Design and Input Allocation for Robots with Saturated Inputs via Genetic Algorithms
Michele Furci, Davide Bicego, Antonio Franchi

To cite this version:
Michele Furci, Davide Bicego, Antonio Franchi. Design and Input Allocation for Robots with Saturated Inputs via Genetic Algorithms. 12th IFAC Symposium on Robot Control (SYROCO), Aug 2018, Budapest, Hungary. 6p. hal-01846467

HAL Id: hal-01846467
https://laas.hal.science/hal-01846467
Submitted on 22 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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:

$$S = \{x \in \mathbb{R}^n | Cx \leq b \}$$  \hspace{1cm} (1)

with $C \in \mathbb{R}^{q \times n}$, $b \in \mathbb{R}^q$, 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 $S^*(a)$ indicates the skew symmetric matrix such that $S^*(a) b = a \times b$ with $\times$ the cross product operator and $a, b$ two vectors of same dimension.

3. PROBLEM FORMULATION

Consider a dynamical system:

$$\dot{x} = f(x, w)$$  \hspace{1cm} (2)

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

$$w(t) = A(\theta(t)) u_p(t)$$  \hspace{1cm} (3a)

$$\dot{\theta}(t) = f_\theta(\theta, u_\theta(t))$$  \hspace{1cm} (3b)

with $A \in \mathbb{R}^{n \times p}$ the allocation matrix, $u_p \in \mathbb{R}^p$ a saturated vector of instantaneous inputs, $p \geq n$, $\theta(t) \in \mathbb{R}^n$ some time-variant parameters (the configurations), $u_\theta \in \mathbb{R}^p$ a vector of saturated inputs for the parameters $\theta$, $f_\theta(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^n$ a locally Lipschitz function. The input $u_p$ is such that $\xi_i \leq u_p, i \leq \bar{\xi}_i$, for $i = 1, \cdots, p$, with $\pi_i, \bar{\pi}_i \in \mathbb{R}$ the saturations limits. The input $u_\theta$ is such that $\overline{\xi}_{i, \theta} \leq u_\theta \leq \underline{\xi}_{i, \theta}$ for $i = 1, \cdots , p$, with $\pi_{i, \theta}, \bar{\pi}_{i, \theta} \in \mathbb{R}$ the saturations limits.

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

$$\begin{cases}
  u_p(t) = A^{-1}w^*(t) & p = n \\
  u_p(t) = A^1w^*(t) & p > n
\end{cases}$$  \hspace{1cm} (4)

where $A^1$ is the constrained pseudoinverse operator Sabharwal and Potter (1998) to guarantee that the solution lies in the feasible saturated $u_p$.

We define two problems:

- **Design Problem:** we suppose that $f_\theta(\theta, u_\theta) = 0$ and $p_\theta = 0$, i.e., the configurations $\theta$ are constant and there are no inputs to steer $\theta$. The goal is to find a static optimal configuration $\theta^*$ that maximizes/minimizes a cost function $V(\theta) \in \mathbb{R}$. 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:

$$\max/\min_{\theta} \quad V(\theta)$$  \hspace{1cm} (5) 

subject to

$$\begin{cases}
  \xi_i \leq u_{p,i} \leq \bar{\xi}_i \\
  \xi_{i, \theta} \leq u_{\theta,i} \leq \bar{\xi}_{i, \theta}
\end{cases}$$  \hspace{1cm} (3)

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

$$\max/\min_{\theta(t), u_{\theta}(t)} \quad \int_0^{t_f} V(\theta(t)) dt$$  \hspace{1cm} (6)

subject to

$$\begin{cases}
  \xi_i \leq u_{p,i} \leq \bar{\xi}_i \\
  \xi_{i, \theta} \leq u_{\theta,i} \leq \bar{\xi}_{i, \theta}
\end{cases}$$  \hspace{1cm} (3)

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

$$U = \{ x \in \mathbb{R}^p | C_U x \leq b_U \}$$  \hspace{1cm} (7)

with $C_U \in \mathbb{R}^{2p \times p}$ defined as:

$$C_U = \begin{bmatrix} I_p \hline -I_p \end{bmatrix}$$  \hspace{1cm} (8)

with $I_p$ an identity matrix of dimension $p$, and $b_U$ defined as:

$$b_U = [\bar{\xi}_1 \cdots \bar{\xi}_p - \xi_1 \cdots - \xi_p]^T$$  \hspace{1cm} (9)

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

$$W = \{ x \in \mathbb{R}^n | C_W x \leq b_W \}.$$  \hspace{1cm} (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 $w^*(t)$. The goal is to find the optimal configuration $\theta^*(t)$ and the configuration input $u_\theta$ such that the manipulability index around $w^*(t)$ is the maximum, considering as manipulability index the radius of the largest sphere inscribed into the polytope $W$ and centered in $w^*(t)$. 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 $w^*(t)$ to compensate external disturbances or disturbances due to non perfect modeling.

If we define as $d_t$ the distance of $w^*(t)$ to the i-th hyperplane that constraints $W$, we define the cost function as:

$$V_M(\theta) = \min(d_i(\theta)), \quad i = 1, \cdots, t, t \leq 2p$$  \hspace{1cm} (11)

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

$$A_\lambda = \begin{bmatrix} \lambda_1 & 0 \\
0 & \ddots \\
0 & 0 & \lambda_n \end{bmatrix}$$  \hspace{1cm} (12)

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

**Energy task:** a desired virtual input trajectory is given by $w^*(t)$. From (4) we can compute $u_p(t)$. 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:

\[
\begin{align*}
V_E(\theta) &= \|A^{-1}(\theta)w^*(t)\|^{-1} \quad \text{while } w^*(t) \in \mathcal{W} \\
V_E(\theta) &= \|A^{-1}(\theta)w^*(t)\|^{-1} \quad \text{while } w^*(t) \notin \mathcal{W}
\end{align*}
\]

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

\[
\begin{align*}
V_E(\theta) &= \|A^\dagger(\theta)w^*(t)\|^{-1} \quad \text{while } w^*(t) \in \mathcal{W} \\
V_E(\theta) &= \|A^\dagger(\theta)w^*(t)\|^{-1} \quad \text{while } w^*(t) \notin \mathcal{W}
\end{align*}
\]

where \(w^*(t)\) 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 \(w^*\) is feasible using the configuration obtained.

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

\[
V_V(\theta) = \text{vol}(\mathcal{W})
\]

where \(\text{vol}(\mathcal{W})\) is the volume of \(\mathcal{W}\).

The cost function can be defined as weighted sum of different tasks, i.e., we can define \(V = \beta_1 V_M + \beta_2 V_E + \beta_3 V_V\), with \(\beta_i\) 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 \(\mathcal{W}\) as function of \(U\) and \(A(\theta)\). The solution is different in case of fully actuated \((p = n)\) or in case of redundant actuation \((p > n)\):

- **Fully Actuation** \((p = n)\): if \(A(\theta)\) is invertible, we can compute \(b_{\mathcal{W}} = b_x\) and \(C_{\mathcal{W}} = C_{U}A(\theta)^{-1}\) (closed form solution). If \(A(\theta)\) is non-invertible, the cost function \(V_M = 0\).
- **Redundant Actuation** \((p > n)\): since \(A\) 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 \(U\) in its V-representation (the vertex representation), transform the vertices of \(U\) via \(A(\theta)\), compute the V-representation of \(\mathcal{W}\) as convex hull of the transformed points using Chazelle (1993). The problem is known as the facet enumeration problem Fukuda (2004).

Once \(\mathcal{W}\) is defined, we can compute \(d_i\) as

\[
d_i = \frac{|C_{\mathcal{W}}w^* - b_{\mathcal{W}}|}{\|C_{\mathcal{W}}\|}
\]

where the \(i\) index refers to the \(i\)-th row or the \(i\)-th hyperplane defining \(\mathcal{W}\).

4.2 Energy

- **Fully Actuation** \((p = n)\): the cost function requires the computation of \(\mathcal{W}\) which can be done in closed form and the computation of \(A^{-1}\) which again can be done in closed form from \(A(\theta)\). Some gradient-based algorithms can be used to compute the optimal \(\theta^*\).

- **Redundant Actuation** \((p > n)\): both the computation of \(\mathcal{W}\) and \(A^\dagger\) 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 \(\mathcal{W}\) 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 \(\theta\). 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 \(\mathcal{W}\) and/or \(A^\dagger\)) 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 \(\theta^*\). In the Control problem, instead, we discretize \(w^*(t)\) with a given time step \(\Delta t\), 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 \(\theta^*(0)\) for the Control problem. The TGA is instead used along the trajectory to find the \(\theta^*(t > 0)\). Since the TGA has to obey (3b) we need to find a suitable mechanism to find feasible solutions of \(\theta^*(t)\). 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 \(\theta_k^* = \theta_{k-1}^* + f_0(u_{k,i})\). Suitable strategies to pick a feasible \(\theta^*\) can be derived from kimo-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 \(w, \theta, A, u_i, f_0, u_0\). In particular \(w = [F_x, F_y, \tau]^T\) is the wrench vector, where \(F_x\) is the body force along x-axis, \(F_y\) is the body force along y-axis and \(\tau\) is the torque around the vertical axis; \(\theta = [\theta_1, \ldots, \theta_4]\) with \(\theta_i\) the...
angle of the i-th turret: \( u_i = [k_1 \omega_1^2, k_2 \omega_2^2, k_3 \omega_3^2, k_4 \omega_4^2]^T \), where \( k_i \) is an aerodynamic constant and \( \omega_i \) the angular velocity of the i-th propeller. \( f(\theta, u_\theta) = u_\theta, \ u_\theta = [u_{\theta,1}, u_{\theta,2}, u_{\theta,3}, u_{\theta,4}]^T \) 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:

\[
A(\theta) = \begin{bmatrix}
c(\theta_1) & \cdots & c(\theta_4) \\
s(\theta_1) & \cdots & s(\theta_4) \\
(\Pi_1)^T & \cdots & (\Pi_4)^T
\end{bmatrix}
\]

(17)

where \( \Pi = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \), \( r_i \in \mathbb{R}^2 \) 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:

\[
\mathbf{w}^*(t) = \begin{bmatrix}
F_x^* \\
F_y^* \\
T
\end{bmatrix}
= \begin{bmatrix}
-M f_2 R \cos(f_1 t) \\
-M f_2 R \sin(f_2 t) \\
0
\end{bmatrix}
\]

(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 \( \theta \) 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 \( \theta_i = \theta_0^i \pm \delta_{\theta,i} \), where \( \theta_0^i \) is the optimal value of \( \theta_i \) from the previous iteration and \( \delta_{\theta,i} \) is the maximum variation of \( \theta_i \) in \( \Delta t \) given a limitation of the \( u_{\theta,i} \). Since (3b) is a single integrator model \( \delta_{\theta,i} = \pi \delta_{\theta,i} \).
- **Selection:** the selection mechanism used is the fitness proportionate selection Blickle and Thiele (1996).
- **Crossover:** the crossover mechanism consists in taking 2 chromosomes (two \( \theta_i \)) 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.](image)

![Fig. 2. Fitness functions for the IGA problem in both manipulability and energy tasks.](image)

![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.](image)

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)
instead have tilted actuators to generate forces in 3 axes and torques in 3 axes. Inspired by these 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 $\mathbf{v}_i \in S^2$ the orientation of the i-th actuator in body frame, where the group $S^2$ defines the unit sphere group $S^2 = \{ x \in \mathbb{R}^{n+1} | ||x|| = 1 \}$. 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 $\mathbf{w}, \theta, \mathbf{A}, \mathbf{u}_p, f_p, \mathbf{u}_g$. In particular $\mathbf{w} = [F, \tau]^T$ is the wrench vector, where $F \in \mathbb{R}^3$ is the body force in three dimensions $x,y,z$ and $\tau \in \mathbb{R}^3$ the torque in three dimensions. $\theta = [\theta_1, \cdots, \theta_6], \mathbf{u}_p = [k_\omega \omega_1 \cdots, k_\omega \omega_6]^T$, where $k_\omega$ is an aerodynamic constant and $\omega$ the angular velocity of the i-th propeller. We consider the design problem, so $\theta = 0$. Finally the allocation matrix is defined as:

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

with $\gamma_i = k \theta_i \sigma_i + S^5(r_i) \mathbf{v}_i, i = 1, \cdots, 6$ 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 $\mathbf{w}^* = [0, 0, M g, 0, 0, 0]^T$, 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 $\sigma_i$ 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 $\mathbf{r}_i$, the spinning direction $\sigma_i$, the saturation levels $S_i, \bar{S}_i$, etc.

The parameters in Table 2 were used in the simulations.

![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.](image)

![Fig. 5. The Fitness evolution for the manipulability and energy tasks.](image)

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

- **Initial Population (IGA):** the initial population is selected randomizing $\mathbf{v}_i \in S^2$.
- **Selection:** the selection mechanism used is the fitness proportionate selection Blecikle 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 $\mathbf{v}_i \in S^2$.

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 $\mathbf{w}^*$ pre-computed 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

| $N_p$ | $N_G$ | $P_M$ | $\sigma_i$ | $\|\mathbf{r}_i\|$ | $\gamma_i$ |
|-------|-------|-------|------------|----------------|--------|
| 500   | 10000 | 0.05  | 6.5 (N)    | 0(N)           | 0.7 (m)| $-1^4$ |

**Table 2. Parameters for the hexa-rotor simulation.**

Preprint version, 5
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 optimal configuration for a generalized hexa-rotor with the three optimization problems.

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 |
|----------|----------------|-----------------------|--------|
| $v_x$    | $[-0.258, -0.059, 0.307]$ | $[-0.258, -0.059, 0.307]$ | $[-0.258, -0.059, 0.307]$ |
| $v_y$    | $[0.561, 0.169, 0.446]$ | $[0.561, 0.169, 0.446]$ | $[0.561, 0.169, 0.446]$ |
| $v_z$    | $[-0.159, -0.904, 0.386]$ | $[-0.159, -0.904, 0.386]$ | $[-0.159, -0.904, 0.386]$ |
| $V_M$    | $[0.542, 0.771, 0.331]$ | $[0.542, 0.771, 0.331]$ | $[0.542, 0.771, 0.331]$ |
| $V_E$    | $[-0.513, 0.399, 0.641]$ | $[-0.513, 0.399, 0.641]$ | $[-0.513, 0.399, 0.641]$ |

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