
- N corresponds to the value of the finite control horizon.

It is worth noting that within the state vector, denoted as $x_a = [\Pi, x]^T$, Π represents a probability vector, and as such, it must preserve a well-defined structure. Consequently, when formulating the minimization problem for the MPC, it is essential to integrate constraints on the environment's state

$$F_{eq} x(k) = f_{eq}, \tag{21a}$$

$$F_{ineq} x(k) \leq f_{ineq}. \tag{21b}$$

In the following subsections, we first derive the constraints matrices $F_{eq}, f_{eq}, F_{ineq}, f_{ineq}$, and we specify the matrices for the cost function (20). Then, we make the structure of the condensed matrices for the MPC explicit.

A. Equality constraint

First, we address the derivation of the equality constraint matrices according to (21a). As already anticipated, given that the first submodel is a DTMC, we need to account for the following properties:

1) $\sum_i \pi_i(k) = 1 \ \forall k$ (in matrix form $\Pi(k)^T \mathbf{1} = 1$). This ensures that the vector of probabilities remains normalized, and
2) $Q\mathbf{1} = \mathbf{1}$ because of the definition of a row-stochastic matrix.

By exploiting these two conditions together and applying them on the top submodel of (7), we obtain

$$\begin{aligned}
(\Pi(k+1))^T &= (Q^T \Pi(k))^T + (Hx(k))^T \\
(\Pi(k+1))^T \mathbf{1} &= (Q^T \Pi(k))^T \mathbf{1} + (Hx(k))^T \mathbf{1} \\
1 &= \Pi(k)^T Q \mathbf{1} + (Hx(k))^T \mathbf{1} \\
1 &= \Pi(k)^T \mathbf{1} + (Hx(k))^T \mathbf{1} \\
1 &= 1 + (Hx(k))^T \mathbf{1},
\end{aligned} \tag{22}$$

and thus we derive the following equality constraint for the environment's state

$$(Hx(k))^T \mathbf{1} = 0 \quad \forall k, \tag{23}$$

which can be rewritten as $\mathbf{1}^T H x(k) = 0$, so that $F_{eq} = \mathbf{1}^T H$ and $f_{eq} = 0$.

B. Inequality constraint

For the inequality constraint, we reflect on the mathematical meaning of $Hx(k)$. It aims at modifying the state probabilities, and then it corresponds to a variation of probability. Thus, it is reasonable to assume that $[Hx(k)]_i \in [-1, 1] \ \forall i$, which is the maximum allowable variation in probability for any DTMC state. These boundaries can be integrated into the control problem as inequalities constraints, as in (21b). We include the two following set of inequalities in the problem.

$$\begin{cases} Hx(k) \geq -1 \ \rightarrow \ -Hx(k) \leq 1, \\ Hx(k) \leq 1. \end{cases} \tag{24}$$

In matrix form, we obtain $F_{ineq} = [-H; H]$ and $f_{ineq} = \mathbf{1}$.

C. Cost function matrices

We defined the matrices Q_{err}, R and S of the cost function (20) as

$$Q_{err} = \boldsymbol{\beta}, \ R = I, \text{ and } \ S = Q_{err} \tag{25}$$

where $\boldsymbol{\beta}$ is a diagonal matrix that can be tuned to improve the tracking and we set as $\boldsymbol{\beta} = \text{diag}([1, 1, 1, 0, 0, 0])$. This implies that our concern is to exclusively minimize the reference tracking error of the DTMC state.

D. Condensed Matrices

The optimization described above can be written more compactly in matrix form by exploiting the so-called *condensed matrices*. We set the finite control horizon $N = 10$ and derive the condensed matrices for the control problem as

$$\begin{aligned}
\mathcal{X}_c(k) &= \begin{bmatrix} x_a^T(k+1) & \dots & x_a^T(k+N) \end{bmatrix}^T, \\
\mathcal{U}_c(k) &= \begin{bmatrix} u^T(k) & \dots & u^T(k+N-1) \end{bmatrix}^T, \\
\mathcal{A}_c &= \begin{bmatrix} \tilde{A} & \tilde{A}^2 & \tilde{A}^3 & \dots & \tilde{A}^N \end{bmatrix}^T, \\
\mathcal{B}_c &= \begin{bmatrix} \tilde{B} & 0 & \dots & 0 \\ \tilde{A}\tilde{B} & \tilde{B} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \tilde{A}^{N-1}\tilde{B} & \dots & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix},
\end{aligned} \tag{26}$$

so that the system becomes

$$\mathcal{X}_c(k) = \mathcal{A}_c x(k) + \mathcal{B}_c \mathcal{U}_c(k). \tag{27}$$

Moreover, (21a) and (21b) must be rewritten in condensed form as $\mathcal{F}_{c,eq} = I_{N\times N} \otimes F_{eq}$, $f_{c,eq} = I_{N\times 1} \otimes f_{eq}$, $\mathcal{F}_{c,ineq} = I_{N\times N} \otimes F_{ineq}$, $f_{c,ineq} = I_{N\times 1} \otimes f_{ineq}$, as well as the cost function matrices as $\mathcal{Q}_{err} = \text{diag}([Q_{err}, \dots, Q_{err}, S])$ and $\mathcal{R} = \text{diag}([R, \dots, R])$. For more details, please refer to [14].

V. Results

In this section, we simulate the model (7) and assess the performances of our controller.

For our tests, we employ the scenario introduced in Example 1 in Section II. Hence, we consider having 2 target agents (groups G_1 and G_2), which are placed within the environment. This environment is representative of the price of the t-shirts, which the company has the possibility to increase or reduce.

The (members of) target agents are called to decide between red, yellow, or blue t-shirts. Therefore, the DTMC of each target agent has $m = 3$ states, which represent the different colors. The aim of the t-shirt company is to steer the decisions of the target agents toward the red t-shirts. We define the entries of the state probability vector $\Pi(k)$ as the probability of purchasing a red, yellow, and blue t-shirt respectively, and the entries of the environment's states $x(k)$ as percentage price variation (from the initial cost) of the red, yellow and blue t-shirts respectively. By construction, the percentage price variation is limited in the interval $[-1, 1]$.

As an illustrative example, we define the matrices for the linear system defined in (7) as follows. In order to perform a correct comparison between G_1 and G_2, we assume that they have the same transition matrix $Q^{(1)} = Q^{(2)} = Q$. The matrices Q and H are set as

$$Q = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.7 \\ 0.4 & 0.5 & 0.1 \end{bmatrix}, \; H = -\begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.6 & 0 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \tag{28}$$

where the matrix H characterizes the influence of t-shirt prices on the probability of selecting a particular color. Thus, in this case, it assigns a higher weight to the variation price of a specific color in relation to the probability of purchasing that color, relative to the other color choices.

Notably, in real-world settings, the entries of these matrices will typically be unknown, and thus they should be estimated from data. The matrix Q could be trivially estimated through a frequentist approach [15], as well as H, also with the aid of surveys that may help to identify the possible influence that prices have on purchases.

The matrices A and B are set as

$$A = I, \;\; B = I. \tag{29}$$

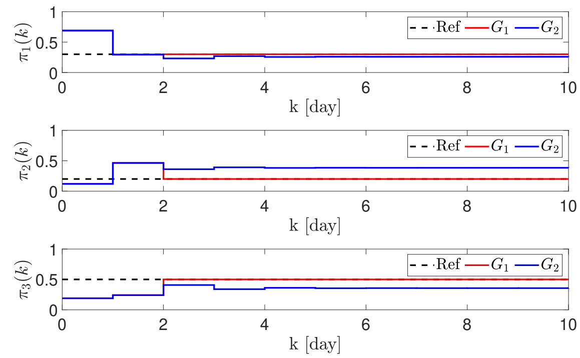

Fig. 1: Evolution of the decision-making process of the target agents G_1 (red line) and G_2 (blue line). G_1 is assumed to be strongly influenced by the environment, according to the model (7), while G_2 is supposed to be unaffected by it. The reference probability to which the controller agent is directing G_1 is denoted by the black dashed line.

and describe the evolution of the t-shirts prices[2].

Lastly, we set

$$\begin{aligned} \Pi_{ref} &= [0.3, \; 0.2, \; 0.5], \\ \Pi_0 &= [0.69, \; 0.12, \; 0.19], \end{aligned} \tag{30}$$

which consists in shifting the overall preference for a future purchase from red to blue t-shirts. Note that we do not define any reference for x, since our goal is to exclusively control the evolution of DTMC. Should there be a need to introduce a reference, one could assign arbitrary values to x_{ref} and set the weights for the tracking error associated with the environment's state in the cost function (20) to be zero.

In order to display different agent's behaviors, we assume that the members of G_1 are strongly influenced by the environment, whereas the members of G_2 are not affected by it.

A. Controller performances

All the simulations were conducted in *Matlab R2023a* for a total of 30 steps, each of which corresponds to a day. Nevertheless, the forthcoming figures will only display the initial 10 steps. This choice is made because no substantial changes were observed beyond this point in the simulations. The evolution of the probability vector state of G_1 and G_2 are shown in Fig. 1, where G_1 is the target agent which is affected by the influence from the environment, while the decision-making process of G_2 remains unperturbed. In simpler terms, the evolution of G_2 represents the normal progression of the DTMC described by the transition matrix Q in (28) and the initial conditions Π_0 in (30), whereas the decision-making process G_1 (although characterized by Q and Π_0 as is the case for G_2) is subject to the influences exerted by the environment. However, before confirming that

[2]Given the matrices A and B that we have designed, we introduce the realistic assumption that the prices of t-shirts remain constant over time unless the company decides to modify them. Moreover, we assume that the choice of altering the price of a specific t-shirt color category does not affect the others.

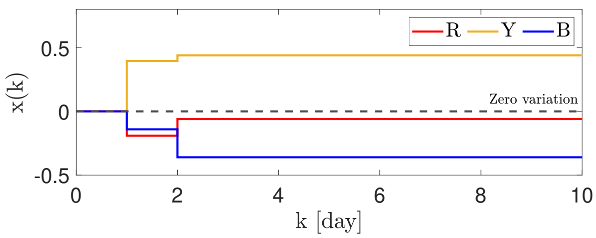

Fig. 2: Percentage variation of the price of different t-shirt colors (R = red, Y = yellow, B = blue).

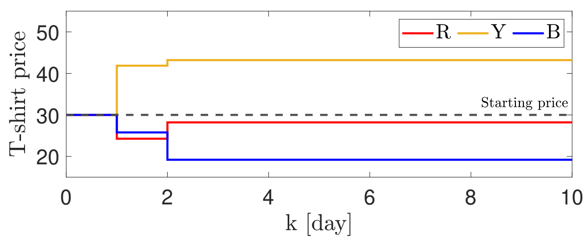

Fig. 3: Price trends of the t-shirts. Starting from the initial price (dashed black line), the evolution of the costs (in euros) of the t-shirts (R = red, Y = yellow, B = blue) is displayed based on Fig. 2.

the controller agent effectively steers the probabilities of G_1 toward the provided reference, we check and confirm that the controlled state $\Pi(k)$ conforms to the definition of a probability vector, meaning that $\sum_i \pi_i(k) = 1 \ \forall k$.
Next, it is interesting to analyze the dynamics of the environment in order to achieve the desired effect on the target agent G_1. Suppose the initial price of each t-shirt is 30 euros, independently of the color. Fig. 2 illustrates the percentage variation in the price of each t-shirt, with a zero variation denoting that the new price remains unchanged. From this plot, we can retrieve the prices (Fig. 3) that the company should impose on the t-shirts. Finally, consider the initial aim to support the purchase of blue t-shirt rather than red or yellow ones and the matrix H (28) of influences, we obtain the intuitive result that the company should decrease the price of blue t-shirts.

VI. Conclusion

In this paper, we have proposed a control framework for an opinion dynamics model, specifically for a Discrete-Time Markov Chain. Different from the common approaches in the literature, we have assumed that the control input is external, and can manipulate the state of a controller agent, which could be the environment. This, in turn, can influence the behavior of the target agents, within a network. We initially demonstrated that our system is controllable, under suitable assumptions, and then we designed an MPC controller to achieve the desired state of the target agent. Next, we exploited a small-scale example from everyday life, to show the outcomes of our controller and to suggest possible applications of our framework. Despite being an introductory work on controlling agents, we believe this work can offer valuable insights into the potentialities of this control scheme. In future works, we will mathematically investigate the properties of our framework, also by relaxing some of the simplifying assumptions and establishing novel conditions for the controllability of the system. Moreover, we will investigate new application scenarios, such as traffic systems.

References

[1] J. Pickett-Baker and R. Ozaki, “Pro-environmental products: marketing influence on consumer purchase decision,” *Journal of Consumer Marketing*, vol. 25, no. 5, pp. 281–293, 2008.

[2] J. Berger, M. Meredith, and S. C. Wheeler, “Contextual priming: Where people vote affects how they vote,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 26, pp. 8846–8849, 2008.

[3] E. Gaetan, L. Giarré, S. Sacone, P. Falcone, and C.-J. Heiker, “Modelling traffic scenarios via markovian opinion dynamics,” in *2023 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[4] A. Dorri, S. S. Kanhere, and R. Jurdak, “Multi-agent systems: A survey,” *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[5] C. Ravazzi, S. Hojjatinia, C. M. Lagoa, and F. Dabbene, “Ergodic opinion dynamics over networks: Learning influences from partial observations,” *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2709–2723, 2021.

[6] P. Bolzern, P. Colaneri, and G. De Nicolao, “Opinion influence and evolution in social networks: A markovian agents model,” *Automatica*, vol. 100, pp. 219–230, 2019.

[7] C.-J. Heiker and P. Falcone, “Decision modeling in markovian multi-agent systems,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 7235–7240.

[8] F. Pasqualetti, S. Zampieri, and F. Bullo, “Controllability metrics, limitations and algorithms for complex networks,” in *2014 American Control Conference*, June 2014, pp. 3287–3292.

[9] F. Dietrich, S. Martin, and M. Jungers, “Control via leadership of opinion dynamics with state and time-dependent interactions,” *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1200–1207, April 2018.

[10] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, “Leader selection in multi-agent systems for smooth convergence via fast mixing,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec 2012, pp. 818–824.

[11] M. L. Puterman, “Chapter 8 markov decision processes,” in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1990, vol. 2, pp. 331–434.

[12] R. Kalman, “On the general theory of control systems,” *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, 1959.

[13] C. E. García, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[14] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[15] D. R. Cox, *Principles of Statistical Inference*. Cambridge University Press, 2006.