Preprint version 12th IFAC Symposium on Robot Control, Budapest, Hungary (2018)

Design and Input Allocation for Robots with Saturated Inputs via Genetic Algorithms

Michele Furci * **Davide Bicego** * **Antonio Franchi** *

* *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France (e-mail: michele.furci@laas.fr, davide.bicego@laas.fr, antonio.franchi@laas.fr).*

Abstract: In this paper we consider fully-actuated and redundantly-actuated robots, whose saturated inputs can have high bandwidth or can be slowly varying (with dynamics). The slowly varying inputs can be considered as *configurations* for the system. The proposed strategy allows to find the optimal actuators' *configuration* to optimize a cost function as the manipulability or the energy consumption. The approach allows for both a static design, which can include actuators' parameters such as position, orientation, saturations, numbers of actuators, and for a dynamic design, where the *configurations* can be controlled by an input of the system. A generalized solution to the optimal problem is proposed with the use of genetic algorithms. The results are validated in two simulation scenarios: a reconfiguration of the actuators orientation of an redundantly-actuated planar robot for trajectory tracking and the design optimization of the orientation of the motors in a generalized hexa-rotor with arbitrary propeller orientation.

Keywords: input allocation, fully actuated/vectored-thrust aerial vehicles, genetic algorithms.

1. INTRODUCTION

In control theory, an redundantly-actuated system consists of a system with more inputs than the dimension of the configuration space or the D.o.F. A common hierarchical control architecture (Johansen and Fossen (2013)) for such systems consists in having a high-level control that generates a virtual input which stabilizes the system. The control allocation strategy then commands the multiple actuators to obtain the desired virtual input, trying to exploit the higher dimension space of the inputs to optimize some cost functions (for energy, saturations, fault tolerance, etc). Very often in physical-mechanical system we might have actuators with very high bandwidth which can be considered as almost instantaneous, and others with low bandwidth evolving with dynamics (Passenbrunner et al. (2016)). In this paper we consider such systems, where some slow varying parameters (we call *configurations*) have dynamics, while some high bandwidth inputs (we call *instantaneous inputs*) can be obtained instantaneously. Moreover, the inputs are subject to saturation constraints. We focus on the optimization of the *configurations*, while the *instantaneous inputs* are found using standard techniques as the constrained pseudo-inverse. In particular, we address two problems: the *control* problem where the *configurations* can actually be controlled and modified by some inputs, and the *design* problem where the configurations are static and the goal is to find an optimal static configuration at design time to optimize a cost function. Possible applications include, but are not limited to: ships and vessels (Lindegaard and Fossen (2003)), aircrafts (Oppenheimer et al. (2006), Boskovic and Mehra (2002)), wheeled mobile robots and multi-rotors unmanned aerial vehicles (Ryll et al. (2012, 2016), Brescianini and D'Andrea (2016), Park et al. (2016)).

Since the resulting problem might be in high dimension, non-linear, with non-differentiable and non-closed form cost functions, we propose a generalized solution with the use of genetic algorithms (Srinivas and Patnaik (1994)). We propose different cost functions associated to different tasks for the robot, in particular we describe cost functions to optimize: the energy consumption of the *instantaneous inputs*, the volume of attainable virtual inputs and the manipulability index, that is the biggest ball of obtainable virtual input around a desired virtual input.
The contribution of this paper is on the use of Genetic Algorithm to solve complex non-analytic problem. In Passenbrunner et al. (2016), for example, a gradient based method is employed which is impossible to apply in our case. In Lindegaard and Fossen (2003), due to the analytic solution, the approach has limitations on the number of saturated actuators. Many approaches try to use Linear or Quadratic Programming techniques, but those require very specific cost functions. With the proposed approach we are able to optimize very complex cost functions as the *manipulability index* which to the best of our knowledge was never optimized for an redundantly-actuated system.
To conclude, we simulated the proposed approach for both a *control* and a *design* problem; the first on a planar robot with actuators consisting in directable thrusters and the second on a generalized Multi Directional Thrust (MDT) hexa-rotor with arbitrary propeller direction.

1.1 Organization

The paper is organized as follows: Section 2 briefly presents the notation used in the paper. In section 3 the problem is introduced and some relevant cost functions for different tasks are outlined. In section 4 a generalized solution with the use of Genetic Algorithms is proposed and finally section 5 presents some results on the simulation for both the *control* and *design* problems.

2. NOTATION

A set $S \subset \mathbb{R}^n$ is called polyhedron if it is the intersection of a finite set of closed halfspaces:

$$S = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{C}\mathbf{x} \leq \mathbf{b}\} \tag{1}$$

with $\mathbf{C} \in \mathbb{R}^{q \times n}$, $\mathbf{b} \in \mathbb{R}^q$, with q the number of constraints defining the hyper-planes. This is the so called H-representation of polyhedron.

A bounded polyhedron is called a polytope.

The operator $\mathbf{S}^{\times}(a)$ indicates the skew symmetric matrix such that $\mathbf{S}^{\times}(a)b = a \times b$ with $\times$ the cross product operator and a, b two vectors of same dimension.

3. PROBLEM FORMULATION

Consider a dynamical system:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{w}) \tag{2}$$

with $\mathbf{x} \in \mathbb{R}^m$ is the state of the system and where $\mathbf{w} \in \mathbb{R}^n$ are some virtual inputs. The virtual inputs depends linearly from the *instantaneous inputs* via an allocation matrix, and in particular:

$$\mathbf{w}(t) = \mathbf{A}(\boldsymbol{\theta}(t))\mathbf{u}_p(t) \tag{3a}$$

$$\dot{\boldsymbol{\theta}}(t) = f_\theta(\boldsymbol{\theta}, \mathbf{u}_\theta(t)) \tag{3b}$$

with $\mathbf{A} \in \mathbb{R}^{n \times p}$ the allocation matrix, $\mathbf{u}_p \in \mathbb{R}^p$ a saturated vector of *instantaneous inputs*, $p \geq n$, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ some time-variant parameters (the *configurations*), $\mathbf{u}_\theta \in \mathbb{R}^{p_\theta}$ a vector of saturated inputs for the parameters $\boldsymbol{\theta}$, $f_\theta(\cdot) : \mathbb{R}^{p_\theta} \to \mathbb{R}^{n_\theta}$ a locally Lipschitz function. The input $\mathbf{u}_p$ is such that $\underline{s}_i \leq \mathbf{u}_{p,i} \leq \overline{s}_i$ for $i = 1, \cdots, p$, with $\overline{s}_i, \underline{s}_i \in \mathbb{R}$ the saturations limits. The input $\mathbf{u}_\theta$ is such that $\underline{s}_{i,\theta} \leq \mathbf{u}_\theta \leq \overline{s}_{i,\theta}$ for $i = 1, \cdots, p_\theta$, with $\overline{s}_{i,\theta}, \underline{s}_{i,\theta} \in \mathbb{R}$ the saturations limits.

In this paper we suppose that a reference virtual command $\mathbf{w}^*(t)$ is generated by a high-level controller and we focus on the optimal design or control of the allocation matrix $\mathbf{A}(\boldsymbol{\theta})$ to optimize some cost function $V(\boldsymbol{\theta})$. Given an optimal configuration $\mathbf{A}^*(\boldsymbol{\theta})$ and a feasible virtual input $\mathbf{w}^*$, the instantaneous input $\mathbf{u}_p$ can be found as:

$$\begin{cases} \mathbf{u}_p(t) = \mathbf{A}^{*-1}\mathbf{w}^*(t) & p = n \\ \mathbf{u}_p(t) = \mathbf{A}^{*\dagger}\mathbf{w}^*(t) & p > n \end{cases} \tag{4}$$

where $\mathbf{A}^\dagger$ is the constrained pseudoinverse operator Sabharwal and Potter (1998) to guarantee that the solution lies in the feasible saturated $\mathbf{u}_p$.

We define two problems:

Design Problem: we suppose that $f_\theta(\boldsymbol{\theta}, \mathbf{u}_\theta) = 0$ and $p_\theta = 0$, i.e., the configurations $\boldsymbol{\theta}$ are constant and there are no inputs to steer $\boldsymbol{\theta}$. The goal is to find a static optimal configuration $\boldsymbol{\theta}^*$ that maximizes/minimizes a cost function $V(\boldsymbol{\theta}) \in \mathbb{R}$. The design problem is useful to design some intrinsic properties of the vehicle such as orientation and position of actuators, maximum power of the actuators, etc.. The optimization problem is defined as:

$$\begin{aligned} \max_{\boldsymbol{\theta}}/\min \quad & V(\boldsymbol{\theta}) \\ \text{subject to} \quad & (3) \\ & \underline{s}_i \leq \mathbf{u}_{p,i} \leq \overline{s}_i \end{aligned} \tag{5}$$

Control Problem: the goal is to find an optimal configuration and configuration input trajectory $\{\boldsymbol{\theta}^*(t), \mathbf{u}_\theta(t)^*\}$ that maximize/minimize a cost function $V(\boldsymbol{\theta}) \in \mathbb{R}$. The optimization problem is defined as:

$$\begin{aligned} \max_{\boldsymbol{\theta}(t), \mathbf{u}_\theta(t)}/\min \quad & \int_0^{t_f} V(\boldsymbol{\theta}(t))dt \\ \text{subject to} \quad & (3) \\ & \underline{s}_i \leq \mathbf{u}_{p,i} \leq \overline{s}_i \\ & \underline{s}_{i,\theta} \leq \mathbf{u}_\theta \leq \overline{s}_{i,\theta} \end{aligned} \tag{6}$$

To describe the problem we have to define the sets $\mathbb{W} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^p$ as the virtual input set and the instantaneous input set respectively, i.e., where the virtual input $\mathbf{w}$ and the instantaneous input $\mathbf{u}_p$ vectors can lie. Due to the nature of the saturated input $\mathbf{u}_p$, $\mathbb{U}$ is a p-orthotope (hyper-rectangle of dimension p) polytope defined in H-representation as:

$$\mathbb{U} = \{\mathbf{x} \in \mathbb{R}^p | \mathbf{C}_\mathbb{U}\mathbf{x} \leq \mathbf{b}_\mathbb{U}\} \tag{7}$$

with $\mathbf{C}_\mathbb{U} \in \mathbb{R}^{2p \times p}$ defined as:

$$\mathbf{C}_\mathbb{U} = \begin{bmatrix} \mathbf{I}_p \\ -\mathbf{I}_p \end{bmatrix} \tag{8}$$

with $\mathbf{I}_p$ an identity matrix of dimension p, and $\mathbf{b}_\mathbb{U}$ defined as:

$$\mathbf{b}_\mathbb{U} = \begin{bmatrix} \overline{s}_1 & \cdots & \overline{s}_p & -\underline{s}_1 & \cdots & -\underline{s}_p \end{bmatrix}^T \tag{9}$$

Due to (3a) $\mathbb{W}$ is the image of $\mathbb{U}$ subject to linear transformation $\mathbf{A}(\boldsymbol{\theta})$, which is again a polytope Zhang (2012) defined in H-representation as:

$$\mathbb{W} = \{\mathbf{x} \in \mathbb{W}^p | \mathbf{C}_\mathbb{W}\mathbf{x} \leq \mathbf{b}_\mathbb{W}\}. \tag{10}$$

3.1 Cost Function Definition

In the following we define suitable cost functions V to be optimized, based on different tasks for the robot.

Manipulability task: a desired virtual input trajectory is given by $\mathbf{w}^*(t)$. The goal is to find the optimal configuration $\boldsymbol{\theta}^*(t)$ and the configuration input $\mathbf{u}_\theta$ such that the manipulability index around $\mathbf{w}^*(t)$ is the maximum, considering as manipulability index the radius of the largest sphere inscibed into the polytope $\mathbb{W}$ and centered in $\mathbf{w}^*(t)$. This goal can be useful to provide robustness to the high level control. The higher the manipulability index, the higher the high level control has margin around the nominal control input $\mathbf{w}^*(t)$ to compensate external disturbances or disturbances due to non perfect modeling.

If we define as d_i the distance of $\mathbf{w}^*(t)$ to the i-th hyperplane that constraints $\mathbb{W}$, we define the cost function as:

$$V_M(\boldsymbol{\theta}) = \min(d_i(\boldsymbol{\theta})),\ i = 1, \ldots, t,\ t \leq 2p \tag{11}$$

In the evaluation of the cost functions using $\mathbb{W}$, we might have different measurements units (for example [N] and [N/m] for forces and torques, respectively) or preferred directions in $\mathbb{W}$ (for example we want higher manipulability in the forces of x-axis). In this case we can simply scale the matrix $\mathbf{A}$ by pre-multiplying it with a scale matrix $\mathbf{A}_\lambda \in \mathbb{R}^{n \times n}$ defined as:

$$\mathbf{A}_\lambda = \begin{bmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_n \end{bmatrix} \tag{12}$$

Remark 1. Optimizing for the manipulability task, it is not guaranteed that the desired $\mathbf{w}^*$, around which we maximize the radius of the sphere, is feasible w.r.t. the constraints on $\mathbf{u}_p$.

Energy task: a desired virtual input trajectory is given by $\mathbf{w}^*(t)$. From (4) we can compute $\mathbf{u}_p(t)$. The cost function has to be chosen to minimize the norm of the *instantaneous* *inputs*. The cost function to maximize in case of $(p = n)$ is defined as:

$$\begin{cases} V_E(\boldsymbol{\theta}) = \left\|\mathbf{A}^{-1}(\boldsymbol{\theta})\mathbf{w}^*(t)\right\|^{-1} & \mathbf{w}^*(t) \in \mathbb{W} \\ V_E(\boldsymbol{\theta}) = \left\|\mathbf{A}^{-1}(\boldsymbol{\theta})\overline{\mathbf{w}}^*(t)\right\|^{-1} & \mathbf{w}^*(t) \notin \mathbb{W} \end{cases} \quad (13)$$

while in case of $(p > n)$ it is defined as:

$$\begin{cases} V_E(\boldsymbol{\theta}) = \left\|\mathbf{A}^{\dagger}(\boldsymbol{\theta})\mathbf{w}^*(t)\right\|^{-1} & \mathbf{w}^*(t) \in \mathbb{W} \\ V_E(\theta) = \left\|\mathbf{A}^{\dagger}(\boldsymbol{\theta})\overline{\mathbf{w}}^*(t)\right\|^{-1} & \mathbf{w}^*(t) \notin \mathbb{W} \end{cases} \quad (14)$$

where $\overline{\mathbf{w}}^*(t)$ is the obtainable virtual input, closest to $w^*(t)$ (can be found with quadratic programming Wolfe (1976)).

Remark 2. Optimizing for the energy task, it is not guaranteed that the desired $\mathbf{w}^*$ is feasible using the configuration obtained.

Wrench maximum volume task: the goal is to find the optimal $\boldsymbol{\theta}^*$ to maximize the volume of $\mathbb{W}$. The cost function to maximize is defined as:

$$V_V(\theta) = vol(\mathbb{W}) \quad (15)$$

where $vol(\mathbb{W})$ is the volume of $\mathbb{W}$.

The cost function can be defined as weighted sum of different tasks, i.e., we can define V as $V = \beta_1 V_M + \beta_2 V_E + \beta_3 V_V$, with β_i some positive scalar.

4. OPTIMIZATION PROBLEM SOLUTION

In the following, we present a solution to the optimal problems (5) and (6) for the mentioned tasks, and we also propose a generalized solution to consider other possible tasks or a cost function mixing the different tasks.

4.1 Manipulability

The solution lies into describing $\mathbb{W}$ as function of $\mathbb{U}$ and $\mathbf{A}(\boldsymbol{\theta})$. The solution is different in case of *fully actuation* $(p = n)$ or in case of *redundant actuation* $(p > n)$:

- *Fully Actuation* $(p = n)$: if $\mathbf{A}(\boldsymbol{\theta})$ is invertible, we can compute $\mathbf{b}_{\mathbb{W}} = \mathbf{b}_{\mathbb{U}}$ and $\mathbf{C}_{\mathbb{W}} = \mathbf{C}_{\mathbb{U}}\mathbf{A}(\boldsymbol{\theta})^{-1}$ (closed form solution). If $\mathbf{A}(\boldsymbol{\theta})$ is non-invertible, the cost function $V_M = 0$.
- *Redundant Actuation* $(p > n)$: since $\mathbf{A}$ is a projection on a lower dimension space, the problem is known in literature for not having a closed form solution. A numerical solution consists in expressing $\mathbb{U}$ in its V-representation (the vertex representation), transform the verteces of $\mathbb{U}$ via $\mathbf{A}(\boldsymbol{\theta})$, compute the V-representation of $\mathbb{W}$ as convex hull of the transformed points using Chazelle (1993). The problem is known as the *facet enumeration problem* Fukuda (2004).

Once $\mathbb{W}$ is defined, we can compute d_i as

$$d_i = \frac{|\mathbf{C}^i_{\mathbb{W}}\mathbf{w}^* - \mathbf{b}^i_{\mathbb{W}}|}{\left\|\mathbf{C}^i_{\mathbb{W}}\right\|} \quad (16)$$

where the i index refers to the i-th row or the i-th hyperplane defining $\mathbb{W}$.

4.2 Energy

- *Fully Actuation* $(p = n)$: the cost function requires the computation of $\mathbb{W}$ which can be done in closed form and the computation of $\mathbf{A}^{-1}$ which again can be done in closed form from $\mathbf{A}(\boldsymbol{\theta})$. Some gradient-based algorithms can be used to compute the optimal $\boldsymbol{\theta}^*$.
- *Redundant Actuation* $(p > n)$: both the computation of $\mathbb{W}$ and $\mathbf{A}^{\dagger}$ cannot be done in closed form, so the resulting cost function is non-differentiable and non-continuous.

4.3 Wrench Maximum Volume

The cost function is directly dependent on the volume of $\mathbb{W}$ which can be computed in a numerical fashion by one of the algorithms in Bueler et al. (2000).

4.4 Other Tasks

The solution of the generalized problem, as described in the following, requires only to be able to evaluate the cost function for a particular configuration $\boldsymbol{\theta}$. Hence complex cost function can be built to be optimized.

4.5 Optimization Problem Solution

In general, the cost function can be non-linear, non-differentiable (for the computation of $\mathbb{W}$ and/or $\mathbf{A}^{\dagger}$) and in high dimensions, so standard optimization algorithms based on differentiation Ruder (2016) or complete solutions cannot be applied. A general solution can instead be derived using genetic algorithms (GA) Srinivas and Patnaik (1994). In case of the *Design* problem a single GA can be executed to find the optimal configuration $\boldsymbol{\theta}^*$. In the *Control* problem, instead, we discretize $\mathbf{w}^*(t)$ with a given time step Δt, and solve a GA for each step. Moreover, we need to consider the dynamic equation (3b) in the solutions of the GA. We have then to introduce the concept of *Initial Genetic Algorithm (IGA)* and *on Trajectory Genetic Algorithm (TGA)*. The IGA is used in the *Design* problem and to find the initial configuration $\boldsymbol{\theta}^*(0)$ for the *Control* problem. The TGA is instead used along the trajectory to find the $\boldsymbol{\theta}^*(t > 0)$. Since the TGA has to obey (3b) we need to find a suitable mechanism to find feasible solutions of $\boldsymbol{\theta}^*(t)$. For sake of compactness we call TGA$(k\Delta t)$ as TGA$_k$ and we use the subscript k to indicate a $k\Delta t$. The choice of the initial population, selection and mutation of each TGA$_k$ should guarantee that $\boldsymbol{\theta}^*_k = \boldsymbol{\theta}^*_{k-1} + f_\theta(\mathbf{u}_{\boldsymbol{\theta},k})$. Suitable strategies to pick a feasible $\boldsymbol{\theta}^*$ can be derived from kino-dynamic planning LaValle and J. J. Kuffner (2001) and are model specific and GA specific.

5. SIMULATIONS

In the following, we present the results of two case studies, one for the *control* problem and one for the *design* problem.

5.1 Control Case: redundantly-actuated planar robot

The platform described in Nainer et al. (2017) and depicted in Figure 1, is composed by a rigid rectangular frame sustained by omnidirectional passive spherical wheels and n actuating modules. Each module consists of a turret, which is orientable by means of a servo motor. Each turret carries a propeller driven by a BLDC motor. We consider the platform to have 4 actuator modules.

We can define equation (3) for this particular robot by defining $\mathbf{w}, \boldsymbol{\theta}, \mathbf{A}, \mathbf{u}_p, f_\theta, \mathbf{u}_\theta$. In particular $\mathbf{w} = [F_x, F_y, \tau]^T$ is the wrench vector, where F_x is the body force along x-axis, F_y is the body force along y-axis and τ the torque around the vertical axis; $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_4]$ with θ_i the angle of the i-th turret; $\mathbf{u}_p = [k_\omega \omega_1^2, k_\omega \omega_2^2, k_\omega \omega_3^2, k_\omega \omega_4^2]^T$, where k_ω is an aerodynamic constant and ω_i the angular velocity of the i-th propeller. $f(\theta, u_\theta) = \mathbf{u}_\theta$, $\mathbf{u}_\theta = [u_{\theta,1}, u_{\theta,2}, u_{\theta,3}, u_{\theta,4}]^T$ where $u_{\theta,i}$ is the control input for the turret's angle. The turret model is a single integrator system. Finally the allocation matrix is defined as:

$$\mathbf{A}(\theta) = \begin{bmatrix} c(\theta_1) & \cdots & c(\theta_4) \\ s(\theta_1) & \cdots & s(\theta_4) \\ (\mathbf{\Pi r}_1)^T \begin{bmatrix} c(\theta_1) \\ s(\theta_1) \end{bmatrix} & \cdots & (\mathbf{\Pi r}_4)^T \begin{bmatrix} c(\theta_4) \\ s(\theta_4) \end{bmatrix} \end{bmatrix} \quad (17)$$

where $\mathbf{\Pi} = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, $\mathbf{r}_i \in \mathbb{R}^2$ the position of the turret w.r.t. the center of gravity of the platform and $c(\cdot)$, $s(\cdot)$ indicating the cos and sin function respectively.

A desired wrench trajectory $\mathbf{w}^*(t)$, derived from the model inversion to track a 8-shaped trajectory, is given by:

$$\mathbf{w}^*(t) = \begin{bmatrix} F_x^* \\ F_y^* \\ \tau^* \end{bmatrix} = \begin{bmatrix} -M f_1^2 R \cos(f_1 t) \\ -M f_2^2 R \sin(f_2 t) \\ 0 \end{bmatrix} \quad (18)$$

with R, f_1, f_2 some parameters to define the size and the frequency of the trajectory and M the mass of the robot.

The following results are for the *manipulability* and *energy* tasks, whose cost functions are given in (11) and (14). Because we are in the redundantly-actuated problem, the Genetic Algorithm solution is employed.

The population is defined as N_p individuals p_i defined by a series of chromosomes. In particular the individuals p_i are defined as the sets $\{\theta_1, \theta_2, \theta_3, \theta_4\}$, i.e., the chromosomes are the parameters θ to be optimized. The fitness function is defined by the cost function (11) or (14), depending on the task. The steps of the genetic algorithm are defined as the following:

- **Initial Population (IGA):** the initial population is selected randomizing $\theta_i \in [-\pi, \pi]$.
- **Initial Population (TGA):** the initial population is selected randomizing $\theta_i = \theta_i^o \pm \delta_{\theta,i}$, where θ_i^o is the optimal value of θ_i from the previous iteration and $\delta_{\theta,i}$ is the maximum variation of θ_i in Δt given a limitation of the $u_{\theta,i}$. Since (3b) is a single integrator model $\delta_{\theta,i} = \overline{u}_{\theta,i}$.
- **Selection:** the selection mechanism used is the *fitness proportionate selection* Blickle and Thiele (1996).
- **Crossover:** the crossover mechanism consists in taking 2 chromosomes (two θ_i) from each parent.
- **Mutation (IGA):** in the mutation process at the initial iteration, each chromosome has a probability P_M of being randomized again in $[-\pi, \pi]$.
- **Mutation (TGA):** in the mutation process along the trajectory, each chromosome has a probability P_M of being increased by $\pm\delta_{\theta,i}$.

The parameters used in the simulations are gathered in Table 1, where we indicate with N_G the number of generations for each GA solution.

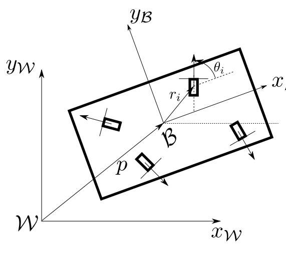


Fig. 1. The ROSPO platform.

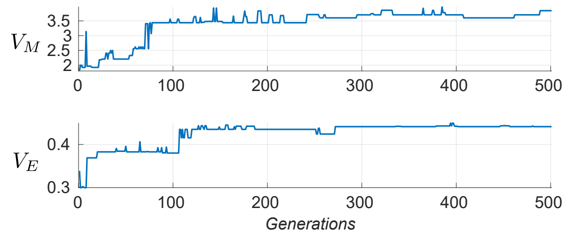
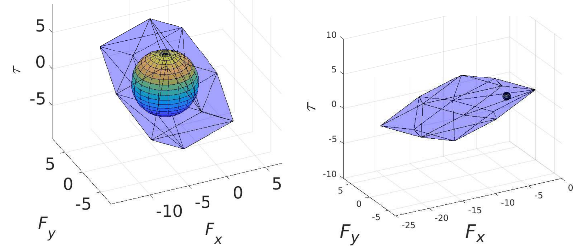


Fig. 2. Fitness functions for the IGA problem in both *manipulability* and *energy* tasks.



(a) Manipulability task. (b) Energy task.

Fig. 3. The obtainable wrench polytope after the solution of IGA ($t = 0$). The inscribed ball is the so called *manipulability index*, which measure the possibility of generating virtual inputs around a reference virtual input for robustness purpose.

In Figure 2 the fitness value for the IGA is depicted. The fitness converges to a steady value in around 500 generations.

In Figure 3 we can see the obtainable virtual input $\mathbf{w}$ polytope (in light blue) with the optimal configuration solution given by the IGA ($t = 0$), for both the manipulability and the energy tasks. Inside the polytope is depicted the sphere indicating the manipulability index centered around the initial desired wrench. We can notice that in the configuration optimized for the manipulability we can obtain a much wider range of wrenches in all direction, while in the configuration optimized for the energy we can obtain a large range in x-axis (the direction of $\mathbf{w}^*(t = 0)$) and very small in the other directions.

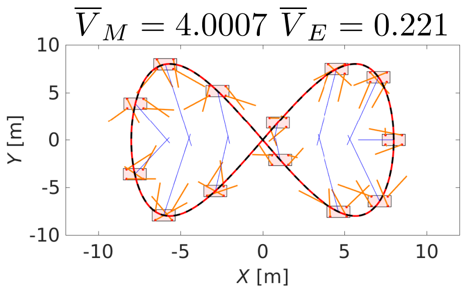
In Figure 4 the system is tested with the high level controller developed in Nainer et al. (2017). We can see the different behavior of the turrets. In the *energy task* the turrets are pointing mainly in the direction of the requested wrench (the blue line), but the manipulability index is very low (the mean $\overline{V}_M = 0.41$), such that the platform cannot compensate big external disturbances. In the *manipulability task* the turrets are more distributed, the manipulability index much higher (the mean $\overline{V}_M = 4$), but the energy consumption is almost double (mean $\overline{V}_E = 0.221$ against $\overline{V}_E = 0.39$).

5.2 Design Case: Fully Actuated Hexa-rotor

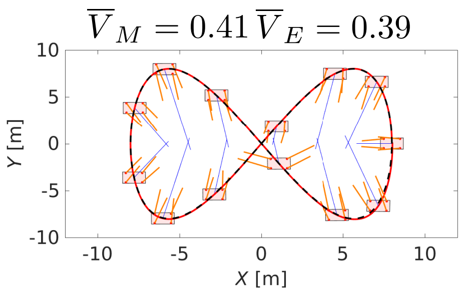
For the design case we consider a hexa-rotor unmanned aerial vehicle (UAV). Typical hexa-rotors have 6 equidistant motor-propeller actuators which point in the vertical z-axis. Some particular hexa-rotors Ryll et al. (2016)

Table 1. Parameters for GCM weight functions.

R	f_1	f_2	Δt	N_p	N_G	P_M	$\overline{s}_i$	$\underline{s}_i$
8	0.3	0.3	0.1(s)	90	500	0.05	5.8 (N)	0(N)



(a) Manipulability task.



(b) Energy task.

Fig. 4. The trajectory of the platform with the high level controller with the solution of the TGA in manipulability and energy task. The direction of the turrets along the trajectory is indicated by orange arrows starting from the platform.

instead have tilted actuators to generate forces in 3 axes and torques in 3 axes. Inspired by those designs, the goal is to define the optimal orientation of each independent propeller to optimize some cost functions. We define $\mathbf{r}_i \in \mathbb{R}^3$ the position vector of the i-th actuator w.r.t. the center of gravity of the vehicle. We define $\mathbf{v}_i \in \mathbb{S}^2$ the orientation of the i-th actuator in body frame, where the group $\mathbb{S}^n$ defines the unit sphere group $\mathbb{S}^n = \left\{x \in \mathbb{R}^{n+1} | \, \|x\| = 1\right\}$. The vector $\mathbf{v}_i$ describes where the propeller is pointing. We define $\sigma_i \in \{-1, +1\}$ the spinning direction of the i-th propeller. We can define equation (3) for this particular robot by defining $\mathbf{w}, \boldsymbol{\theta}, \mathbf{A}, \mathbf{u}_p, f_\theta, \mathbf{u}_\theta$. In particular $\mathbf{w} = [\mathbf{F}, \boldsymbol{\tau}]^T$ is the wrench vector, where $\mathbf{F} \in \mathbb{R}^3$ is the body force in three dimensions x, y, z and $\boldsymbol{\tau} \in \mathbb{R}^3$ the torque in three dimensions. $\boldsymbol{\theta} = [\mathbf{v}_1, \cdots, \mathbf{v}_6]$. $\mathbf{u}_p = [k_\omega \omega_1^2, \cdots, k_\omega \omega_6^2]^T$, where k_ω is an aerodynamic constant and ω_i the angular velocity of the i-th propeller. We consider the design problem, so $\dot{\boldsymbol{\theta}} = 0$. Finally the allocation matrix is defined as:

$$\mathbf{A}(\boldsymbol{\theta}) = \begin{bmatrix} & \boldsymbol{\theta} & \\ \gamma_1 & \cdots & \gamma_6 \end{bmatrix} \qquad (19)$$

with $\boldsymbol{\gamma}_i = k\boldsymbol{\theta}_i\sigma_i + \mathbf{S}^\times(\mathbf{r}_i)\mathbf{v}_i$, $i = 1, \cdots, 6$ and k an aerodynamic constant that relates the thrust of the propeller with the torque produced around the rotation axis.

In the following, we present the results of the optimization problem to optimize the manipulability and energy task for the hovering condition (the typical working point), that is $\mathbf{w}^* = [0, 0, Mg, 0, 0, 0]^T$, with M the mass of the vehicle and g the gravitational acceleration. To simplify the problem, only the directions of the propellers are optimized, while $\mathbf{r}_i$ and σ_i are decided a priori.

Remark 3. The generalized approach with GA is so powerful that one can optimize almost any parameter, including the number of actuators, the position r_i, the spinning direction σ_i, the saturation levels $\overline{s}_i$, $\underline{s}_i$, etc.

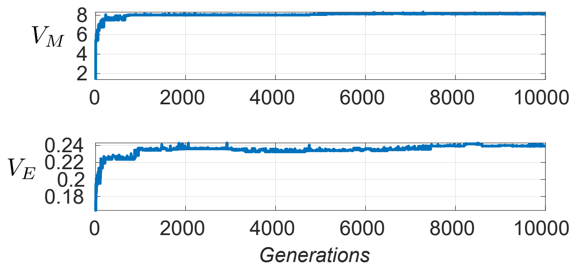


Fig. 5. The Fitness evolution for the *manipulability* and *energy* tasks.

The steps of the genetic algorithm are defined as the following:

- **Initial Population (IGA):** the initial population is selected randomizing $\mathbf{v}_i \in \mathbb{S}^2$.
- **Selection:** the selection mechanism used is the *fitness proportionate selection* Blickle and Thiele (1996).
- **Crossover:** the crossover mechanism consists in taking 3 chromosomes (three $\mathbf{v}_i$) from each parent.
- **Mutation (IGA):** in the mutation process each chromosome has a probability P_M of being randomized such that $\mathbf{v}_i \in \mathbb{S}^2$.

The parameters in Table. 2 were used in the simulations.

We considered 3 optimization problems:

- a) the cost function (11) for manipulability.
- b) the cost function (11) for manipulability but with a scale matrix $\mathbf{A}_s = diag\,\{1, 1, 1, 3, 3, 8\}$ to scale the torques w.r.t. the forces.
- c) the cost function (13) for energy and $\mathbf{w}^*$ a feasible wrench (i.e., $V_M > 0$).

In Fig. 5 the fitness function for the problems a) and c), while in Fig. 6 the visual representation of the optimal configurations. Finally, in Table 3 the optimized configurations for the problems a), b), c).

It is easy to note how the orientations of the configurations optimized for the manipulability are much more tilted, while the ones optimized for the energy are almost vertical. This corresponds to the capability of resisting an external wrench coming from any direction (in the first two cases), or just the one produced by gravity.

Furthermore, it is worth to notice that the configurations shaped for the manipulability use around twice and four times the energy of the configuration optimized for the energy. On the other hand, the configuration optimized for the energy can generate a very small wrench around the hovering condition.

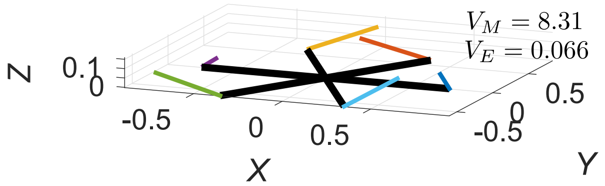
Given the consistent computational time required by the solving of the optimization problem with Genetic Algorithms w.r.t. other state-of-the-art techniques, we limited its usage to *offline* optimization, with a feasible $\mathbf{w}*$ precomputed for all the trajectory.

6. CONCLUSION AND FUTURE WORKS

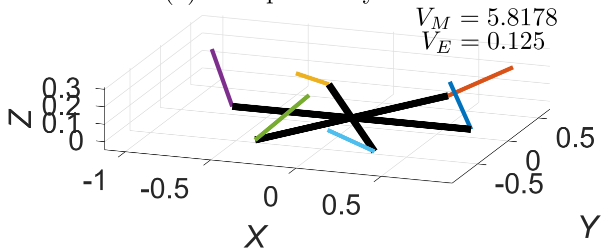
In this paper we considered fully-actuated or redundantly-actuated system, where a virtual input can be considered

Table 2. Parameters for the hexa-rotor simulation.

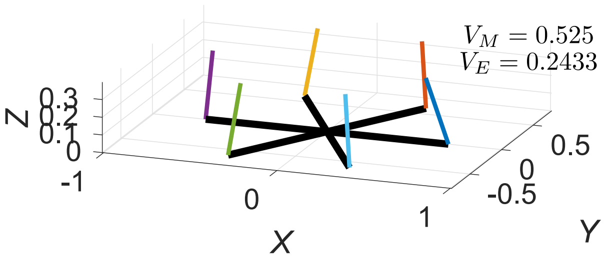
N_p	N_G	P_M	$\overline{s}_i$	$\underline{s}_i$	$\|r_i\|$	σ_i
500	10000	0.05	6.5 (N)	0(N)	0.7(m)	-1^i



(a) Manipulability task.



(b) Manipulability task with scaled allocation matrix.



(c) Energy task.

Fig. 6. The configurations of the hexarotor propellers for the three optimization problems.

as a linear combination of the *instantaneous inputs* of the system via an allocation matrix that depends on some *configuration* parameters. An optimization problem was defined, to optimize the *configurations* of the system to maximize or minimize a task-specific cost function, such as manipulability, energy consumption and volume of the obtainable virtual input. Due to the non-differentiable nature of the costs functions, the general problem was approached with Genetic Algorithms. The proposed solution allowed to optimize static parameters of the system for a design problem, as well as to find the optimal trajectory of the *configurations* in the *control* problem, where the configurations can be controlled by mean of some inputs. The approach was tested on two scenarios: finding the reconfiguration of the orientation of some actuators in an redundantly-actuated mobile robot, and on the design of the optimal configuration for a generalized hexa-rotor with arbitrary propeller orientation. The preliminary results were very promising.

Future works include: the definition of cost functions for other tasks, the application of the method to other robotic platforms for the design problem and the implementation of the allocation strategy on the real robotic platform in Nainer et al. (2017).

Table 3. Optimal configuration for the different tasks.

Manipulability	Manipulability Scaled	Energy
$\mathbf{v}_1 = [0.230, -0.89, 0.393]^T$	$\mathbf{v}_1 = [-0.561, 0.698, 0.443]^T$	$\mathbf{v}_1 = [-0.271, -0.150, 0.950]^T$
$\mathbf{v}_2 = [-0.967, -0.093, 0.234]^T$	$\mathbf{v}_2 = [0.876, 0.207, 0.434]^T$	$\mathbf{v}_2 = [-0.003, -0.149, 0.988]^T$
$\mathbf{v}_3 = [0.773, 0.582, 0.247]^T$	$\mathbf{v}_3 = [-0.157, -0.910, 0.382]^T$	$\mathbf{v}_3 = [0.197, -0.022, 0.980]^T$
$\mathbf{v}_4 = [0.542, -0.771, 0.331]^T$	$\mathbf{v}_4 = [-0.513, 0.599, 0.614]^T$	$\mathbf{v}_4 = [0.115, -0.034, 0.992]^T$
$\mathbf{v}_5 = [-0.969, 0.150, 0.195]^T$	$\mathbf{v}_5 = [0.727, 0.180, 0.662]^T$	$\mathbf{v}_5 = [0.109, 0.196, 0.974]^T$
$\mathbf{v}_6 = [0.391, 0.892, 0.223]^T$	$\mathbf{v}_6 = [-0.411, -0.778, 0.474]^T$	$\mathbf{v}_6 = [-0.129, 0.191, 0.972]^T$

REFERENCES

Blickle, T. and Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4, 361–394.

Boskovic, J. and Mehra, R. (2002). Control allocation in overactuated aircraft under position and rate limiting. In *in Proceedings of American Control Conference, 2002*.

Brescianini, D. and D'Andrea, R. (2016). Design, modeling and control of an omni-directional aerial vehicle. In *2016 IEEE Int. Conf. on Robotics and Automation*, 3261–3266. Stockholm, Sweden.

Bueler, B., Enge, A., and Fukuda, K. (2000). Exact volume computation for convex polytopes: a practical study. In *in PolytopesCombinatorics and Computation, DMV-Seminar*.

Chazelle, B. (1993). An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10, 377–409.

Fukuda, K. (2004). Frequently asked questions in polyhedral computation. `ftp://ftp.math.ethz.ch/users/fukudak/reports/polyfaq040618.pdf`.

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

LaValle, S.M. and J. J. Kuffner, J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20, 378–400.

Lindegaard, K.P. and Fossen, T. (2003). Fuel-efficient rudder and propeller control allocation for marine craft: Experiments with a model ship. *IEEE Transactions on Control Systems Technology*, 11, 850–862.

Nainer, C., Furci, M., Seuret, A., Zaccarian, L., and Franchi, A. (2017). Hierarchical control of the over-actuated rospo platform via static input allocation. In *20th IFAC World Congress*.

Oppenheimer, M., Doman, D., and Bolender, M. (2006). Control allocation for over-actuated systems. In *14th Mediterranean Conference on Control and Automation*. Ancona (Italy).

Park, S., J. Her, J., Kim, J., and Lee, D. (2016). Design, modeling and control of omni-directional aerial robot. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1570–1575. Daejeon, South Korea.

Passenbrunner, T.E., Sassano, M., and Zaccarian, L. (2016). Optimality-based dynamic allocation with nonlinear first-order redundant actuators. *European Journal of Control*, 3–40.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arxiv*.

Ryll, M., Bicego, D., and Franchi, A. (2016). Modeling and control of FAST-Hex: a fully-actuated by synchronized-tilting hexarotor. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1689–1694. Daejeon, South Korea.

Ryll, M., Blthoff, H.H., and Giordano, P.R. (2012). Modeling and control of a quadrotor uav with tilting propellers. In *in Proceedings of Robotics and Automation (ICRA), 2012*.

Sabharwal, A. and Potter, L. (1998). Convexly constrained linear inverse problems: iterative least-squares and regularization. *IEEE Transactions on Signal Processing*, 46, 2345–2352.

Srinivas, M. and Patnaik, L. (1994). Genetic algorithms: a survey. *Computer*, 27, 17–26.

Wolfe, P. (1976). Finding the nearest point in a polytope. *Mathematical Programming*, 11, 128–149.

Zhang, Z. (2012). Polyhedron under linear transformations. *arxiv*.