Machine Learning Approach to Synthesize a Robust Controller For An Autonomous Agent Through Regret Minimization Technique

Alejandro José Ibáñez de la Puente
dept. Electrical Engineering
Stevens Institute of Technology
Hoboken, USA
aibanezd@stevens.edu

Serban Sabau
dept. Electrical Engineering
Stevens Institute of Technology
Hoboken, USA
ssabau@stevens.edu

***Abstract*—Classical linear quadratic control is mainly centered around linear time invariant systems, and for networks of interconnected systems that might not have a proper transfer function matrix, we propose in this paper a design that enables $\mathcal{H}_\infty$ optimization. The procedure is guaranteed to converge through a convex iterative optimization process that will provide the expected distributed controllers.**

***Index Terms*—Distributed process, LTI system, convex optimization, dynamic controls**

I. Introduction

Machine learning techniques are susceptible to provide reliable, practical solutions for planning and control of the behavior of autonomous systems functioning in real-life environments. Such systems include (semi) autonomous vehicles able to interact with complex, heterogenous environments whereas conditions change throughout time and are difficult to model, thus imposing the need for data-driven decision making and control. However, if machine learning techniques are to be used towards such practical applications, it is paramount that they be accompanied by quantitative certifications of reliability, robustness, and safety since failures during functioning are not permissible. To address these challenges, the proposed research is focused on developing learning-based control strategies for the design of provenly safe, robust autonomous networked systems.

A. *Related Work*

Distributed LQ Control: There has been a great interest in the study of distributed linear quadratic regulator (LQR) over the years, and it has been reflected in the control literature available. There are many areas of study, one of them being multi-agent systems with identical coupled dynamics that are known. In the work performed by Mosebach et. al [1], a derivation of the necessary condition for optimality of the distributed controller generates an optimization problem with a non-convex feasible set. Cao et. al. propose a multi-agent network in their work [2] where a single integrator is assigned to each agent to model its dynamics. It is clearly stated in their work the difficulty of finding the optimal distributed controller when the information of all agents and the graph are unknown. To address the lack of information regarding the network and successfully obtain the controllers, Furieri et. al [3] studied how distributed controllers would converge to a global minimum employing first-order methods for systems that where quadratically invariant.

Online LQ Control: There is a recent line of research studying the application of online learning techniques to the study of linear dynamical systems that have a cots function that varies over time. These works focus on two different scenarios for known and unknown systems. For unknown systems we can face those that are fully observable or partially observable. In this work we will focus on the study of fully observable unknown systems for which Hazan et. al. [4] employed time-varying convex functions with adversarial noises and could derive the regret of $O(T^{2/3})$.

B. *Main Contributions*

In this work we will lay down a two-step model-based approach to learn: the unknown dynamics of a system of autonomous agents, and to compute locally the control sequences that minimizes the sum of the costs for all agents. We will employ a distributed variant of the adaptive-Q algorithm where each agent will be able to adapt its estimated model from the initial controller. Each agent will then be able to solve the optimum controller from a convexification of the feasible set from the synthesis of the LQG controller (based on the agents estimated model).

II. Problem Statement

For illustrative simplicity, we will summarize the main thrust of the proposed research by means of a meaningful example. Consider an autonomous drone: under normal operating conditions, simple and well understood models are sufficient for motion control. However, translating ranging sensors data (e.g. pixels from a camera or point clouds from a lidar) to measurements that are useful for motion-control (e.g. safe distances from terrain obstacles, the navigation path etc.) requires a complicated map that must be learned. As the output of this uncertain map will be fed directly into the safety critical motion-control loop, it is essential to be able to provide guarantees about the accuracy of its output, as well as to understand under what conditions those guarantees are valid. Learning algorithms produce estimates and predictions that are inherently uncertain, therefore when using such algorithms in safety and performance critical control loops, it becomes imperative to explicitly quantify and account for this uncertainty.

A. *Notation*

We will use the following notation in this work:

$[n]$	The set of $\{1, 2, ..., n\}$ for any integer n
A_{ij}	The entry in the i-th row and the j-th column of A
$\mathbb{R}^n$	The space of n-dimensional real vectors
$\|\| \cdot \|\|$	l_2 norm of a vector (matrix)

B. *Distributed Online LQR*

We will consider a network of communicating autonomous agents, where each agent is modeled as a LTI system (e.g. a double integrator with a first order actuator). To each LTI agent we associate a time-varying quadratic costs that is available sequentially. The goal of the group of agents is to collectively (i) identify the unknown dynamics and (ii) compute locally (in a distributed manner) control sequences competitive to that of the best centralized policy in hindsight that minimizes the sum of costs for all agents. This problem is formulated as a regret minimization. We are aiming for a distributed variant of the online LQG algorithm where each agent "learns" its estimated model during an exploration stage. The agent then solves some tailored convexification of the robust LQG controller synthesis (e.g. semi-definite programming (SDP)) whose feasible set is based on the agent's estimated model.

III. Preliminary Ideas and Results

We now lay out the main ideas and formulations to achieve the desired synthesis of the LQG controller through the following methods.

A. *Learning-Q Control*

The Q-parametrization has been long used in learning and adaptive control. The main ideas of these two methods were originated in the 1990's and we will lay now the underlying principles and their application in this work.

*1) **Adaptive-Q Control**:* To better understand this control method, we can try to visualize it as an online Q-design. We can obtain the optimal parameters for our controller thanks to the affine form of these parameters. So the approach is to first stabilize the system with a robust controller and secondly, by small incremental adaptation performed online, improve the performance of the system to obtain the optimal parameters for all the controllers. Our problem is well suited to this control approach since we will be dealing with a fully observable system and the model uncertainty will be minimal. The controller will then be optimized or re-tuned during the online stage since we allow a certain slack in the design criterion and some aspects of the reference signal can be unknown in the initial stage.

The model can be represented as:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Bw_{1,k} \\ y_k &= Cx_k + Du_k + w_{2,k} \end{aligned} \tag{1}$$

The adaptive-Q method can be implemented in discrete time by the restricting the dynamic complexity as stated in the following expression. Let M be a stable dynamic system which output can be adaptively adjusted

$$M : \begin{cases} x_M(k+1) = A_M x_M(k) + B_M r(k) \\ x_M(0) = x_{m,0} \\ s(k) = \Theta(k) x_M(k) \end{cases} \tag{2}$$

Here $x_M \in \mathbb{R}^{n_q}$ is the state of the adaptive Q filter, $A_M \in \mathbb{R}^{n_q \times n_q}$ is a stable matrix with the parameters chosen by design. $\Theta(k) \in \mathbb{R}^{p \times n_q}$ is the adaptation parameter which will be a $p \times n_q$ adaptation matrix.

At this stage we will need to restrict the desired signal that we will use in the control formulation. The aim is to obtain a fast regulator that can reject disturbances. The resulting signal will be $d_k = \begin{bmatrix} y_k \\ u_k \end{bmatrix}$. We now need to determine the implementation of the class of the stabilizing controllers to continue this idea and obtain a state space realization to restrict our control problem.

*2) **Adaptive-Q Algorithm**:* We are going to make the assumption that the stability is guaranteed a priori, which will allow us to focus on the performance of the adaptive controller, and we will not have to deal with the closed-loop stability that typically affects adaptive systems. This is due to our goal to achieve the highest performance.

We are going to adjust Θ_k to minimize the following expression:

$$J(\Theta) = \lim_{N \to 0} \frac{1}{N} \sum_{k=1}^{N} d_k' R d_k, \qquad R = R' \geq 0 \tag{3}$$

We will update Θ_k recursively with steepest descent algorithm:

$$\Theta_{ij,k+1} = \Theta_{ij,k} - \mu \left. \frac{\partial d_k'}{\partial \Theta_{ij}} \right|_{\Theta_k} R d_k$$

Which can be simplified to:

$$\Theta_{ij,k+1} = \Theta_{ij,k} - \mu \gamma_{ij,k}' R d_k; \quad i = [p], \; j = [n_q] \tag{4}$$

In this expression, $\Theta_{ij,k}$ is the ijth element of the matrix Θ_k, and $\gamma_{ij,k}'$ is a column vector that holds the sensitivity functions.

In the equation (4), μ is a design parameter that will contro the adaptation speed of the gradient descent algorithm and wil be set to a small positive constant.

We have described the adaptive mechanism in equation (4) but this doesn't guarantee that we will obtain a bounded Θ_k. I the ideal case when $\mu\gamma'_{ij,k}$ is independent of $\Theta_{ij,k}$, it turns ou that the algorithm displays this property. But we are also abl to guarantee the boundedness of Θ_k following two differen approaches which is either the introduction of a certain leakag in the equation (4) or projecting Θ_k back into a set that i known to be bounded.

The algorithm will be reduced to the following form if we project Θ_k onto a ball denoted as $B(0,\rho)$ (radius ρ, centered in 0):

$$\begin{aligned} \Theta^*_{ij,k+1} &= \Theta_{ij,k} - \mu\gamma'_{ij,k} R d_k, \\ \Theta_{k+1} &= \Theta^*_{k+1} \quad if\ \Theta^*_{k+1} \in B(0,\rho), \\ &= \Theta^*_{k+1} \frac{\rho}{||\Theta^*_{k+1}||} \quad otherwise. \end{aligned} \tag{5}$$

We will also try implementing the leakage approach through the following equation:

$$\Theta_{ij,k+1} = (1-\mu\alpha)\Theta_{ij,k} - \mu\gamma'_{ij,k} R d_k \tag{6}$$

Where the leakage factor is $\alpha \in (0,1)$ that will limit Θ_k to zero. This is due to our preference to have a controller design that isn't modified from the initial estimation.

We will further explore this approach when we get more information regarding the dynamical network.

B. Dynamical Network

To be able to translate the previous expressions into a practical solution for our specific problem, we need to define an architecture for the agent controller and for the information network. We are going to associate a quadratic cost to each LTI agent, and we will provide each one of them with two pieces of information: information from onboard sensor regarding the relative distance to the previous agent on the network z_k, and the control actions taken by the previous agent u_{k-1} which is available and corresponds to an LTI filtering Ψ_k. Since the goal is to achieve the controller synthesis for the group, the dynamical model of the predecessor G_{k-1} is known.

From this network topology we can obtain the following network operator that contains the LTI system and the input z and output u signals. For simplicity we are going to consider a network formed by 3 agents and their known dynamics.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \Psi_1 \\ \Psi_2 & 0 & 0 \\ 0 & \Psi_3 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

It can be seen that the Ψ factor of size 3×3 holds the meaning of the network directed graph adjacency matrix, and the weights of each of its edges is respectively hold by the dynamical systems Ψ_1, Ψ_2 and Ψ_3. The input terminals of the network are defined by the 3×3 K factor. With these

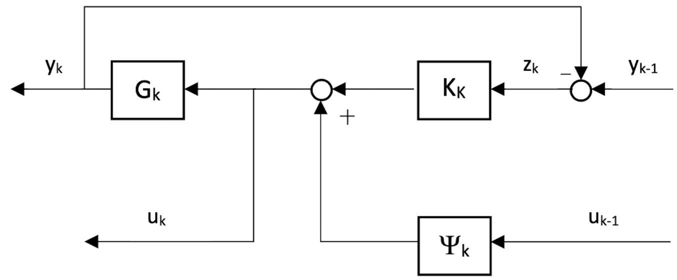

Fig. 1. Information Network

implications, the left factorization of the operator from the input signal z to the output u would be:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} I & 0 & -\Psi_1 \\ -\Psi_2 & I & 0 \\ 0 & -\Psi_3 & I \end{bmatrix}^{-1} \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \tag{7}$$

As we can see, this factorization can describe the network structure completely, and due to the identity matrix entries that are on the diagonal block of of the Ψ factor, this factorization will be unique among all its left factorizations. This mathematical description proposed has the advantage of capturing both the network dynamics and its topology.

To bridge the gap between the network operator and the distributed controller we need to obtain a transfer function that allows that u variables are controllable from the z inputs. To achieve this we will consider the *sparsity pattern* of the associated network operator which outlines a relevant type of constraints used in distributed control. Making use of this paradigm we can transform the decentralized essence of the controller into sparsity constraints to the transfer function matrix. We will prove that this network operator inherited all the main properties of the LTI system at hands.

If we assume that the closed network proposed in this work is a continuous time LTI system, we can then observe that the variables u are controllable from z inputs if we can prove that the controllability matrix stated next has full row normal rank.

$$\begin{bmatrix} I & 0 & -\Psi_1(s) & K_1(s) & 0 & 0 & \\ -\Psi_2(s) & I & 0 & 0 & K_2(s) & 0 & \\ 0 & \Psi_3(s) & 0 & I & 0 & 0 & K_3(s) \end{bmatrix} \tag{8}$$

This expression is a generalization of the controllability criterion proposed by Popov-Belevitch-Hautus and it follows by the standard argument that for the $s \in \mathbb{C}$ that have a normal rank drop in (7), are the same ones that are canceled out in the transfer matrix product from (8).

IV. Numerical Results and Analysis

A. Numerical Algorithm for Sparse Control

As we have seen before, the distributed control problem can be resumed to solving (5) for a $\mathbf{Q}$ that belongs to the chosen set. However, if we recall that $\mathbf{Q} \in \mathcal{RH}_\infty^{m \times p}$ and $\mathbf{Q} = \mathbf{Q}_0 + \widehat{\mathbf{Q}}$, where $\mathbf{Q}_0$ represents a stable stable solution of a simulated model matching problem. This provides us with the means to express $\text{vec}(\widehat{\mathbf{Q}}) = \mathbf{B}\mathbf{x}$ with a $\mathbf{x} \in \mathcal{RH}_\infty^{q \times 1}$ that can be freely chosen. Additionally, we would like to impose

$$\left(Z_1^k + Z_2^k \mathbf{Q}(\infty) Z_3^k\right)^\top \left(Z_1^k + Z_2^k \mathbf{Q}(\infty) Z_3^k\right) \succ 0, \tag{9}$$

If we partition the minimal stable basis, which is denoted by $\mathbf{B}$ by its columns, and we represent each column by the following expression

$$\mathbf{B}_i = \left[\begin{array}{c|c} A_i^{\mathbf{B}} & B_i^{\mathbf{B}} \\ \hline C_i^{\mathbf{B}} & D_i^{\mathbf{B}} \end{array}\right], for\ i \in 1, q$$

If we write the stable Right Coprime Factorization of each $\mathbf{B}_i = \mathbf{N}_{\mathbf{B}_i} \mathbf{M}_{\mathbf{B}_i}^{-1}$, then the minimal realization will be as follows,

$$\begin{bmatrix} \mathbf{M}_{\mathbf{B}_i} \\ \mathbf{N}_{\mathbf{B}_i} \end{bmatrix} := \left[\begin{array}{c|c} A_i^{\mathbf{B}} + B_i^{\mathbf{B}} F_i^{\mathbf{B}} & B_i^{\mathbf{B}} \\ \hline F_i^{\mathbf{B}} & 1 \\ C_i^{\mathbf{B}} + D_i^{\mathbf{B}} F_i^{\mathbf{B}} & D_i^{\mathbf{B}} \end{array}\right] \tag{10}$$

In the previous equation $F_i^{\mathbf{B}}$ can be any matrix that fulfills that all the eigenvalues of $A_i^{\mathbf{B}} + B_i^{\mathbf{B}} F_i^{\mathbf{B}} - sI$ are fully imaginary with negative real part. We can see that because because $\mathbf{B}_i$ are columns of minimal stable basis for a right null space, then $\mathbf{N}_{\mathbf{B}_i}$ are also basis since they belong to a stable RCF obtained from the minimal state-space realizations of every $\mathbf{B}_i$. And they form

$$\widehat{\mathbf{B}} := \left[\begin{array}{c:c:c:c:c} \mathbf{N}_{B_1} & \cdots & \mathbf{N}_{B_i} & \cdots & \mathbf{N}_{B_q} \end{array}\right] \in \mathcal{RH}_\infty^{mp \times q} \tag{11}$$

and hence we can define

$$\overline{\mathbf{B}} := \left[\begin{array}{c:c:c} \widehat{\mathbf{B}}_1 & \cdots & \widehat{\mathbf{B}}_p \end{array}\right] = \left[\begin{array}{c|c} A_{\overline{\mathbf{B}}} & B_{\overline{\mathbf{B}}} \\ \hline C_{\overline{\mathbf{B}}} & D_{\overline{\mathbf{B}}} \end{array}\right] \in \mathcal{RH}_\infty^{m \times pq} \tag{12}$$

and since $\widehat{\mathbf{B}}$ is stable and a minimal basis of the right null space that is desired, then we can express $\text{vec}(\widehat{\mathbf{Q}}) = \widehat{\mathbf{B}}\mathbf{x}$ and define

$$\widehat{\mathbf{Q}} = \left[\begin{array}{c:c:c} \widehat{\mathbf{B}}_1 \mathbf{x} & \cdots & \widehat{\mathbf{B}}_p \mathbf{x} \end{array}\right] = \left[\begin{array}{cc|c} A_{\overline{\mathbf{B}}} & B_{\overline{\mathbf{B}}} \mathcal{D}_p(C_x) & B_{\overline{\mathbf{B}}} \mathcal{D}_p(d_x) \\ 0 & \mathcal{D}_p(A_x) & \mathcal{D}_p(b_x) \\ \hline C_{\overline{\mathbf{B}}} & D_{\overline{\mathbf{B}}} \mathcal{D}_p(C_x) & D_{\overline{\mathbf{B}}} \mathcal{D}_p(d_x) \end{array}\right] \tag{13}$$

Which corresponds to the affine expression of all variable matrices A_x, b_x, C_x, d_x, along with $A_{\overline{\mathbf{B}}}$ and $C_{\overline{\mathbf{B}}}$

In order to find a stabilizing controller using the above DCF and an arbitrary $\mathbf{Q}$ we must satisfy

$$\begin{array}{cl} \min\limits_{\mathbf{Q} \in \overline{\mathcal{S}}_{\mathbf{Q}}} & \left\|\mathbf{T}_1^\epsilon + \mathbf{T}_2^\epsilon \mathbf{Q} \mathbf{T}_3^\epsilon\right\|_\infty, \\ \text{s.t.} & \left\|\mathbf{T}_1^\epsilon + \mathbf{T}_2^\epsilon \mathbf{Q} \mathbf{T}_3^\epsilon\right\|_\infty < 1. \end{array} \tag{14}$$

And we can now express the system fixed part form the previous expression as given by

$$\mathbf{T}^f := \left[\begin{array}{c:c} \mathbf{T}_1 + \mathbf{T}_2 \mathbf{Q}_0 \mathbf{T}_3 & \mathbf{T}_2 \\ \hdashline \mathbf{T}_3 & 0 \end{array}\right] = \left[\begin{array}{c|c:c} A^f & B_1^f & B_2^f \\ \hline C_1^f & D_{11}^f & D_{12}^f \\ \hdashline C_2^f & D_{21}^f & 0 \end{array}\right].$$

We must note that if we let $\mathbf{T}^\epsilon$ to be partitioned partitioned with an F that make of $A + BF - sE$ a policy that is admisible and includes a stable RCF with E_r invertible. Then for the given F with any H that conforms an admissible $A + HC - sE$, we can obtain a DCF over $\mathcal{RH}_\infty$ of $\mathbf{T}_{22}^\epsilon$ that is provided by

$$\left[\begin{array}{c:c} \widetilde{\mathbf{Y}}^\epsilon & -\widetilde{\mathbf{X}}^\epsilon \\ \hdashline -\widetilde{\mathbf{N}}^\epsilon & \widetilde{\mathbf{M}}^\epsilon \end{array}\right] := \left[\begin{array}{c|c:c} A + HC - sE & -B - HD & H \\ \hline F & I & 0 \\ \hdashline C & -D & I \end{array}\right], \tag{15a}$$

$$\left[\begin{array}{c:c} \mathbf{M}^\epsilon & \mathbf{X}^\epsilon \\ \hdashline \mathbf{N}^\epsilon & \mathbf{Y}^\epsilon \end{array}\right] := \left[\begin{array}{c|c:c} A + BF - sE & B & -H \\ \hline F & I & 0 \\ \hdashline C + DF & D & I \end{array}\right]. \tag{15b}$$

Now, before proposing the optimization procedure, if we further simplify the expression in (9) and bring it to a form that is more tractable, recall the expression (13), and partition $\overline{\mathbf{B}}(\infty)$ into its columns

$$\overline{\mathbf{B}}(\infty) = \left[\begin{array}{c:c:c:c:c} D_1^{\overline{\mathbf{B}}} & \cdots & D_i^{\overline{\mathbf{B}}} & \cdots & D_{pq}^{\overline{\mathbf{B}}} \end{array}\right]$$

We can the rewrite (9) and (14) and combine them into the following expression

$$\begin{cases} \min\limits_{A_x, b_x, C_x, d_x, \overline{d}_x, F_i^{\mathbf{B}}, P, \overline{P}, \overline{P}^D} \left\| \begin{bmatrix} T_C + XY - T_A Y - X T_B & T_A - X \\ T_B - Y & I_{n_T} \end{bmatrix} \right\|_* \\ \left\| \mathcal{F}_\ell(\mathbf{T}^f, \widehat{\mathbf{Q}}) \right\|_\infty < 1 \\ (\widehat{Z}_1^k)^\top \widehat{Z}_1^k + \sum_{i=1}^q d_{xi} (\widehat{Z}_1^k)^\top \widehat{Z}_{2i}^k + \sum_{i=1}^q \overline{d}_{xi} (\widehat{Z}_{2i}^k)^\top \widehat{Z}_1^k + \\ + \sum_{i=1}^q \sum_{j=1}^q d_{xi} d_{xj} (\widehat{Z}_{2i}^k)^\top \widehat{Z}_{2j}^k \succ 0, \forall k \in 1 : N_Z, \\ \widehat{Z}_1^k := Z_1^k + Z_2^k \mathbf{Q}_0(\infty) Z_3^k, \\ \widehat{Z}_{2i}^k := Z_2^k \overline{\mathbf{B}}^i(\infty) Z_3^k, \forall i \in 1 : q. \end{cases} \tag{16}$$

This previous expression will be the core of the convex iterative process introduced in this paper. Now, given that a bounded solution to problem (16) exists if the optimization problem is feasible, then the solution can be found by the iterative procedure proposed in Algorithm 1 that involves the convex optimization problem proposed in (16)

B. Numerical Example

Lets consider a set of 5 subsystems that are interconnected in a dynamical network as expressed in Fig. 1 with a ring topology. Then we can define the input-output model of each subsystem as follows

$$\mathbf{y}_{(i \bmod \ell)+1} = \mathbf{\Phi}_{\overline{\mathbf{G}}} \mathbf{y}_{((i-1) \bmod \ell)+1} + \mathbf{\Gamma}_{\overline{\mathbf{G}}} \mathbf{u}_{(i \bmod \ell)+1}, \forall i \in 1 : \ell,$$

where,

Initialization: Solve the linear matrix inequality system, along with the constraint $d_x - \overline{d}_x = 0$, for the variables

$$A_x^0, b_x^0, C_x^0, d_x^0, \overline{d}_x^0, \left(F_i^{\mathbf{B}}\right)^0, P^0, \overline{P}^0, \left(\overline{P}^D\right)^0$$

Using these variables, form T_A^0, T_B^0, T_C^0 and then set $k = 0$ along with $f^0 = \left\|T_C^0 - T_A^0 T_B^0\right\|_*$;

repeat

  if k *mod* $2 < 1$ **then**

    Set $k = k + 1$ and then $\Theta^k = T_B - T_B^{k-1}$;

  else

    Set $k = k + 1$ and then $\Theta^k = T_A - T_A^{k-1}$;

  end

  Solve the optimization problem for $\Big(A_x^k, b_x^k, C_x^k, d_x^k, \overline{d}_x^k,$ $(F_i^{\mathbf{B}})^k, P^k, \overline{P}^k, \left(\overline{P}^D\right)^k$ and use them to form T_A^k, T_B^k, T_C^k;

  Compute $f^k := \left\|T_C^k - T_A^k T_B^k\right\|_*$;

until $f^k - f^{k-1} < \eta_1$ *or* $f^k < \eta_2$;

Algorithm 1: Convex approach to solving (16)

$$\mathbf{\Phi}_{\overline{\mathbf{G}}}(s) := \left[\begin{array}{c|c} A_{\mathbf{\Phi}} - sE_{\mathbf{\Phi}} & B_{\mathbf{\Phi}} \\ \hline C_{\mathbf{\Phi}} & 0 \end{array}\right] = \left[\begin{array}{cc|c} -1-s & 0 & 1 \\ 0 & -1 & 1 \\ \hline 1 & 1 & 0 \end{array}\right]$$

$$\mathbf{\Gamma}_{\overline{\mathbf{G}}}(s) := \left[\begin{array}{c|c} A_{\mathbf{\Gamma}} - sE_{\mathbf{\Gamma}} & B_{\mathbf{\Gamma}} \\ \hline C_{\mathbf{\Gamma}} & 0 \end{array}\right] = \left[\begin{array}{cc|c} 1-s & 0 & 11 \\ 0 & -1 & 4 \\ \hline 1 & 1 & 0 \end{array}\right]$$

so we can obtain the network's TFM which will be stabilizable and have a detectable realization, as follows

$$\overline{\mathbf{G}}(s) = \left[\begin{array}{cc|c} \mathcal{D}_\ell(A_{\mathbf{\Phi}} - sE_{\mathbf{\Phi}}) + \mathcal{D}_\ell(B_{\mathbf{\Phi}})\Xi(1)\mathcal{D}_\ell(C_{\mathbf{\Phi}}) & \mathcal{D}_\ell(B_{\mathbf{\Phi}})\Xi(1)\mathcal{D}_\ell(C_{\mathbf{\Phi}}) & O_{2\ell,\ell} \\ O_{2\ell} & \mathcal{D}_\ell(A_{\mathbf{\Gamma}} - sE_{\mathbf{\Gamma}}) & \mathcal{D}_\ell(B_{\mathbf{\Gamma}}) \\ \hline \mathcal{D}_\ell(C_{\mathbf{\Phi}}) & \mathcal{D}_\ell(C_{\mathbf{\Gamma}}) & O_\ell \end{array}\right] \tag{17}$$

Additionally we can see that the descriptor vector is in fact a concatenation of the vectors that belong to the given realizations for all the subsystems $\mathbf{\Phi}_{\overline{\mathbf{G}}}$ that conform our network. This previous expression is very connected with an improper and unstable TFM given the pole distribution, with a pole located at ∞, another 20 poles in $\mathbb{C}\backslash\mathbb{C}^-$ and 19 poles in $\mathbb{C}^-$.

We are looking for a control law for $\mathbf{K}_u, \mathbf{K}_y$ for

$$\mathbf{u}_{(i \bmod \ell)+1} = \mathbf{K}_u \mathbf{u}_{((i-1) \bmod \ell)+1} + \mathbf{K}_y \mathbf{y}_{(i \bmod \ell)+1}, \ \forall i \in 1:\ell$$

and we are going to assume that $\overline{\mathbf{G}}(s)$ can be approximated to $\mathbf{G}(s) := \mathcal{D}_\ell(\mathbf{\Psi})\Omega$ with

$$\mathbf{\Psi}(s) := \left[\begin{array}{c|c} A_{\mathbf{\Psi}} - sE_{\mathbf{\Psi}} & B_{\mathbf{\Psi}} \\ \hline C_{\mathbf{\Psi}} & D_{\mathbf{\Psi}} \end{array}\right] = \left[\begin{array}{cc|c} 1 & -s & 0 \\ 0 & 1 & -1 \\ \hline 1 & 0 & 4 \end{array}\right]$$

which will allow us to obtain a strong stabilization with feedbacks $F_{\mathbf{\Psi}} = \begin{bmatrix} 1 & 5 \end{bmatrix}$ and $H_{\mathbf{\Psi}}^\top = \begin{bmatrix} -5 & -1 \end{bmatrix}$ that are admissible so we can express

$$\mathbf{G}(s) = \left[\begin{array}{c|c} \mathcal{D}_\ell(A_{\mathbf{\Psi}} - sE_{\mathbf{\Psi}}) & \mathcal{D}_\ell(B_{\mathbf{\Psi}})\Omega \\ \hline \mathcal{D}_\ell(C_{\mathbf{\Psi}}) & \mathcal{D}_\ell(D_{\mathbf{\Psi}})\Omega \end{array}\right] \tag{18}$$

and we obtain admissible feedbacks for both $H = \mathcal{D}_\ell(H_{\mathbf{\Psi}})$, which we can use to compute stable NRCF and obtain a maximum stability radius of $b_{opt} > 0.9925$ for $\mathbf{G}$ which has an upper bound of $\mu(\overline{\mathbf{Q}}) < 0.5609$ from the gap metric among $\overline{\mathbf{G}}$ and $\mathbf{G}$. We can then fix $\epsilon = 0.7$ and the same arguments used in [5] to obtain $\overline{\mathbf{Q}} \in \mathcal{RH}_\infty^{\ell\times\ell}$. We must note the fact that $\mu(\overline{\mathbf{Q}}) < 1$ will force $\det \overline{\mathbf{Q}} \not\equiv 0$ as the opposite would mean that $\exists\ \mathbf{v} \in \operatorname{Ker} \overline{\mathbf{Q}} \cap \mathcal{RH}_\infty^{\ell\times 1}$ that will throw a $\mathcal{H}_2$ unit norm with which we can prove $\mu(\overline{\mathbf{Q}}) \geq 1$.

We can then perform the realization of F and (18) to obtain an RCF similar to the one in (10) that is stable, and can be used together with H to compute $(\mathbf{N}^\epsilon, \widetilde{\mathbf{N}}^\epsilon, \mathbf{M}^\epsilon, \widetilde{\mathbf{M}}^\epsilon, \mathbf{X}^\epsilon, \widetilde{\mathbf{X}}^\epsilon, \mathbf{Y}^\epsilon, \widetilde{\mathbf{Y}}^\epsilon)$ and $\mathbf{T}_1^\epsilon$, $\mathbf{T}_2^\epsilon$ and $\mathbf{T}_3^\epsilon$ from equation (15a) and (15b). Then computing

$$\mathbf{K} = (\Omega\widetilde{\mathbf{Y}}^\epsilon + \widetilde{\mathbf{Q}}\widetilde{\mathbf{N}}^\epsilon)^{-1}(\Omega\widetilde{\mathbf{X}}^\epsilon + \widetilde{\mathbf{Q}}\widetilde{\mathbf{M}}^\epsilon) \tag{19}$$

and

$$\mathcal{F}_\ell(\mathbf{T}^\epsilon, \mathbf{K}) = \mathbf{T}_1^\epsilon + (\mathbf{T}_2^\epsilon\Omega^{-1})\widetilde{\mathbf{Q}}\mathbf{T}_3^\epsilon \tag{20}$$

we can obtain the new DCF over $\mathcal{RH}_\infty$ that will conform the controller stated in (19) and further optimize the $\mathcal{H}_\infty$ norm of (20) over $\widetilde{\mathbf{Q}} \in \overline{\mathcal{S}}_{\widetilde{\mathbf{Q}}}$.

The control law will be achieved from a controller with $\mathbf{\Phi} \in \widehat{\mathcal{S}}_{\Xi(1)}$ and $\mathbf{\Gamma} \in \widehat{\mathcal{S}}_{I_\ell}$. We can then obtain a solution for $\widetilde{\mathbf{Q}}_0 = 0$ with a stable minimal basis obtained for the null space and that can be expressed as in (9), having $q = \ell$ and $\mathbf{N}_{\mathbf{B}_i} = \widetilde{e}_{1+(\ell+1)(i-1)}$, $\forall i \in 1:\ell$, where $\widetilde{e}_i$ corresponds to the vector in the i^{th} position of the canonical basis of $\mathbb{R}^{\ell^2\times 1}$. We can then execute Algorithm 1 with $\widetilde{\mathbf{Q}}(s) = \mathcal{D}_\ell(5.9844)$ that will generate the distributed control laws stated below

$$\mathbf{u}_{(i \bmod \ell)+1} = -2\mathbf{u}_{((i-1) \bmod \ell)+1} + \frac{64.11s + 257.4}{s+4}\mathbf{y}_{(i \bmod \ell)+1}.$$

And obtaining the following outputs of the information network for the iterative optimization of the distributed controller.

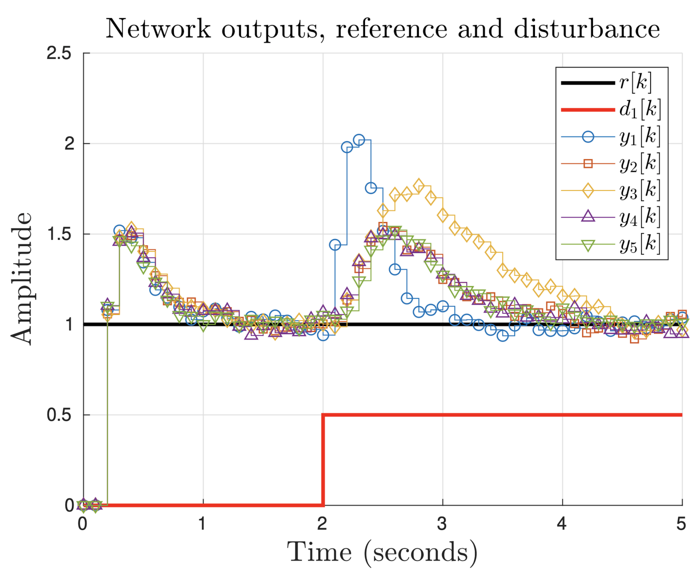

Fig. 2. Iterative optimization

V. CONCLUSIONS

In this paper, we considered the distributed control optimization problem with unknown and decoupled LTI systems that are identical, with time-varying quadratic costs. We demonstrated that it can be tackled if we impose certain constraints on affine expressions of the free parameter that will define the class of the stabilizing controllers. We have relaxed this problem with additional constraints that reduced the problem to solving a norm contraction for a structurally-constrained $\mathcal{H}_\infty$. This problem is solved in an iterative manner through the convex optimization algorithm proposed that is guaranteed to converge.

REFERENCES

[1] A. Mosebach and J. Lunze, "Synchronization of autonomous agents by an optimal networked controller," in 2014 European Control Conference (ECC), 2014, pp. 208–213

[2] Y. Cao and W. Ren, "Optimal linear-consensus algorithms: An lqr perspective," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 40, no. 3, pp. 819–830, 2010.

[3] L. Furieri and M. Kamgarpour, "First order methods for globally optimal distributed controllers beyond quadratic invariance," in 2020 American Control Conference (ACC). IEEE, 2020, pp. 4588–4593.

[4] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in Algorithmic Learning Theory, 2020, pp. 408–421.

[5] Georgiou and M. Smith, "Optimal robustness in the gap metric,"IEEE Transactions on Automatic Control, vol. 35, no. 6, pp. 673–686, 1990.