A New Procedure for Tuning an Allocator and Designing a Robust High-Level Control Law for Over-Actuated Systems. ⋆

Hélène Evain * **Mathieu Rognant** ** **Daniel Alazard** ***
Jean Mignot ****

* *Ph.D student, ONERA, Toulouse, France (e-mail: helene.evain@onera.fr).*
** *Research Scientist, ONERA, Toulouse, France*
*** *Professor, ISAE SUPAERO, Toulouse, France*
**** *AOCS Senior Expert, CNES, Toulouse, France*

Abstract: This paper presents a new integrated procedure to tune a control law for over-actuated mechanical systems that may encounter singularities. First, the allocator that divides the commands among the actuators is tuned thanks to a genetic optimization algorithm, that computes the optimal values of its parameters. Then, the open-loop system including the allocator is identified and a robust closed-loop controller is computed with the structured H_∞ method. Indeed, near singularities, the system and the allocator may create errors to deviate from these points or create delays to reconfigure the actuators, hence there is a need to create a closed-loop controller robust to these characteristics and to parameter variations. This procedure is carried out on a planar redundant robotic manipulator example. Simulations of the behaviours of the open-loop system with the allocator and in a closed loop are presented.

Keywords: Allocation, Guidance, Navigation and Control, Design Methodologies, Robot manipulators

1. INTRODUCTION

Controlling a mechanical input-redundant system is a common problem in the fields of robotics and aerospace. It often requires a control allocation strategy in order to distribute the total control command among the actuators while respecting some constraints like saturation, singularity avoidance or algorithm complexity. In addition, a high-level controller is most of the time necessary to regulate the system with some fixed performance.

Two approaches stand out to design the control law of an over-actuated system. The first one is to deal with the closed-loop dynamics and the allocation in only one step by using optimal control (see Lewis and Syrmos (1995)). The second approach is to design the high-level controller and the control allocation separately, each one meeting their own distinct objectives. The block-diagram representing this strategy is given in Fig. 1. The high-level controller sets the dynamics of the closed-loop system and calculates the total amount of control commands sent to the allocator that distributes it among the actuators. In this strategy, as mentioned by Johansen and Fossen (2013), the allocator usually deals with the actuator failures, saturations, and redundancy and the high-level controller does not have access to this information. Nevertheless, if the control command required by the high-level controller is impossible to handle for the allocator, or if the allocator has a dynamics that can interfere with the desired closed-loop dynamics, then the stability and/or the performance can be affected. This has been studied by researchers like Page and Steinberg (1999) who have compared the performance of different high-level controllers and allocators in the case of the control of an aircraft, or by Buffington et al. (1998) who studied the zero dynamics of the allocator related to the stability in nonlinear dynamic inversions. In addition, the global stability of the closed-loop containing the high-level control law and the allocator is only proved in special cases. As explained by Hu et al. (2014), asymptotic stability can be theoretically proved for both the allocator and the high-level controller separately, but rarely for the entire closed-loop. This is the case in the article of Johansen (2004) that shows the global exponential convergence of a closed-loop comprising a stable controller and an asymptotically optimal control allocation.

Therefore, because of these drawbacks, one can prefer to choose the first approach. However, as explained by Härkegård and Glad (2005), having two controllers can facilitate the tuning since changing a parameter will not affect both the desired dynamics and the allocation, and in case of actuator failures for instance, only the allocator has to change. It can be noted that both approaches give the same performance if the parameters of the laws are set in certain ways as demonstrated by Härkegård and Glad (2005).

In this paper, we want to keep the modularity capabilities of having two control laws, and therefore a new procedure is proposed to tune first the allocator, and then the high-level control law. This control law takes into account the

⋆ This work was supported by ONERA and CNES.

10.1016/j.ifacol.2017.08.1827

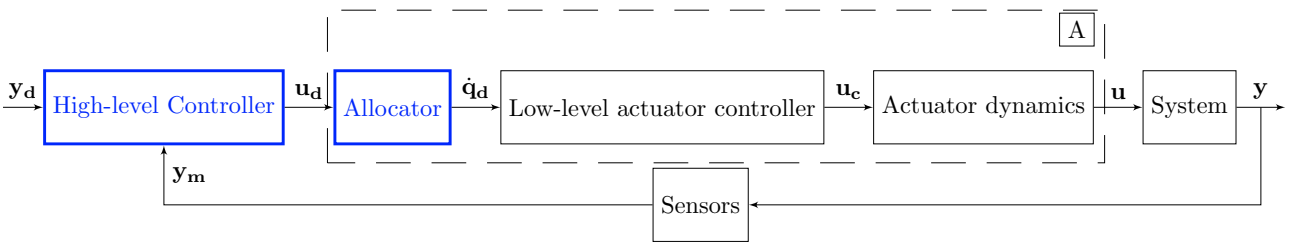

Fig. 1. Block diagram of a typical control strategy of an over-actuated system.

properties of the allocator, in order to deal with the issues previously noted. To describe this procedure, a general nonlinear system is considered (1). In this paper, vectors are written in bold small letters and the matrices are in bold capital letters.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \dot{\mathbf{y}} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \end{cases} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^l$ is the state, $\mathbf{u} \in \mathbb{R}^n$ the control input, $\mathbf{y} \in \mathbb{R}^m$ the controlled output, $\mathbf{f}, \mathbf{g}$ differentiable functions. This means that there are n actuators acting on the system. If the actuator dynamics is added in the system and is assumed linear, then the system becomes linear with respect to the input, which gives the system (2) where $\mathbf{x}$ is the state vector augmented with the actuator internal states.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u_c} \\ \dot{\mathbf{y}} = \mathbf{g}(\mathbf{x}) \end{cases} \tag{2}$$

with $\mathbf{G} \in \mathbb{R}^{l \times n}$ a matrix function. Let's still assume $\mathbf{x} \in \mathbb{R}^l$, $\mathbf{u_c} \in \mathbb{R}^n$, and $\mathbf{y} \in \mathbb{R}^m$. $n \geq m$ for an over-actuated (redundant) system.

Then, as shown in Fig. 1, the high-level controller gives the command input $\mathbf{u_d}$ that the actuators have to create on the system so that $\mathbf{y_d}$ can be imposed. The allocation problem we consider here comes down to inverting a relation of type (3) with constraints depending on the application.

$$\mathbf{u_d} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

with $\mathbf{q} \in \mathbb{R}^n$, and $\mathbf{J} \in \mathbb{R}^{m \times n}$ the Jacobian matrix of a function $\mathbf{h}$ that usually links the actuator variables to some state variables of the system. This function depends on the system. For clarity, an application on the redundant manipulator robot is described later in the paper. $\mathbf{J}$ is rectangular in redundant systems and can become rank-deficient depending on the values of $\mathbf{q}$; in these cases the actuators are in a singularity, which means that there is at least one direction along which they cannot create a motion. Then, if $\dot{\mathbf{q}}_\mathbf{d}$ is not equal to $\mathbf{u_c}$, a low-level actuator controller is required.

The goal of this paper is to show a new procedure to design a closed loop that controls an over-actuated system, assuming that the allocator structure is fixed, but that its parameters can be optimized, and to compute a robust control law that takes into account the properties of the allocator. Indeed, in our view, the goal of the allocator is to provide a control command to the system $\mathbf{u}$ so that the whole block "A = Allocator+Low-level actuator controller+Actuator dynamics" in Fig. 1 is linearized. However, the allocator may have to create errors to avoid the singularities or create delays when using the null space of $\mathbf{J}$ to reconfigure the actuators, hence creating nonlinearities. To become robust to the lack of performance when faced to singularities in terms of time response and decoupling, a robust control law is chosen.

A typical issue in allocation is the input $\mathbf{u}$ saturation, corresponding to physical limitations of the actuators. If this aspect is not tackled, unstability may arise. As mentioned in the book by Hu and Lin (2001), two main strategies stand out to deal with actuator saturation. The first one uses anti-windup techniques as in the paper by Morabito et al. (2003). They add feedback loops in the control law to prevent the saturations. The second strategy takes into account this constraint from the control law design phase (see Sun et al. (2017)) and can use the null controllable region studies (see Hu and Lin (2001)) to tune the control laws. In our approach, the saturation can be dealt with from the design phase of the allocator, however, no theoretical proofs are given for the avoidance of the saturations, hence anti-windup techniques may be added if needed.

This article begins by presenting a robotic manipulator model. Throughout the paper, the different concepts will be demonstrated and applied to this system. Then, a numerical method to tune an allocator is presented. The temporal response of block "A" is next identified as a linear transfer function, with nonlinearities and uncertainties associated. Finally, the design of a robust control law as the high-level controller that takes the allocator properties and the remaining nonlinearities into account is presented.

2. PRESENTATION OF THE APPLICATION

The same example as in the article of Evain et al. (2016) is studied : a 2-DOFs (degrees of freedom) rigid planar robotic manipulator with three arms and revolute joints. The output variables to be controlled are the end-effector position (y_1, y_2) and the controlled parameters are the joint relative angular rates $\dot{\mathbf{q}} = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$. The lengths of the arms are noted l_1, l_2 and l_3, all equal to 0.4m.

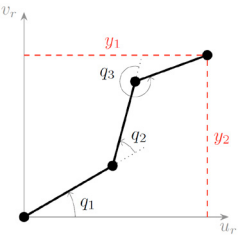

Fig. 2. Parametrization of the system studied

The dynamics of the system is given by equation (4) from the book of Khalil and Dombre (2004).

$$\ddot{\mathbf{q}} = -\mathbf{A}^{-1}(\mathbf{q})\mathbf{C_q}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{A}^{-1}(\mathbf{q})\mathbf{u} \quad (4)$$

with $\mathbf{A}$ the inertia matrix and $\mathbf{C_q}$ the quadratic velocity vector including the Coriolis and the centrifugal forces. The joints are controlled by imposing the torque vector $\mathbf{u}$.

Then, a low-level controller is added. In this paper, it is considered already fixed. It computes: $\mathbf{u_c} = \mathbf{K}(\dot{\mathbf{q}}_\mathbf{d} - \dot{\mathbf{q}}_\mathbf{m}) + \mathbf{C_{qe}}(\dot{\mathbf{q}}_\mathbf{m}, \mathbf{q_m})$ with $\mathbf{K}$ a matrix gain chosen as $\mathbf{K} = k_w\mathbf{A}(\mathbf{q_m})$, $\mathbf{C_{qe}}$ the estimated vector $\mathbf{C_q}$, $\dot{\mathbf{q}}_\mathbf{m}$ and $\mathbf{q_m}$ the measured joint angular rates and positions. The gain k_w is tuned to ensure a fast dynamics that will not interfere with the other closed-loop dynamics of the allocator and of the high-level controller. It is then assumed for simplicity that $\mathbf{u} = \dot{\mathbf{q}}_\mathbf{d}$ in first approximation. More precise models can be integrated.

The kinematic equations are given by equations (5).

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{h}(\mathbf{q}),\ \dot{\mathbf{y}} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (5)$$

The goal of the allocator is then to compute $\dot{\mathbf{q}}_\mathbf{d}$ so that $\dot{\mathbf{y}} = \mathbf{u_d}$ while respecting constraints, and where $\mathbf{u_d}$ is given by the high-level controller (see the block diagram 1: $\mathbf{u_d} = \dot{\mathbf{y}}_\mathbf{d}$). However, in our application, only the kinematic equations are considered for designing the allocator and the high-level controller, not the whole dynamics of the system (used only for the final simulation). Indeed, it is assumed that the low-level controller linearizes the dynamic nonlinearities with the term $\mathbf{C_{qe}}(\dot{\mathbf{q}}_\mathbf{m}, \mathbf{q_m})$ and the synthesis is carried out on the following simplified system (Fig. 3).

In this paper, the procedure presented is applied to this manipulator to show a simple possible implementation.

3. OPTIMISATION OF THE PARAMETERS OF THE ALLOCATOR

In this section, the optimization can be carried out for any allocator with parameters to tune, but here one specific allocator is chosen. It should be noted that the method proposed here is efficient only if tuning is not possible analytically, for instance if the problem is not convex or has many parameters. Only the open-loop model between $\dot{\mathbf{y}}_\mathbf{d}$ and $\dot{\mathbf{y}}$ (subsystem "A" in Fig. 3) is considered.

3.1 Presentation of the problem

First, the requirements that our allocator has to meet are defined :

- Remain stable;
- Avoid/escape singularities;
- Limit the output errors;
- Can be calculated in real-time in space applications.

Many allocators have been proposed in the literature for the past decades. For a comprehensive survey of usual allocation methods (also called kinematic control), see the work of Siciliano (1990) and Siciliano and Sciavicco (2000) for instance. In order to avoid singularities, Singular Robust Inverse methods have been proposed in particular by Wampler (1986) and Nakamura and Hanafusa (1986), which give allocators of the type (6).

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \mathbf{W})^{-1}\dot{\mathbf{y}} \quad (6)$$

with $\mathbf{W}$ a matrix to tune. Other methods from the pseudo-inverse include the gradient-based method (using null-motion) (see Yoshikawa (1985)) and the Extended Jacobian described by Baillieul (1985). Methods deriving from optimization strategies requiring complex solving algorithms also exist (see Fossen and Johansen (2006)).

The allocator detailed in the paper by Evain et al. (2016) is chosen here because it can be computed in real-time in space applications and has the potential of verifying the other constraints if correctly tuned. It uses the formalism of the Extended Kalman Filter (EKF) and it contains a new formulation of the kinematic equations that helps escaping or avoiding singularities. Equality constraints can be directly added in the model in a flexible way, and inequality constraints (as actuator saturations) may be added whether as an additional step minimizing a quadratic problem like in the paper by Simon and Simon (2006) or directly as equality contraints decreasing the number of calculations but with a loss of modelling precision. Such an EKF-based allocator involves a reduced set of tunable parameters $\mathbf{P}_{alloc}$. Stability (or convergence) results exist (Song and Grizzle (1995)) but can make the tuning of $\mathbf{P}_{alloc}$ difficult. In this paper, a new way to tune $\mathbf{P}_{alloc}$, based on genetic algorithms, is presented.

Usually, the response of the system between $\dot{\mathbf{y}}$ and $\mathbf{u_d}$ is nearly linear if the allocator is correctly tuned (see figure 3). When near singularities, this linearization can be degraded and errors can occur, whether by delays or bad decoupling between the axes. In this part, the aim is to find the appropriate parameters that achieve the best linearization, and also to quantify these errors due to the singularities so as to take them into account in the high-level controller synthesis.

For this numerical tuning, since analytical tuning is assumed impossible, a set of typical maneuvers has to be considered. Indeed, it is assumed that the tasks of the robot are defined precisely and that the movements on which performance is sought are known. In this paper, it is chosen to simulate maneuvers from a given initial position $\mathbf{q_0} = \left[\frac{5\pi}{6}\ \frac{2\pi}{3}\ \frac{2\pi}{3}\right]$ where $\mathbf{y_0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ towards the saturation in position (at $||\ \mathbf{y_d}\ || = l_1 + l_2 + l_3$) in every direction θ_i of the plane ($\theta_i = \frac{2ki}{N}, i = 0, ..., N-1$), with a step command on $\dot{\mathbf{y}}_\mathbf{d}$: $\dot{\mathbf{y}}_\mathbf{d} = \left[\frac{\Delta y}{T_s}\cos(\theta_i)\ \frac{\Delta y}{T_s}\sin(\theta_i)\right]^T$ with $T_s = 0.01$s the sampling period of the allocator and Δy a given increment. It should be noted that the results for other initial values of q_1 can be derived easily by simply applying a rotation on these results. Therefore, the previous maneuvers chosen are representative from any initial position (initial meaning $\mathbf{y_0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$). Let's assume the equations of the allocator are fixed and that only some parameters $\mathbf{P}_{alloc}$ are to be tuned. In Fig. 4 is the proposed iterative procedure to tune these parameters.

The performance criteria chosen to test the individuals are (see Fig. 5):

- The maximum deviation between the desired direction of movement and the output direction of the end-effector. It is computed by calculating the minimum

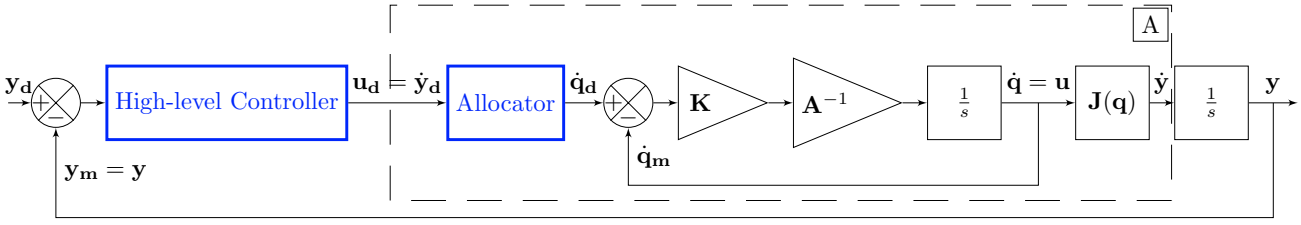

Fig. 3. Simplified Block diagram for our application.

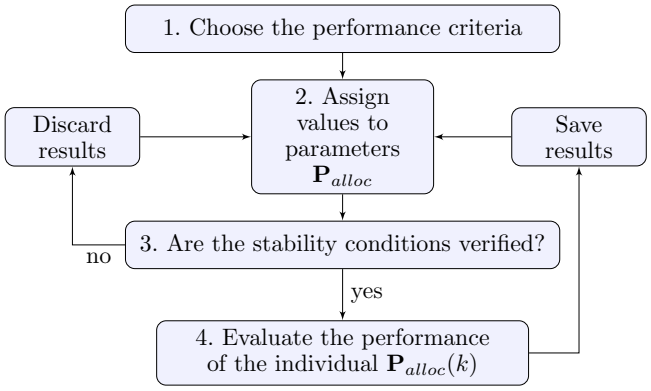

Fig. 4. Flow diagram of the iterative tuning procedure.

of the inverse of the cosinus of the deviations. It represents the errors created by the allocator.

- The mean time taken to reach the position saturation in the given directions. If during a simulation the saturation is not reached, then a high value is imposed. This can happen if the system cannot go through a singularity.

The verification of the stability conditions is carried out using the conditions described in Song and Grizzle (1995).

To implement this procedure, a genetic optimization algorithm has been used. It uses the MOGA II algorithm developped by Poloni and Pediroda (1997). It was helpful to generate the possible settings (values of $\mathbf{P}_{alloc}$). It uses the performance of the previous settings and creates new populations of possible settings through evolutions and mutations from the previous ones.

3.2 Application

The allocator chosen to be optimized is defined below, in a form very similar to the one described in Evain et al. (2016) and as a discrete-time controller. It uses the structure of the Extended Kalman Filter (EKF) to find the best $\Delta\mathbf{q} = \dot{\mathbf{q}} \times T_s$ (T_s is the discrete timestep) that verifies the kinematic equations and the constraints. The estimated variables $\hat{\mathbf{v}}$ of the EKF and the model equations are given in (7).

$$\begin{cases} \mathbf{v}_1 = \Delta\mathbf{q} \\ \mathbf{v}_2 = \Delta^2\mathbf{q} \\ \mathbf{v}_3 = \Delta\mathbf{y} \end{cases} \quad \text{and} \quad \begin{cases} \hat{\mathbf{v}}_1(t+T_s) = \frac{2}{3}\hat{\mathbf{v}}_1(t) \\ \hat{\mathbf{v}}_2(t+T_s) = \frac{4}{9}\hat{\mathbf{v}}_2(t) \\ \hat{\mathbf{v}}_3(t+T_s) = \frac{1}{2}\hat{\mathbf{v}}_3(t) + \frac{1}{2}\mathbf{w}(t) \end{cases} \tag{7}$$

with $\mathbf{w}$ the reference input imposed equal to $\mathbf{u_d}$. The measurement equations of the EKF read (8).

$$\begin{cases} \mathbf{m}_1 = \sum_{j=1}^{n} (\Delta q_j) \frac{\mathbf{H}'_j}{2} \Delta\mathbf{q} = -\mathbf{J}\mathbf{v}_1 - \frac{\mathbf{H}}{2}\mathbf{v}_2 + \mathbf{v}_3(t) \\ \mathbf{m}_2 = \mathbf{0}_{3\times1} = -\mathbf{v}_1^2 + \mathbf{v}_2 \\ \mathbf{m}_3 = \mathbf{k} = \mathbf{v}_1 \end{cases} \tag{8}$$

with $\mathbf{H}$ and $\mathbf{H}'$ the Hessian matrices of the system, defined in the paper by Evain et al. (2016). The first two equations of (8) are imposed by the constraints on the problem. $\mathbf{k}$ was chosen to help steering the manipulator away from singularities (9).

$$\mathbf{k} = \frac{1}{(1 + 1000 * det(\mathbf{J}\mathbf{J}^T))^5} \mathbf{I}_{3\times1} \tag{9}$$

In this application, only the covariance matrices of the model and of the measurement equations $\mathbf{R}$ and $\mathbf{Q}$ of the EKF are optimized (i.e. $\mathbf{P}_{alloc} = \{\text{diag}(\mathbf{R})\ \ \text{diag}(\mathbf{Q})\}$), but one can imagine adding the parameters of the third equation of (8) as tuning variables. This would be similar for instance to tuning a Singular Robust Inverse where the matrix $\mathbf{W}$ often depends on the determinant of $\mathbf{J}\mathbf{J}^T$ (Siciliano (1990)). For our application, Fig. 5 shows the results after analyses of 10 generations of 40 individuals, and the setting that minimizes the deviation is selected and analyzed more in-depth.

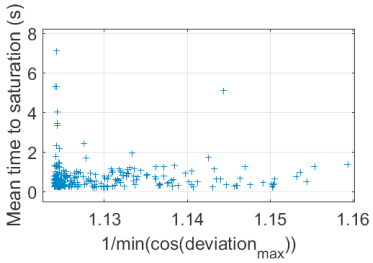

Fig. 5. Results of the genetic optimization

Once the setting has been chosen, the analysis of the system has to be carried out. The end-effector positions throughout the simulations in the required directions and the time to reach position saturation along these directions are presented in Fig. 6. One can check from this figure if the results are compatible with the performance required. In Fig. 6, the major instantaneous deviation occurs at the beginning of the end-effector movement. Another optimization should be carried out if the results are not satisfying. It should be noted that these analyses are in open-loop, so the performance should be improved with the high-level controller feedback.

An analysis on the impact of the input values can also be useful.

Here, it is assumed that the design point matches the requirements and is then chosen for the rest of the paper.

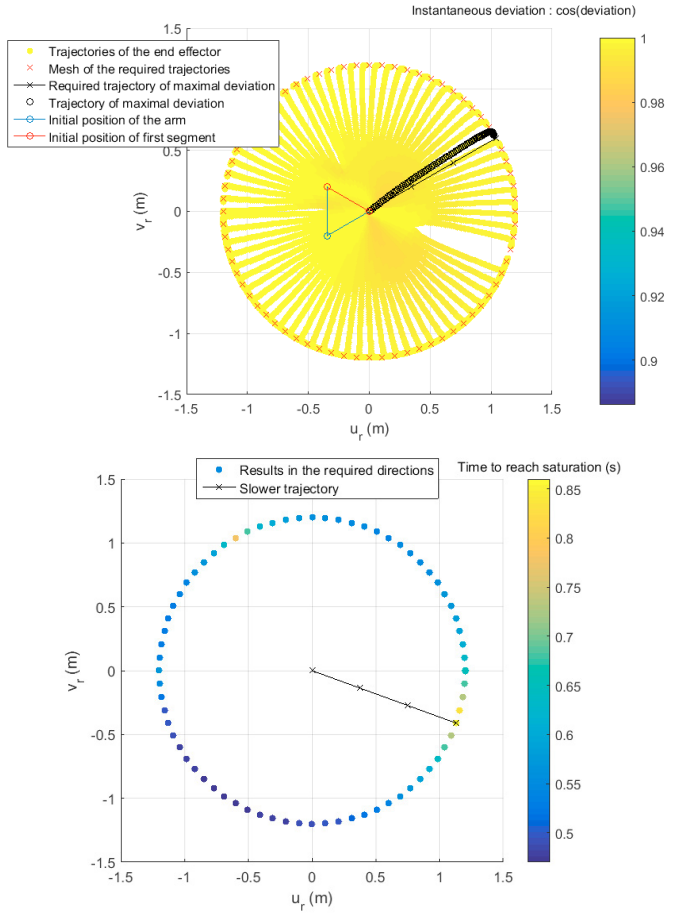

Fig. 6. Analyses of the deviations (up) and time to reach saturation (bottom)

4. IDENTIFICATION OF THE SYSTEM

The kinematic model (subsystem A in Fig. 3) will now be identified. The possible errors created by the nonlinear dynamic decoupling in the low-level controller are to be dealt with by the high-level controller if needed.

The two-input two-output model between $\dot{\mathbf{y}}_\mathbf{d}$ and $\dot{\mathbf{y}}$ is chosen according to the structure given in Fig. 7. An indirect identification method using a least-square parametric optimization has been carried out on the results of the open-loop simulation. The parameters to be estimated are t_1, t_2 and c, the coupling term whose only upper bound is computed.

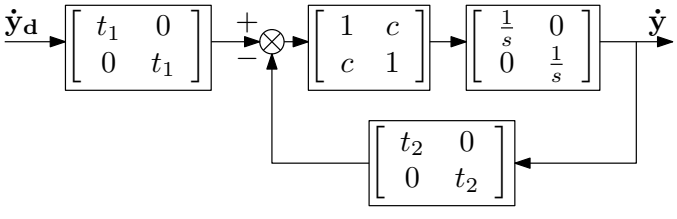

Fig. 7. Block diagram of the identification model

The transfer matrix of Fig. 7 reads :

$$\dot{\mathbf{y}}(s) = t_1 \frac{\begin{bmatrix} s+t_2(1-c^2) & cs \\ cs & s+t_2(1-c^2) \end{bmatrix}}{s^2+2t_2 s+t_2^2(1-c^2)} \dot{\mathbf{y}}_\mathbf{d} \qquad (10)$$

with s the Laplace variable. $c < 1$ for the system to be stable.

Nevertheless, to account for the varying performance along the directions, the variations of the parameters t_1 and t_2 are identified (see Fig. 8). In this figure, the identification is done for each direction.

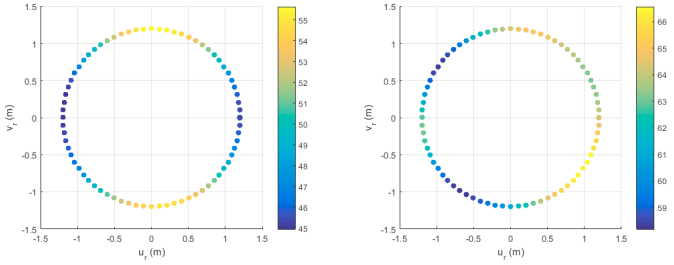

Fig. 8. Identification of the parameters t_1 (left) and t_2 (right) in each direction for an input norm of $\dot{y}_d = 1$ m/s

The nominal values of t_1 and t_2 are then chosen as their mean values, and the minimal ($\underline{t_1}$ and $\underline{t_2}$) and maximal values ($\bar{t_1}$ and $\bar{t_2}$) are also computed to synthetise the robust controller. Analyzing the impact of input variations helps ensuring the identification hypothesis (equation 10) is correct.

In addition, since deviations are also present near singularities, the cross-coupling disturbances c have to be studied. For the H_∞ synthesis, the maximal norm of these deviations is sufficient. In Fig. 9, the simulation with the maximal deviation is presented and the maximal value of c can be derived: $c_{max} = 0.66\ m/s$.

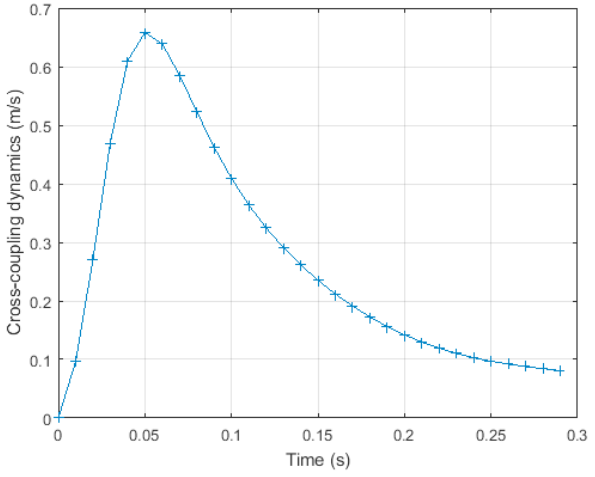

Fig. 9. Simulation with the worst deviation

5. ROBUST CONTROLLER DESIGN

Once the allocator is tuned and identified, the high-level controller can be derived. In order to take into account the uncertainties on the parameters of the identification and

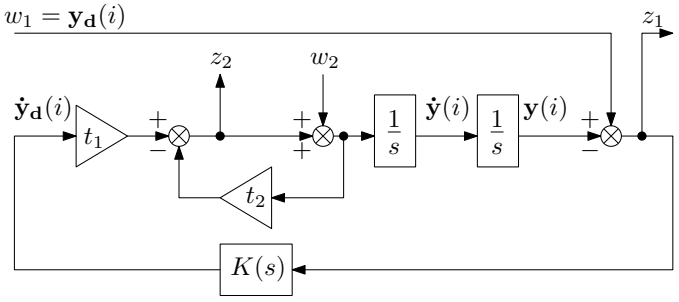

Fig. 10. H_∞ control problem on a single degree-of-freedom $i = 1, 2$.

remain robust in spite of the nonlinearities, a robust H_∞ synthesis has been chosen. Different solvers exist for the H_∞ problem, in particular the ones described by Arzelier et al. (2011) and Apkarian and Noll (2006). In our case, the synthesis was carried out with the MATLAB Robust Control Toolbox.

The theory of the H_∞ problem is not detailed here but can be found in the book of Alazard et al. (1999). The goal of the design is to find a single controller $K(s)$, applied to both degrees-of-freedom ($\mathbf{y}(i)$, $i = 1, 2$), that stabilizes the closed-loop system and ensures :

- performance, through a template $S_{des}(s)$ on the output sensitivity function, i.e.: the transfer from w_1 to z_1 on the block-diagram presented Fig. 10,
- unsensitivity to cross-coupling disturbances through an upper bound on the magnitude of the transfer from w_2 to z_2,
- unsensitivity to parameter variations on t_1 and t_2.

The problem to solve can be stated as : find a PI (Proportional Integral) controller $K(s)$ minimizing:

$$\| \frac{1}{S_{des}(s)} \frac{t_1\, K(s)}{s^2 + t_2\, s + t_1\, K(s)} \|_\infty$$

such that:

$$\| \frac{t_1\, K(s) + t_2\, s}{s^2 + t_2\, s + t_1\, K(s)} \|_\infty < 1/c_{max}$$

$$\forall \quad t_1 \in [\underline{t_1} \quad \bar{t_1}], \quad t_2 \in [\underline{t_2} \quad \bar{t_2}]$$

The template $S_{des}(s)$ is chosen as:

$$S_{des}(s) = \frac{s^2}{s^2 + 2\xi_d \omega_d s + \omega_d^2}$$

with $\xi = 0.7$ and $\omega_d = 3$ rad/s.

In our case, the synthesis gives the controller :

$$K(s) = (6.56 + \frac{13.74}{s})(\mathbf{y_d}(i) - \mathbf{y}(i)), i = 1, 2$$

The Bode diagrams enable us to check if the synthesis meets the constraints imposed (Fig. 11). In this figure, the uncertain system is represented by realizations of the transfer functions for different values of the parameters t_1 and t_2.

As can be noted, the closed-loop system meets the requirements imposed since the different transfers are all below the required weightings, even with some margins for the

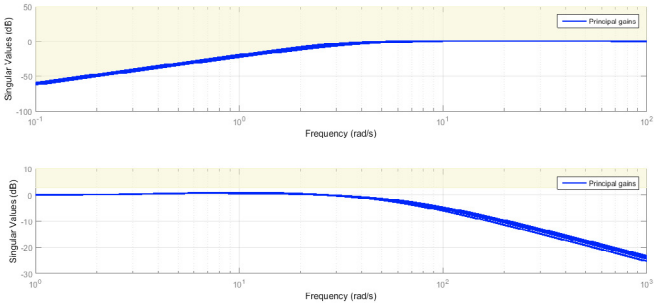

Fig. 11. Analysis of the constraints (up: sensitivity function requirement, down: cross-coupling requirement)

second requirement (cross-coupling robustness) that was the hard constraint.

Finally, a simulation showing the behaviour of the closed-loop system, with the true dynamics explained in section 2 and the controller synthetised, is presented in Fig. 12. The simulation is carried out along the direction of maximal deviation presented in Fig. 6, with a ramp input of slope 1m/s to be consistent with the previous open-loop simulations. It can be noticed that the deviation becomes very limited thanks to the design of the closed-loop control law.

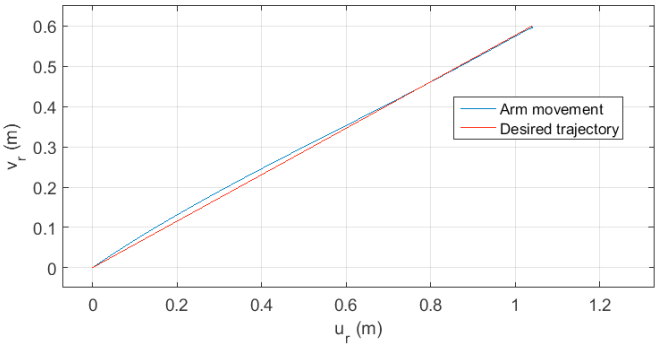

Fig. 12. Simulation with the closed-loop system

6. CONCLUSION

With the linearization method, if the modelling of the nonlinear effects around singularities is conservative, then the modulus margins can be chosen and derived from the H_∞ controller, ensuring robust stability. For systems with many singularities and errors created, this technique can be useful to avoid unstability. Each step of the procedure can be improved regardless of the others. A more adapted integration of constraints in the allocator could provide better optimization results. Depending on the results, robust stability can be impossible to achieve for a fixed-structure controller. Two main strategies stand out to avoid cases of unstabilities, whether work on the settings of the allocator, or change the structure of the controller so that more degrees of freedom can be imparted. To test the control loop in a real environment, an hardware experiment is planned with a Control Moment Gyro cluster (see Evain et al. (2017) for the description of the experimental setup). The results of the procedure will be tested on this system in a parabolic flight campaign carried out by Novespace and in the frame of the 2017

Fly Your Thesis! programme by ESA (European Space Agency) Education Office with the Flying Squirrels team. For these experiments, delays, other uncertainties and unmodelled behaviors will have to be taken into account and particularly in the H_∞ controller. To make the simulations more representative, the work will now focus on improving the modelling of the system and the sensors, hence adding noise and other parameters uncertainties. Comparisons of the results with other controllers as model predictive control will be carried out.

REFERENCES

Alazard, D., Cumer, C., Apkarian, P., Gauvrit, M., and Ferreres, G. (1999). *Robustesse et Commande Optimale*. Cépaduès Editions, Toulouse.

Apkarian, P. and Noll, D. (2006). Nonsmooth h-infinity synthesis. *IEEE Transactions on Automatic Control*, 51(1), 71–86.

Arzelier, D., Deaconu, G., Gumussoy, S., and Henrion, D. (2011). H2 for hifoo. *International Conference on Control and Optimization with Industrial Applications, Bilkent University, Ankara, Turkey*.

Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 722–728.

Buffington, J., Enns, D., and Teel, A. (1998). Control allocation and zero dynamics. *Journal of Guidance, Control and Dynamics*, 21(3), 458–464.

Evain, H., Rognant, M., Alazard, D., and Mignot, J. (2016). Nonlinear dynamic inversion for redundant systems using the ekf formalism. *American Control Conference (ACC)*, 348–353.

Evain, H., Solatges, T., Brunet, A., Dias Ribeiro, A., Sipile, L., Rognant, M., Alazard, D., and Mignot, J. (2017). Design and control of a nano-control moment gyro cluster for experiments in a parabolic flight campaign. *IFAC World Congress*.

Fossen, T. and Johansen, T. (2006). A survey of control allocation methods for ships and underwater vehicles. In *14th Mediterranean Conference on Control and Automation*, 1–6.

Härkegård, O. and Glad, T. (2005). Resolving actuator redundancy optimal control vs. control allocation. *Automatica*, 41, 137–144.

Hu, Q., Li, B., and Zhang, Y. (2014). Nonlinear proportional-derivative control incorporating closed-loop control allocation for spacecraft. *Journal of Guidance, Control and Dynamics*, 27(3), 799–812.

Hu, T. and Lin, Z. (2001). *Control Systems with Actuator Saturation : Analysis and Design*. Birkhäuser, Boston.

Johansen, T. (2004). Optimizing nonlinear control allocation. *IEEE Conference on Decision and Control*, 4, 403–410.

Johansen, T. and Fossen, T. (2013). Control allocation - - a survey. *Automatica*, 49, 1087–1103.

Khalil, W. and Dombre, E. (2004). *Modeling, identification and control of robots*. Butterworth-Heinemann, Oxford.

Lewis, F. and Syrmos, V. (1995). *Optimal Control*. Wiley, New York.

Morabito, F., Teel, A.R., and Zaccarian, L. (2003). High performance anti-windup for robot manipulators. *European Control Conference (ECC)*, 489–494.

Nakamura, Y. and Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *Transactions of ASME Journal of Dynamic Systems, Measurement and Control*, 108, 163–171.

Page, A. and Steinberg, M. (1999). Effects of control allocation algorithms on a nonlinear adaptive design. *Guidance, Navigation, and Control Conference and Exhibit. Portland, OR, U.S.A.*, 1664–1674.

Poloni, C. and Pediroda, V. (1997). Ga coupled with computationally expensive simulations: tools to improve efficiency. *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, 267–288.

Siciliano, B. (1990). Kinematic control of redundant robot manipulators. *Journal of Intelligent and Robotic Systems*, 3, 201–212.

Siciliano, B. and Sciavicco, L. (2000). *Modeling and Control of Robot Manipulators*. Springer-Verlag, London.

Simon, D. and Simon, D. (2006). Kalman filtering with inequality constraints for turbofan engine health estimation. *IEE Proceedings - Control Theory and Applications*, 153(3), 371–378.

Song, Y. and Grizzle, J. (1995). The extended kalman filter as a local asymptotic observer for discrete-time nonlinear systems. *Journal of Mathematical Systems, Estimation, and Control*, 5(1), 59–78.

Sun, N., Fang, Y., Chen, H., and Lu, B. (2017). Amplitude-saturated nonlinear output feedback antiswing control for underactuated cranes with double-pendulum cargo dynamics. *IEEE Transactions on Industrial Electronics*, 64(3), 2135–2146.

Wampler, C. (1986). Manipulator inverse kinematic solutions based on damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics*, 16, 93–101.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotic Research*, 4(2), 3–9.