

Cluster Space LPV Control of Robot Formations

Alejandro S. Ghersin, Juan I. Giribet, and Ignacio Mas,

Abstract—Within the Cluster Space robot formation control scheme, a new approach is presented where a cascaded control scheme is proposed. On one hand, a simple secondary (inner) loop is used to control the formation’s velocity in Robot Space, while on the other hand a quasi-LPV (Linear Parameter Varying) approach is employed for the design of the main (outer) control loop where the geometry and kinematics of the problem are captured by the LPV formulation. Certain aspects of the design such as stability are tackled, and the use of parameter-dependent weighting functions is discussed through a design example tested in simulations in order to illustrate the use of the method.

Index Terms—Cluster Space, Linear Parameter Varying, LPV, Kinematic Control

I. INTRODUCTION

The idea of having multiple robots working together in a coordinated fashion has called the attention of the robotics research community in recent years. Being able to distribute tasks has the potential to increase performance, capabilities and fault tolerance. Among the tools available to tackle the problem of formation control, the Cluster Space Control approach has become appealing to a considerable number of investigators in the past few years. In [1], the Cluster Space approach to Multirobot Systems Control is presented. In this work, dynamics are mostly neglected, treating each robot in the formation as a unit capable of instantly following velocity commands. Nevertheless, the reduced Control Problem is presented where the direct and inverse kinematic transformations needed to transform the so called Robot Space into the so called Cluster Space, are presented.

On the other hand in [2], the Cluster Space dynamic control problem is presented. On top of the direct and inverse kinematic transformations employed in [1], the direct and inverse generalized velocity and force transformations are presented, which allow to approach the full dynamic control problem successfully. In [2], a nonlinear partition control law is proposed in Cluster Space which consists of an approach stemming from manipulator robotics ([3]). A stability proof is presented for the closed loop system as well.

A.S. Ghersin is with Departamento de Ingeniería Electrónica, Instituto Tecnológico de Buenos Aires (ITBA), and with Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET), Buenos Aires, Argentina, e-mail: aghersin@itba.edu.ar. J.I. Giribet is with Departamento de Ingeniería Electrónica and Departamento de Matemática, Facultad de Ingeniería, Universidad de Buenos Aires, Argentina and Instituto Argentino de Matemática “Alberto Calderón” – CONICET, e-mail: jgiribet@fi.uba.ar. I. Mas is with Centro de Sistemas y Control (CeSyC), Departamento de Matemática – ITBA and CONICET, e-mail: iamass@itba.edu.ar.

This research has been supported by grants PICT-2014-2055, PICT-2015-3040 and PICT-2016-2016 from the Argentinean National Research Agency (ANPCyT).

A. Cluster Space/Robot Space Duality

When looking into the problem of controlling a formation of mobile robots, a certain set of *robot coordinates* is given appending together all the coordinates needed to describe the kinematics of each robot.

From the Cluster Space Point of View, a vector of Cluster variables “*c*” is defined which gives a more natural context to specify the state of a robot formation. Together with “*c*”, a vector of appended robot variables “*r*” is defined as well to specify the state of the robot formation. From the Formation point of view, “*r*” has barely any sense at all, as it is merely a collection of variables stemming from each individual member of the formation. In a certain way, the “*r*” variables parallel joint space variables of a robotic manipulator in the sense that they are the variables where the real or natural dynamics evolve. A key aspect regarding this consideration, is concerned with the domain where control action and/or measured variables belong to.

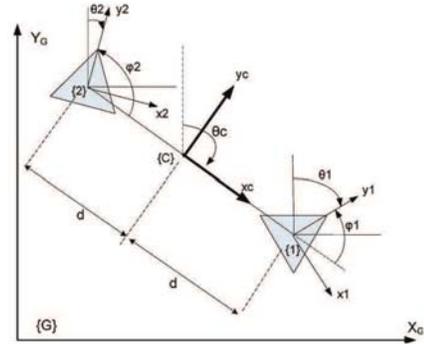


Fig. 1. Two rovers example.

Two Rover Example: An example is presented in order to illustrate the *Cluster Space* approach to describing the state of a Robot Formation. Let the two rover formation of Fig. 1 be described in *Robot Space* by the *r* vector constructed as follows:

$$r = [x_1 \quad y_1 \quad \theta_1 \quad x_2 \quad y_2 \quad \theta_2]^T \quad (1)$$

The definition of *r* is the result of simply appending the two sets of three variables which give for each rover, is position and pose on a given plane with respect to a given inertial reference frame.

In general, for a formation of *m* mobile robots each of them having *n* degrees of freedom (DOF), the *Cluster Space* “*c*” variable is given by the *Forward Kinematics* function defined as:

$$c = f(r) \quad (2)$$

with $r(t)$ and $c(t)$ being vectors in $\mathbb{R}^{m,n}$. In the example, $m = 2$ and $n = 3$. and, the *Forward Kinematics* function is given as follows. Let $d_x = x_1 - x_2$ and $d_y = y_1 - y_2$. According to the geometry depicted in Fig. 1, the *Cluster Space* “ c ” variable is defined as:

$$c = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]^T \quad (3)$$

$$[x_c \ y_c \ \theta_c \ d \ \phi_1 \ \phi_2]^T \quad (4)$$

with

$$f_1 = x_c = \frac{x_1 + x_2}{2} \quad f_2 = y_c = \frac{y_1 + y_2}{2} \quad (5)$$

$$f_3 = \theta_c = \text{atan2}(d_y, d_x) \quad f_4 = d = \frac{\sqrt{d_x^2 + d_y^2}}{2} \quad (6)$$

$$f_5 = \phi_1 = \theta_1 - \theta_c \quad f_6 = \phi_2 = \theta_2 - \theta_c. \quad (7)$$

B. Kinematic LPV Control Cascaded with Velocity Control

Fundamental to the discussion, is the definition of the Jacobian matrix

$$J(r) = \frac{\partial f}{\partial r} \quad (8)$$

such that the *Robot Space* and *Cluster Space* velocities are related as follows:

$$\dot{c} = J(r)\dot{r} \quad (9)$$

The Jacobian matrix for the example is:

$$J(r) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \cos(\theta_c) & -\cos(\theta_c) & \sin(\theta_c) & -\sin(\theta_c) & 0 & 0 \\ -\frac{\sin(\theta_c)}{d} & \frac{\sin(\theta_c)}{d} & \frac{\cos(\theta_c)}{d} & -\frac{\cos(\theta_c)}{d} & 0 & 0 \\ \frac{\sin(\theta_c)}{d} & -\frac{\sin(\theta_c)}{d} & -\frac{\cos(\theta_c)}{d} & \frac{\cos(\theta_c)}{d} & 1 & 0 \\ \frac{\sin(\theta_c)}{d} & -\frac{\sin(\theta_c)}{d} & -\frac{\cos(\theta_c)}{d} & \frac{\cos(\theta_c)}{d} & 0 & 1 \end{pmatrix} \quad (10)$$

Together with a discussion where a model for the dynamics of a formation in *Cluster Space* is derived, in [2], examples are given where the Equations which describe the dynamics of a formation in *Robot Space* are obtained based upon Newton-Euler or Lagrange methods. As a consequence in *Robot Space*, the model for the dynamics of a formation is given by:

$$A(r)\ddot{r} + b(r, \dot{r}) + g(r) = \Gamma \quad (11)$$

with $A(r)$ being the Inertia Matrix, $b(r, \dot{r})$ the vector which accounts for friction and Coriolis terms, and $g(r)$ the vector which accounts for gravitational interaction.

Many practical cases exists in mobile robotics, where a velocity control loop is available for each individual robot. This is equivalent to saying there is a velocity control available in *Robot Space* for the formation. It can be said that readily available from Eq.(11), a Velocity Control loop can be proposed based upon the following model. Let $v = \dot{r}$ be the cluster’s velocity in *Robot Space*. Eq.(11) can be rewritten as:

$$A(r)\dot{v} + b(r, v) + g(r) = \Gamma. \quad (12)$$

A state space model for the system’s velocity dynamics can be written as:

$$\dot{v} = A(r)^{-1} [-b(r, v) - g(r) + \Gamma] \quad (13)$$

In general, “ r ” can be considered a varying parameter for the above Eq., and a “*computed torques*” strategy can be employed in order to synthesize a velocity control law as:

$$\Gamma = g(r) + b(r, v) + A(r)\Gamma_m \quad (14)$$

rendering the following velocity equivalent dynamics:

$$\dot{v} = \Gamma_m \quad (15)$$

In many practical cases ([1], [2], [4]–[7]), the $A(r)$ matrix is constant and the $b(r, v)$ vector turns out to be $b(r, v) = B(r)v$ with a constant $B(r)$ matrix.

Based upon the model of Eq.(15), a stabilizing proportional closed loop velocity control can be proposed with with zero steady state tracking error as follows:

$$\Gamma_m = K(v_{cmd} - v). \quad (16)$$

With this control law, the transfer matrix from v_{cmd} to v turns out to be stable, diagonal and of the low pass kind:

$$\dot{v} = K(v_{cmd} - v). \quad (17)$$

For the *Cluster Space* c variable control, the relationship of Eq.(9) allows for casting the *Cluster Space Kinematic Control Problem* in a form which is suitable for LPV design, accounting for the velocity loop as follows:

$$\dot{c} = J(r)v \quad (18)$$

$$\dot{v} = K(v_{cmd} - v).$$

This model accounts for the *Robot to Cluster Space* problem geometry through the Jacobian matrix, coupled with a previously stabilized velocity secondary loop in a cascaded control scheme.

Two Rover Example (Continuation): In this work, the application of LPV synthesis is tested on the *Two Rover Example* corresponding to Fig.1. In previous research ([8]), this problem was approached through a non-linear control strategy. Note in Eq.(10), that J depends on two *quasi*-parameters, namely d and θ_c . Note as well, that

$$|J| = -\frac{1}{d}. \quad (19)$$

As for practical reasons the case where $d = 0$ can never arise, it is guaranteed that the *Cluster* will not fall in any singular configurations ($|J| \neq 0$, see [2]).

For the two rover formation, the $A(r)$, $b(r, \dot{r})$ and $g(r)$ terms of Eq. (11) that give the *Robot Space* model of the formation’s dynamics are as follows:

$$A(r) = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \quad A_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & I_i \end{bmatrix} \quad (20)$$

$$b(r, \dot{r}) = [b_1 \ b_2]^T \quad b_i = [b_i^x \dot{x}_i \quad b_i^y \dot{y}_i \quad b_i^\theta \dot{\theta}_i]^T \quad (21)$$

$$g(r) = [g_1 \ g_2]^T \quad g_i = [0 \ 0 \ 0]^T \quad (22)$$

with $i = 1, 2$. Related to the model of Eq.(12) note for this particular case that being $A(r)$ constant, $b(r, v_r) = Bv_r$ with a constant B and $g(r) = 0$, a proportional control can be designed for the velocity loop without the need to resort to a “computed torques” strategy in any sense.

II. BACKGROUND ON LPV CONTROL

In this section a brief description of the LPV control synthesis method is presented. Stemming from \mathcal{H}_∞ control theory ([9]), in most approaches ([10], [11]), LPV control sets a framework for control systems design, where a feedback controller is sought for an LPV plant such that stability and performance are guaranteed.

A. Elements of the LPV Setup

Next, the LPV Parameter Trajectories, Open loop Plant, Controller, and Closed Loop Plant, are presented.

LPV Parameter Trajectories: Let the set $\mathcal{P} \subset \mathbb{R}^s$ be, such that for each

$$\rho = (\rho_1, \dots, \rho_s) \in \mathcal{P}, \text{ with } \rho_i \in [\underline{\rho}_i, \bar{\rho}_i] \text{ and } \underline{\rho}_i < \bar{\rho}_i. \quad (23)$$

On the other hand, for some $\bar{\nu} = (\bar{\nu}_1, \dots, \bar{\nu}_s) \in \mathbb{R}^s$ with $\bar{\nu}_i > 0$, all $\nu = (\nu_1, \dots, \nu_s) \in \mathcal{V} \subset \mathbb{R}^s$, are such that $|\nu_i| \leq \bar{\nu}_i$. Throughout this paper, systems state space matrices depend on s -dimensional parameter trajectories evolving in the set $\mathcal{F}_\rho^\nu = \{\rho \in \mathcal{C}^1(\mathbb{R}^+, \mathbb{R}^s) : \rho(t) \in \mathcal{P}, \dot{\rho}(t) \in \mathcal{V}, \forall t \in \mathbb{R}^+\}$ where $\mathcal{C}^1(\mathbb{R}^+, \mathbb{R}^s)$ is the set of continuously differentiable functions of time onto \mathbb{R}^s .

LPV Plant: In order to state the control problem, consider an LPV parameter-dependent plant is given by:

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{A}(\rho(t)) & \mathcal{B}_1(\rho(t)) & \mathcal{B}_2(\rho(t)) \\ \mathcal{C}_1(\rho(t)) & \mathcal{D}_{11}(\rho(t)) & \mathcal{D}_{12}(\rho(t)) \\ \mathcal{C}_2(\rho(t)) & \mathcal{D}_{21}(\rho(t)) & \mathcal{D}_{22}(\rho(t)) \end{bmatrix}}_{\mathcal{G}(\rho(t))} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} \quad (24)$$

where $\rho(t) \in \mathbf{P}$, $\dot{x}, x \in \mathbb{R}^{n_x}$, $w \in \mathbb{R}^{n_w}$ is the disturbance input, $z \in \mathbb{R}^{n_z}$ is the controlled output, $u \in \mathbb{R}^{n_u}$ is the control input and $y \in \mathbb{R}^{n_y}$ is the measurement for control. $\mathcal{G}(\rho)$ is a continuous matrix function of ρ .

LPV Controller: The class of LPV controllers sought is of the form

$$\begin{bmatrix} \dot{x}_k(t) \\ u(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{A}_k(\rho(t), \dot{\rho}(t)) & \mathcal{B}_k(\rho(t), \dot{\rho}(t)) \\ \mathcal{C}_k(\rho(t), \dot{\rho}(t)) & \mathcal{D}_k(\rho(t), \dot{\rho}(t)) \end{bmatrix}}_{\mathcal{K}(\rho(t))} \begin{bmatrix} x_k(t) \\ y(t) \end{bmatrix} \quad (25)$$

where $x_k \in \mathbb{R}^{n_k}$. $\mathcal{K}(\rho)$ is a continuous matrix function of ρ .

LPV Closed Loop: Assuming a controller as in Eq.(25) can be found, the resulting closed loop system is of the form

$$\begin{bmatrix} \dot{x}_{cl}(t) \\ w(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{A}_{cl}(\rho(t), \dot{\rho}(t)) & \mathcal{B}_{cl}(\rho(t), \dot{\rho}(t)) \\ \mathcal{C}_{cl}(\rho(t), \dot{\rho}(t)) & \mathcal{D}_{cl}(\rho(t), \dot{\rho}(t)) \end{bmatrix}}_{\mathcal{F}_l[\mathcal{G}(\rho(t)), \mathcal{K}(\rho(t))]} \begin{bmatrix} x_k(t) \\ w(t) \end{bmatrix}. \quad (26)$$

where $\mathcal{F}_l[\mathcal{G}(\rho(t)), \mathcal{K}(\rho(t))]$ represents the closed loop system. In the sequel, $\mathcal{G}(\rho(t))$, $\mathcal{K}(\rho(t))$ and $\mathcal{F}_l[\mathcal{G}(\rho(t)), \mathcal{K}(\rho(t))]$ shall

be employed to denote respectively the open loop, controller and closed loop matrices of systems, or the systems themselves, according to context. The “ $\mathcal{F}_l(\cdot, \cdot)$ ” notation comes from the Robust Control Framework ([9]) denoting the Lower Linear Fractional Interconnection or Star Product. See [10], [11] where the formulas can be found to calculate the closed loop matrix $\mathcal{F}_l[\mathcal{G}(\rho(t)), \mathcal{K}(\rho(t))]$.

LPV Stability and γ -performance: In order to Synthesize an LPV controller, many approaches can be used (see [10]–[12] among others). Among the cited papers, differences can be found with respect to the kind of Lyapunov matrix picked to carry out optimization on, the performance criterion, and the possibility to tackle (or not) the design, performing semidefinite programming based upon convex optimization with a finite number of Linear Matrix Inequality (LMI) constraints. For this work, a constant Lyapunov matrix was chosen for simplicity ([10], [12]).

Let ρ be the parameter trajectory with $\rho \in \mathcal{F}_\rho^\nu$. The γ -performance LPV control problem restricted to a Single Quadratic Lyapunov Function (SQLF) ([11]), consists in finding an LPV controller as the one of Eq.(25), such that for the closed loop system of equation (26) the following analysis LMI,

$$\begin{bmatrix} \mathcal{A}_{cl}^T X + X \mathcal{A}_{cl}(\rho) & X \mathcal{B}_{cl}(\rho) & \mathcal{C}_{cl}^T(\rho) \\ \mathcal{B}_{cl}^T(\rho) X & -\gamma I & \mathcal{D}_{cl}^T(\rho) \\ \mathcal{C}_{cl}(\rho) & \mathcal{D}_{cl}(\rho) & -\gamma I \end{bmatrix} < 0 \quad (27)$$

is feasible for some symmetric matrix $X \in \mathbb{R}^{2n_x \times 2n_x}$ with $X > 0$. If there exists such matrix the closed loop system is *Quadratically Stable* and its norm is $\|G_{\rho}^{cl}\| < \gamma$ ([11]). Eq.(27) which is based upon a constant Lyapunov matrix, poses a semidefinite programming optimization problem with an infinite number of constraints, namely, inequality (27) must be fulfilled $\forall \rho \in \mathcal{P}$. Many approaches can be found in the literature to tackle the problem of reducing the problem with infinite constraints to one based upon a finite number of constraints. Gridding the parameter set \mathcal{P} and checking if inequality (27) is satisfied on the points of the grid, is one of the most practical ones ([12]). Considering this, gridding has been the chosen approach for this work.

B. Design Specification

Parameters Grid: As it stems from Eq.(18), the LPV dynamics of the system depend on r . This can without loss of generality be posed as depending on c . Eq.(10) shows the Jacobian depends on d and θ_c rendering as a consequence a Quasi-LPV plant, *i.e.* a LPV Plant whose parameters are state variables. The parameter variation set \mathcal{P} was chosen to be the Cartesian product between the specified parameter variation intervals for θ_c and d which turn out to be such that $d \in [1, 10]$ and $\theta_c \in [-\pi, \pi]$.

Mixed Sensitivities: A brief background on the the mixed sensitivities approach is here discussed. Mixed sensitivities are extensively explained in the aforementioned Robust Control literature ([9], [13]). In a linear time invariant (LTI) setup, the key idea is to shape the frequency response of closed loop systems according to criteria originated in classical control theory, extended to multi-input multi-output (MIMO) systems.

Standard classical control is usually concerned with delivering control system designs which render an adequate trade off between tracking, bandwidth and control effort. Through shaping the frequency response of different sensitivity functions, this objectives can be fulfilled.

In order to account for tracking error, it is sought to shape the frequency response of the so called closed loop *sensitivity function*, *i.e.* the mapping from reference signal to tracking error. In order to account for control effort, it is sought to shape the frequency response of the so called closed loop *noise sensitivity function*, *i.e.* the mapping from reference signal to control action. In the LTI case, considering a closed loop \mathcal{H}_∞ norm minimizing tool such as \mathcal{H}_∞ Control is at hand, the use of Weighting Functions is mandatory in order to have the minimizing algorithm achieve the desired result (see [9], chapter 6). In the LPV case, the frequency response LTI key ideas carry over in order to achieve similar goals.

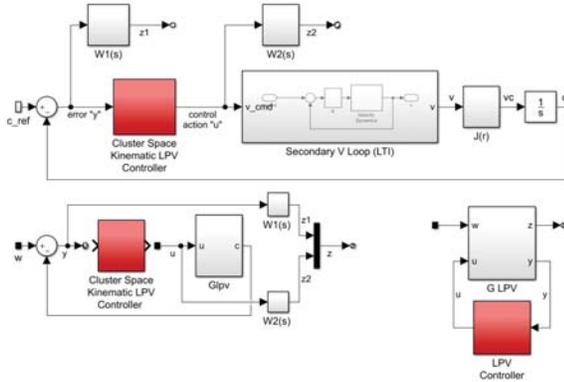


Fig. 2. Augmented Plant for the LPV synthesis.

The design through mixed sensitivities employing weighting functions is illustrated in Fig.2 where three block diagrams can be seen. A block representing the controller, which is common to the three, is highlighted in red. The bottom right diagram, shows the LPV augmented plant $\mathcal{G}(\rho)$ corresponding to Eq.(24) with the LPV controller corresponding to Eq.(25) forming the $\mathcal{F}_l[\mathcal{G}(\rho), \mathcal{K}(\rho)]$ mapping from “ w ” to “ z ”. The $\mathcal{G}(\rho)$ matrix is fed to LPV control synthesis algorithm which evaluates feasibility and returns the state space representation of the controller.

The top and bottom left diagrams of Fig.2, show the plant with the controller and the weighting functions W_1 and W_2 . The top diagram shows the system consisting of the cluster of robots with the secondary LTI velocity loop together with the $J(r)$ Jacobian and the integrator rendering a model as proposed in Eq.(18). The system represented by this Eq. is summarized through the “ G_{lpv} ” block in the bottom left diagram, where the Weighting Functions can be seen as well.

Weighting Functions: Consider for a given Sensitivity Function as the mapping from w to y (Fig.2, bottom left diagram). It is usual for this mapping to seek a high pass frequency response in order to achieve zero steady state tracking error to constant references. As discussed in [9] (Ch.6) for the LTI case, a W_1 of a low pass kind renders through the \mathcal{H}_∞ synthesis method a controller that shall provide integral

action achieving the goal. As a consequence, the proposed W_1 weighting function is:

$$W_1(s) = \frac{10}{s + 0.01}. \quad (28)$$

The “10” factor serves as a parameter for tuning the design.

On the other hand, in order to limit the control bandwidth, high frequency content of the control action signal u should be penalized. This is usually fulfilled with a weighting function W_2 as follows ([9], Chap. 6):

$$W_2(s) = \frac{0.5s}{s + 200}. \quad (29)$$

The “0.5” factor and the pole at “200” serve as a parameters for tuning the design as well.

It must be pointed out, that a slight abuse in the jargon is committed when speaking about frequency response in the LPV case. Rigorously speaking, frequency response considerations in the LPV case are usually made on a point-wise basis, that is, holding the ρ parameter vector constant. Nevertheless, this LPV frequency response considerations prove useful in practice.

LPV Weighting Functions: A second set of Weighting Functions has been proposed for this work, in order to take advantage of the LPV tools. Both Weighting functions were modified incorporating the “size” of the Cluster d as a parameter for the Weights. The W_1^{pv} weight, was changed to

$$W_1^{pv}(s) = \frac{5 + 0.5d}{s + 0.01}. \quad (30)$$

This modification serves the purpose of requiring a more aggressive response when the formation is “expanded”. The W_2^{pv} weight, was changed to

$$W_2^{pv}(s) = \frac{0.5s}{s + [(1 + \frac{d}{10})200]}. \quad (31)$$

This modification serves the purpose of allowing control action with higher frequency contents when the formation is “expanded” as well. In Eqs.(30) and (31), we incur in a slight abuse of notation as the Laplace “ s ” operator is concerned, for the sake of clarity. These are parameter varying weighting functions.

The optimization problem is based upon a grid resulting from the Cartesian product of the following finite sets:

$$d^{set} = \text{logspace}(0, 1, 10), \quad \theta_c^{set} = \text{linspace}(-\pi : \frac{15}{180}\pi : \pi).$$

III. RESULTS AND DISCUSSION

To carry out all simulations, it was assumed that both rovers are equal with their mass being $1Kg$ and their inertia being $0.1Kgm^2$. Hence, the $A(r)$ matrix of Eq.(20) is $A(r) = \text{diag}([110.1110.1])$ and the $b(r, \dot{r})$ vector of Eq. (21) is $b(r, \dot{r}) = 0$. In all cases, the gain of the secondary or inner loop is picked such that the transfer function from commanded to fulfilled velocity in *robot space* is equal to $\frac{20}{s+20}$.

Two simulations were carried out. As it will be illustrated through Figures 3 and 6, the controllers are commanded a step from 1 to 5 meters in “ d ” at time $t = 10sec.$. A pulse from 0 to 90 degrees in “ θ_c ” at time $t = 1sec.$ and another such pulse

at time $t = 15\text{sec}$. are commanded as well. The purpose of such reference signals to be tracked is testing the controllers' performance with the formation contracted ($d = 1$) and with the formation expanded ($d = 5$).

The first simulation was carried out in order to test the response of an LPV kinematic controller based upon the constant weighting functions of Eqs. (28) and (29), which shall be called “LPV1” controller in the sequel for brevity. This simulation carries out a comparison between the response rendered by the LPV1 controller and the response rendered by PI Kinematic Controller tuned “*by hand*” which shall be simply called “PI” controller in the sequel. See Figures 3, 4 and 5. As it can be seen, the transient response achieved by both controllers is very similar. In *Cluster Space*, minor differences in the variables of interest can be seen (Fig. 3) while the control action is fairly similar as seen in both *cluster space* (Fig.4) and *robot space* (Fig.5).

The second LPV controller tested through simulations, which shall be called “LPV2” controller in the sequel for brevity, is the one based upon the parameter varying weighting functions of Eqs.(30) and (31). Considering the performance of the PI controller is very similar to the performance the LPV1 controller, in the second simulation a comparison is carried out between the PI controller and the LPV2 controller.

The aspect of the performance rendered by the LPV2 controller that should be noted in this case, is its ability to provide a truly different transient behavior as “ d ” changes its value. Note in the lower curve of Fig.6, how the PI controller renders a more aggressive response during the first pulse starting at $t = 1\text{sec}$. with $d = 1$ (formation contracted), while the LPV2 controller renders a more aggressive response during the second pulse starting at $t = 15\text{sec}$. with $d = 5$ (formation expanded). The different transient behavior in the control action can be noticed as well in Figures 7 and 8.

Another relevant difference between between controller LPV1 and the PI on one side, and controller LPV2 on the other side must be noted. When looking at the velocities commanded by both the PI and the LPV1 controllers, the $V_{\theta_c}^{comm}$ signal is the same for the pulses at $t = 1\text{sec}$. and $t = 15\text{sec}$. (lower curve Fig.4). Translated to *robot space*, notice the different commanded velocities depending on whether the formation is contracted or expanded in order to fulfill the same $V_{\theta_c}^{comm}$. See Fig.5. It can be said that the higher *robot space* commanded velocities represent an adequate adaptation to the change in the geometry which takes place when “ d ” goes from 1 to 5.

On the other hand, based upon the parameter dependent weighting functions, controller LPV2 not only renders an adaptation to the aforementioned change in the geometry which takes place when “ d ” goes from 1 to 5, but it also renders a more aggressive control in cluster space which can be seen in the $V_{\theta_c}^{comm}$ of Fig.7.

IV. CONCLUSION AND RESEARCH DIRECTIONS

The outcome of the application of LPV methods to Cluster Space Kinematic control of robot formations is satisfactory as it shows the many possibilities that can be exploited by the application of more advanced techniques to an increasingly

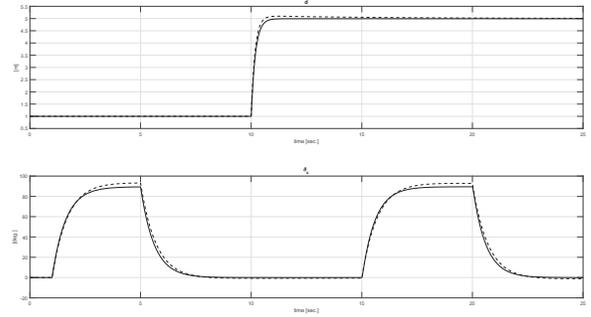


Fig. 3. Cluster Space variables, LPV1 vs. PI (full vs dashed line).

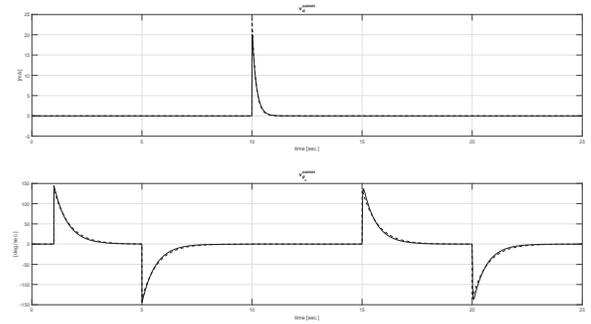


Fig. 4. Control Action in Cluster Space, LPV1 vs. PI (full vs dashed line).

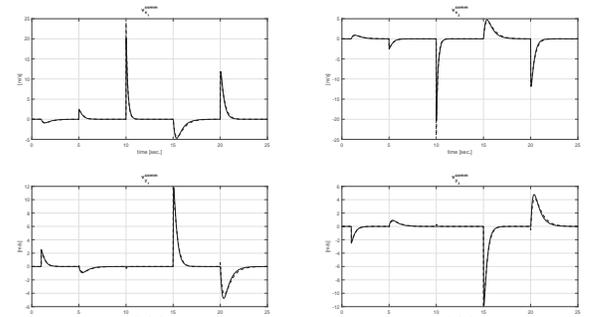


Fig. 5. Control Action in Robot Space, LPV1 vs. PI (full vs dashed line).

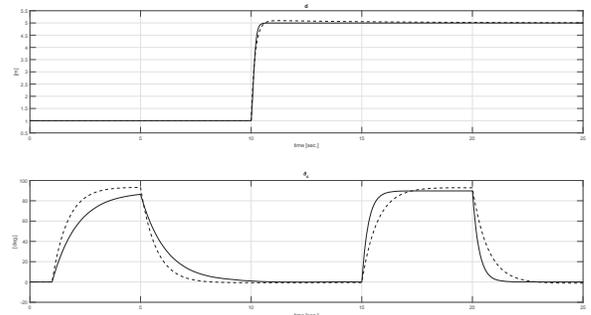


Fig. 6. Cluster Space Variables, LPV2 vs. PI (full vs. dashed line).

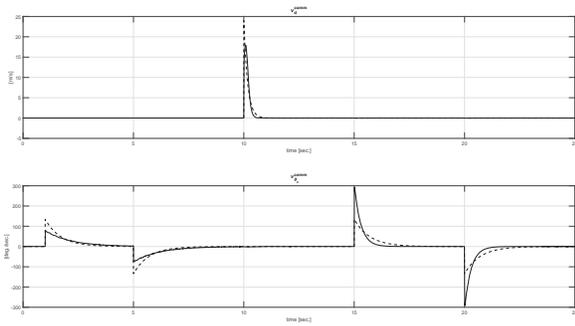


Fig. 7. Control Action in Cluster Space, LPV2 vs. PI (full vs. dashed line).

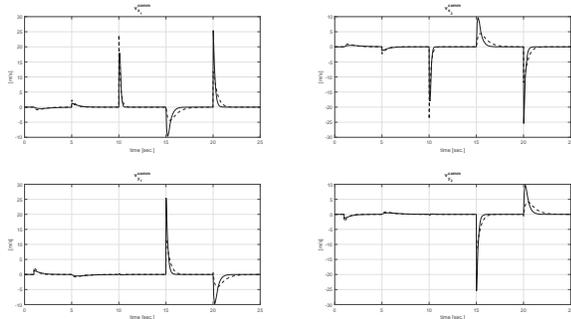


Fig. 8. Control Action in Robot Space, LPV2 vs. PI (full vs. dashed line).

appealing field of research. The design illustrated that it is possible to guarantee stability and performance through this method.

A brief summary of the achievements is the following. On one hand a comparison was presented between “*Controller LPVI*” and a *hand tuned* PI controller delivering very similar performance and transient behavior. The positive aspect of employing LPV control in this case, is the fact that it guarantees stability and performance of the closed loop system. The use of LTI weighting functions and a mixed sensitivities approach, have shown that it is relatively straightforward to synthesize a controller for a formation of robots through the cluster space

ACKNOWLEDGMENT

The research presented in this article was partially supported by ITBA through grant ITBACyT 28, and the *Agencia Nacional de Promoción Científica y Tecnológica* of Argentina through grants PICT-2014-2055 and PICT-2016-2016.

approach. On the other hand a comparison was presented between controller LPV2 and the PI controller, demonstrating that the use of parameter dependent weighting functions for controller design, can achieve an even more flexible adaptation to changes in the formation’s operating parameters.

This result is encouraging enough to consider the versatility of LPV control for other purposes in the future. Improvements to the design can be stated as future research directions including the use of parameter dependent Lyapunov Functions, and LPV techniques to account for control with actuator saturation. Nonlinear approaches could be further investigated as well.

REFERENCES

- [1] C. A. Kitts and I. Mas, “Cluster space specification and control of mobile multirobot systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 207–218, April 2009.
- [2] I. A. Mas and C. A. Kitts, “Dynamic control of mobile multirobot systems: The cluster space formulation,” *IEEE Access*, vol. 2, pp. 558–570, 2014.
- [3] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall, 2005.
- [4] I. Mas, S. Li, J. Acain, and C. Kitts, “Entrapment/escorting and patrolling missions in multi-robot cluster space control,” *Intelligent Robots and Systems. IEEE/RSJ International Conference on*, pp. 5855–5861, Oct 2009.
- [5] I. Mas, P. Moreno, J. I. Giribet, and D. V. Barzi, “Formation control for multi-domain autonomous vehicles based on dual quaternions,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 723–730.
- [6] C. D. Pose, J. I. Giribet, and A. S. Ghersin, “Hexacopter fault tolerant actuator allocation analysis for optimal thrust,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 663–671.
- [7] J. I. Giribet, C. D. Pose, A. S. Ghersin, and I. Mas, “Experimental validation of a fault tolerant hexacopter with tilted rotors,” *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 7, no. 2, 2018. [Online]. Available: <http://ijeetc.com/>
- [8] I. Mas, C. Kitts, and R. Lee, “Model-based nonlinear cluster space control of mobile robot formations,” in *Multi-Robot Systems*, T. Yasuda, Ed. Rijeka: InTech, 2011, ch. 4. [Online]. Available: <https://doi.org/10.5772/12923>
- [9] K. Zhou, *Essentials of Robust Control*. Prentice-Hall, 1998.
- [10] A. Packard, “Gain scheduling via LFTs,” *Systems and Control Letters*, vol. 22, pp. 79–92, 1994.
- [11] F. Wu, “A generalized LPV system analysis and control synthesis framework,” *International Journal of Control*, vol. 74, no. 7, pp. 745–75, 2001.
- [12] A. Hjartarson, P. Seiler, and A. Packard, “Lpvttools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems,” *IFAC-PapersOnLine*, vol. 48, no. 26, 2015.
- [13] R. S. Sánchez Peña and M. Sznaier, *Robust Systems Theory and Applications*. John Wiley & Sons, Inc., 1998.