

# A Direct Memetic Approach to the Solution of Multi-Objective Optimal Control Problems

Massimiliano Vasile

Department of Mechanical & Aerospace Engineering  
University of Strathclyde  
Glasgow, UK  
Email: massimiliano.vasile@strath.ac.uk

Lorenzo Ricciardi

Department of Mechanical & Aerospace Engineering  
University of Strathclyde  
Glasgow, UK  
Email: lorenzo.ricciardi@strath.ac.uk

**Abstract**—This paper proposes a memetic direct transcription algorithm to solve Multi-Objective Optimal Control Problems (MOOCP). The MOOCP is first transcribed into a Non-linear Programming Problem (NLP) with Direct Finite Elements in Time (DFET) and then solved with a particular formulation of the Multi Agent Collaborative Search (MACS) framework. Multi Agent Collaborative Search is a memetic algorithm in which a population of agents combines local search heuristics, exploring the neighbourhood of each agent, with social actions exchanging information among agents. A collection of all Pareto optimal solutions is maintained in an archive that evolves towards the Pareto set. In the approach proposed in this paper, individualistic actions run a local search, from random points within the neighbourhood of each agent, solving a normalised Pascoletti-Serafini scalarisation of the multi-objective NLP problem. Social actions, instead, solve a bi-level problem in which the lower level handles only the constraint equations while the upper level handles only the objective functions. The proposed approach is tested on the multi-objective extensions of two well-known optimal control problems: the Goddard Rocket problem, and the maximum energy orbit rise problem.

## I. INTRODUCTION

In many practical cases in which a dynamical system needs to be controlled, it is desirable to generate multiple alternative solutions that are optimal with respect to a number of conflicting cost functions. In the literature, few approaches have been proposed to tackle multi-objective optimal control problems: in Coverstone et al. [1] the authors combined Genetic Algorithms and optimal control theory in a dual loop algorithm. In the outer loop, NSGAI was generating vectors of co-states and times of flight. For each set, the inner loop was solving a single objective optimal control problem with given time of flight, minimising the propellant consumption. In Ober-Blobaum et al. [2] a direct transcription approach is used, coupled with an approach that scalarises the multi-objective vector along directions pointing at predefined unreachable points in the criteria space. Each scalar problem is then solved with a standard NLP solver. In [3] a similar approach is proposed that uses a smoothed version of Tchebycheff scalarisation to scalarise the MOO problem. In [4] the authors proposed a dual loop algorithm in which the outer loop solves a multi-objective problem handling a set of categorical variables and the inner loop solves a set of single objective constrained optimal control problems using Monotonic Basin Hopping.

This paper proposes a memetic approach to find a set of Pareto optimal control policies that satisfy a set of dynamic and algebraic constraints. The memetic solver is based on the Multi-Agent Collaborative Search framework [5]. The optimal control problem is first transcribed into a Non-linear Programming Problem with Direct Finite Element Transcription (DFET) [6] and then into two optimisation problems: a bi-level optimisation where constraints are satisfied in an inner level and objectives are handled at an outer level, and a single level optimisation problem that works on a scalar version of the original constrained multi-objective optimisation problem.

The single level approach exploits the ability of MACS to generate displacements (in criteria space) within descent cones that slide along predefined directions. A normalised Pascoletti-Serafini scalarisation is used to turn the original multi-objective NLP problem into a single objective problem with a given direction in criteria space. This scalarisation does not suffer from the inability to converge to non convex regions of the Pareto front, typical of the weighted sum scalarisation, and, in the implementation proposed in this paper, preserves the extremal values of each individual objective function.

This paper is structured as follows: section II briefly gives the mathematical description of multi-objective optimal control problems, section III explains the DFET transcription method, section IV describes the solution approach and section V presents some numerical tests. Finally section VI draws conclusions.

## II. MULTI OBJECTIVE OPTIMAL CONTROL PROBLEM FORMULATION

This paper is concerned with the following multi-objective optimal control problem:

$$\begin{aligned} \min_{\mathbf{u} \in U} \mathbf{J} &= (J_1, J_2, \dots, J_i, \dots, J_m) \\ \text{s.t.} \\ \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{g}(\mathbf{x}, \mathbf{u}, t) &\geq 0 \\ \psi(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) &\geq 0 \\ t &\in [t_0, t_f] \end{aligned} \quad (1)$$

where  $\mathbf{J}$  is a vector of objectives  $J_i$ , that are functions of the state variable  $\mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^n$ , control variable  $\mathbf{u} \in L^\infty$  and time  $t$ . The functions  $\mathbf{x}$  belong to the Sobolev space

$W^{1,\infty}$  while the objective functions are  $J_i : \mathbb{R}^{n+2} \times \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$ . The objective vector is subject to a set of dynamic constraints with  $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}^n$ , algebraic constraints  $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}^s$ , and boundary conditions  $\mathbb{R}^{2n+2} \rightarrow \mathbb{R}^q$ . Note that problem (1) can include a number of additional static parameters, is generally non-smooth and can have many local minima.

### III. OPTIMAL CONTROL PROBLEM TRANSCRIPTION

Problem (1) is here translated into a multi-objective non-linear programming problem via DFET transcription. DFET transcription for optimal control problems was initially proposed in [7] and uses finite elements in time on spectral bases to transcribe the differential equations into a set of non-linear algebraic equations. Finite Elements in Time for the indirect solution of optimal control problems were initially proposed by Hodges et al. in [8], and during the late 1990s evolved to the discontinuous version. The motivation for the use of DFET for the transcription of optimal control problems is twofold. First, as pointed out by Bottasso et al. in [9], FET for the forward integration of ordinary differential equations are equivalent to some classes of implicit Runge-Kutta integration schemes, can be extended to arbitrary high-order, are very robust (i.e. they can be used for the long term propagation of stiff problems) and allow full h-p adaptivity. Second, in the past decade direct transcription with FET on spectral bases has been successfully used to solve a range of difficult single objective optimal control problems: from the design of low-thrust multi-gravity assist trajectories to Mercury [10], to the Sun [11], to the design of transfers to the Moon in the full 4-body problem, low-thrust transfers in the restricted three body problem and optimal landing trajectories to the Moon [7].

For each individual cost function let's here consider the following typical optimal control problem (known as Bolza's problem):

$$\min_{\mathbf{u} \in \mathcal{U}} J_i = \alpha_i \phi_i(\mathbf{x}_0, \mathbf{x}_f, \mathbf{u}_0, \mathbf{u}_f, t_0, t_f) + \beta_i \int_{t_0}^{t_f} L_i(\mathbf{x}, \mathbf{u}, t) dt \quad (2)$$

where  $\alpha_i$  and  $\beta_i$  are positive weights,  $\phi_i$  is a general function of initial and final time and corresponding initial and final time states and controls, and  $L_i$  is a general function of their time history with a possible explicit time dependence. From a multi-objective optimisation perspective, this formulation corresponds to a weighted sum scalarisation, which is known to be unable to represent points on non-convex regions of the Pareto front. Therefore, to avoid this problem and without loss of generality, in this paper only cases with  $(\alpha_i = 1, \beta_i = 0)$  or  $(\alpha_i = 0, \beta_i = 1)$  will be considered: the multiplicity of objectives will be handled by the multi-objective optimisation algorithm. The differential constraints can be recast in weak form and integrated by parts, leading to

$$\int_{t_0}^{t_f} \dot{\mathbf{w}}^T \mathbf{x} + \mathbf{w}^T \mathbf{F}(\mathbf{x}, \mathbf{u}, t) dt - \mathbf{w}_f^T \mathbf{x}_f + \mathbf{w}_0^T \mathbf{x}_0 = 0 \quad (3)$$

where  $\mathbf{w}$  are generalised weight functions and  $\mathbf{x}^b$  are the boundary state values. Let the time domain  $D$  be decomposed into  $N$  finite elements, such that  $D = \bigcup_{j=1}^N D_j(t_{j-1}, t_j)$ , and parametrise, over each  $D_j$ , the states, controls and weight functions as

$$\mathbf{x}(t) = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \mathbf{X}_j = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \sum_{s=0}^l f_{sj}(t) \bar{\mathbf{x}}_{sj} \quad (4)$$

$$\mathbf{u}(t) = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \mathbf{U}_j = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \sum_{s=0}^m g_{sj}(t) \bar{\mathbf{u}}_{sj} \quad (5)$$

$$\mathbf{w}(t) = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \mathbf{W}_j = \overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}} \sum_{s=0}^{l+1} h_{sj}(t) \bar{\mathbf{w}}_{sj} \quad (6)$$

where  $\overset{\cup}{\underset{j=1}{\overset{N}{\mathcal{U}}}}$  denotes the juxtaposition of the polynomials defined over each sub-interval,  $f_{sj}(t)$ ,  $g_{sj}(t)$  and  $h_{sj}(t)$  indicate the  $s$ -th polynomial over element  $j$  and are chosen among the space of polynomials of degree  $l$ ,  $m$  and  $l+1$  respectively, while  $\bar{\mathbf{x}}_{sj}$ ,  $\bar{\mathbf{u}}_{sj}$  and  $\bar{\mathbf{w}}_{sj}$  denote the nodal values of the states, control and test functions. It is practical to define each  $D_j$  over the normalised interval  $[-1, 1]$  through the transformation  $\tau = 2 \frac{t - \frac{t_j - t_{j-1}}{2}}{t_j - t_{j-1}}$ . This way it is easy to express the polynomials  $f_{sj}(t)$ ,  $g_{sj}(t)$  and  $h_{sj}(t)$  as the Lagrange interpolation on Gauss nodes in the normalised interval:

$$f_{sj}(t) = \tilde{f}_{sj}(\tau) = \prod_{k=0, k \neq s}^l \frac{\tau - \tau_k}{\tau_s - \tau_k} \quad (7)$$

where  $\tau_*$  indicates a Gauss node. Similarly it can be done for  $g_{sj}$  and  $h_{sj}$ . Different Gauss nodes will lead to schemes with slightly different characteristics. In this work, Gauss-Lobatto nodes will be used for the generation of the polynomials for states and weight functions, while Gauss-Legendre nodes will be used for the controls. Substituting the definitions of the polynomials into the objective functions and integrating with Gauss quadrature formulas leads to objective function

$$\tilde{J}_i = \alpha_i \phi_i(\mathbf{X}_0^b, \mathbf{X}_f^b, \mathbf{U}_0^b, \mathbf{U}_f^b, t_0, t_f) + \beta_i \sum_{j=1}^N \sum_{k=1}^{l+1} \sigma_k L_i(\mathbf{X}_j(\tau_k), \mathbf{U}_j(\tau_k), \tau_k) \frac{\Delta t_j}{2} \quad (8)$$

and, for each element  $j$ , constraints

$$\sum_{k=1}^{l+1} \sigma_k \left[ \dot{\mathbf{W}}_j(\tau_k)^T \mathbf{X}_j(\tau_k) + \mathbf{W}_j(\tau_k)^T \mathbf{F}_j(\tau_k) \frac{\Delta t_j}{2} \right] - \mathbf{W}_{p+1,j}^T \mathbf{X}_j^b + \mathbf{W}_{1,j}^T \mathbf{X}_{j-1}^b = 0 \quad (9)$$

where  $\tau_k$  and  $\sigma_k$  are Gauss nodes and weights, and  $\mathbf{F}_j(\tau_k)$  is the shorthand notation for  $\mathbf{F}(\mathbf{X}_j(\tau_k), \mathbf{U}_j(\tau_k), \tau_k)$ . Gauss-Legendre weights and nodes are used for the numerical quadrature in this work, i.e., the polynomials generated through the Lagrange interpolation over the Gauss-Legendre or Gauss-Lobatto nodes are then evaluated at the Gauss-Legendre nodes over each interval. Inter element matching conditions, weakly enforcing continuity of the states, will

naturally remove the dependency on  $\mathbf{X}_j^b$  for all elements except for the first and last ones, so that only  $\mathbf{X}_1^b$  and  $\mathbf{X}_N^b$  remain. With DFET, optimal control problem (2) was transcribed into the following non-linear programming problem:

$$\begin{aligned} & \min_{\mathbf{p}} \tilde{\mathbf{J}}(\bar{\mathbf{x}}, \mathbf{p}) \\ & s.t. \\ & \mathbf{c}(\bar{\mathbf{x}}, \mathbf{p}) \geq 0 \end{aligned} \quad (10)$$

where the vector  $\bar{\mathbf{x}}$  contains all the nodal values for the states except the boundary ones and  $\mathbf{p} = [\bar{\mathbf{u}}, \mathbf{x}_0, \mathbf{x}_f, t_0, t_f]^T$  collects all the static and dynamic control variables. In general,  $t_0, t_f, \mathbf{x}_0$  and  $\mathbf{x}_f$  can be free and thus regarded as continuous static control variables.

#### IV. SOLUTION WITH MULTI-AGENT COLLABORATIVE SEARCH

Multi-Agent Collaborative Search is a meta-heuristics to combine local and global search heuristics. A set of agents is endowed with a list of possible actions that can involve other agents or simply collect information on a neighbourhood of each agent. In MACS2 the idea of search directions was introduced in the logic of the agents which could select new candidate solutions according to either dominance or Tchebycheff scalarisation. The same logic and the ability of the agents to incorporate local gradient-based actions are here exploited to solve problem (10). The general MACSoc (MACS for optimal control) scheme is summarised in Algorithm 1. The individualistic and social actions are described in the following section and are related to the solution of two different problems. The population  $P_0$  (Line 1 in Algorithm 1) is initialised randomly with Latin Hypercube sampling, while the weights of Tchebycheff scalarisation  $\lambda$  (Line 2 in Algorithm 1) are generated as in section IV-B. After performing individualistic and social actions (lines 4 and 7 in Algorithm 1) both the population and the archive are updated. The filtering process (Lines 6 and 9 in Algorithm 1) that updates the global archive  $A_g$ , where all Pareto optimal solutions are stored, is redistributing solutions so that a pseudo-electric potential function (function of the reciprocal distance of the elements in the archive) is minimised (see [12] for further details). Finally, at each iteration, the descent direction (or scalar subproblem) allocated to each agent is updated (line 10 in Algorithm 1).

##### A. Problem Formulation in the MACS Framework

In order to solve (10) with MACS, the problem is transcribed in two different forms amenable to a solution with either the individualistic or social actions. The first form is a normalised version of the Pascoletti-Serafini scalarisation, also known as goal attainment method. Using the Pascoletti-Serafini scalarisation, for each agent  $j$ , problem (10) is transcribed into:

$$\begin{aligned} & \min_{\alpha > 0} \quad \alpha \\ & s.t. \\ & \lambda_{ij} \vartheta_{ij}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \leq \alpha \quad i = 1, \dots, m \\ & \mathbf{c}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \geq 0 \end{aligned} \quad (11)$$

---

#### Algorithm 1 MACSoc Framework

---

- 1: Initialise population  $P_0$  and global archive  $A_g$
  - 2: Initialise search directions  $\mathbf{d}$  and Tchebycheff weights  $\lambda$
  - 3: **while**  $n_{feval} < max\_fun\_eval$  **do**
  - 4:   Run individualistic heuristics on  $P_k$
  - 5:    $P_k \rightarrow P_k^+$
  - 6:   Update archive  $A_g$  with potential field filter
  - 7:   Run social heuristics combining  $P_k^+$  and  $A_g$
  - 8:    $P_k^+ \rightarrow P_{k+1}$  based on Tchebycheff scalarisation
  - 9:   Update archive  $A_g$  with potential field filter
  - 10:   Update subproblem allocation
  - 11: **end while**
- 

where  $\lambda_j$  is the vector of Tchebycheff weights associated to agent  $j$ ,  $\vartheta_{ij}$  is the  $i$ -th component of the rescaled objective vector of the  $j$ -th agent and  $\alpha$  is a slack variable. This reformulation of the problem is constraining the  $j$ -th agent the move, in criteria space, within the descent cone defined by the point  $\alpha \mathbf{d}_j + \zeta_j$  along the direction  $\mathbf{d}_j = (1/\lambda_{1j}, \dots, 1/\lambda_{ij}, \dots, 1/\lambda_{mj})$ . The rescaled objective vector is

$$\vartheta_{ij}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) = \frac{\tilde{J}_{ij}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) - \tilde{z}_i}{z_{ij}^* - \tilde{z}_i} \quad i = 1, \dots, m \quad (12)$$

where  $z_j^*$  is equal to  $\tilde{J}_j(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  and  $(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  is the initial guess for the solution of (11). This way the components of  $\vartheta_j(\bar{\mathbf{x}}, \bar{\mathbf{p}})$  have value 1 at the beginning of the local search and if the agent converges to the utopia point  $\tilde{\mathbf{z}}$ , the components of  $\vartheta_j(\bar{\mathbf{x}}, \bar{\mathbf{p}})$  become all equal to 0. The choice of  $\lambda_j$  and  $\tilde{\mathbf{z}}$  will be discussed in the following subsection. From the normalisation one can derive the components of the vector  $\zeta_j$ :

$$\zeta_{ij} = \frac{z_i}{z_{ij}^* - \tilde{z}_i} \quad i = 1, \dots, m \quad (13)$$

The presence of the rescaling of the objectives, together with the choice of  $\lambda_j$  and  $\tilde{\mathbf{z}}$ , are the elements that distinguish the proposed approach from the one given in [3]. Note that solving problem (11) already provides a non-dominated solution that can be potentially inserted in  $A_g$  and used to update  $P_k$ . The pseudocode can be found in Algorithm 2. Note that the size of the neighbourhood  $B_j$  is given by the parameter  $\rho_j$ . The point  $(\bar{\mathbf{x}}, \mathbf{p}^c)$  is taken at random in  $B_j$  if  $(\bar{\mathbf{x}}, \mathbf{p})_j$  did not change from the previous iteration, otherwise  $(\bar{\mathbf{x}}, \mathbf{p}^c) = (\bar{\mathbf{x}}, \mathbf{p})_j$ . If the local search returns an infeasible solution a penalty value  $M$  is assigned to all cost functions and the solution is rejected.

The second form is a bi-level formulation of problem (10) in which the upper level is handling only the objective functions and the lower level the constraint functions. Problem (10) is then transcribed for each agent  $j$  in the following general bi-level optimisation problem:

$$\begin{aligned} & \min_{\mathbf{p}^*} \tilde{\mathbf{J}}_j(\bar{\mathbf{x}}^*, \mathbf{p}^*) \\ & s.t. \\ & (\bar{\mathbf{x}}^*, \mathbf{p}^*)_j = \operatorname{argmin}\{f(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c) | \mathbf{c}(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c) \geq 0\} \end{aligned} \quad (14)$$

where  $f = 1$  in this implementation and  $(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  is a candidate solution generated with a Differential Evolution (DE) operator.

The DE operator is applied to a mix of agents associated to a particular weight  $\lambda$  and elements of the archive  $A_g$ . If the inner level returns a feasible solution, that solution is selected for possible inclusion in the population  $P_{k+1}$  using Tchebycheff criterion (line 8 of Algorithm 1). The pseudocode for both levels can be found in Algorithms 3 and 4. Note that if the inner level return an infeasible solution a penalty value  $M$  is assigned to the cost functions and the solution is rejected.

### B. Selection of $\lambda$ and $\tilde{\mathbf{z}}$

In [3], the MOOCP was tackled by first solving each of the two individual objectives, and then choosing a set of evenly spaced weights, obtaining a set of directions  $\mathbf{d}$ . This approach has a two main limitations: first, since only a local strategy was employed, there is the possibility that the extreme values of the Pareto front generated are on a local Pareto front. Second, that approach is not easy to generalise for more than two objectives: as already stated by its authors regarding the extension to the three objective case: "it is well known that the boundary of the Pareto front may very well lie outside the triangle formed by the three points". The proposed approach instead consists in assigning vector  $\lambda_i = (0, 0, i, \dots, 0, 0)$  to agents solving subproblem  $i$  and vector  $\lambda_j = \frac{(1, 1, 1, \dots, 1)}{\|(1, 1, 1, \dots, 1)\|}$  to all the other agents. The modified utopia point  $\tilde{\mathbf{z}}$  is given by

$$\tilde{\mathbf{z}} = 2\mathbf{z} - \mathbf{z}^*_A \quad (15)$$

where  $\mathbf{z}$  and  $\mathbf{z}^*_A$  are respectively the utopia and nadir points of the current approximation to the Pareto front that is contained in the archive  $A_g$ . When an agent  $j$  solving subproblem  $i$  has locally converged and is not displaced by any action, its subrproblem is updated with  $\lambda_j = \frac{(1, 1, 1, \dots, 1)}{\|(1, 1, 1, \dots, 1)\|}$ , conversely an agent associated to  $\lambda_j = \frac{(1, 1, 1, \dots, 1)}{\|(1, 1, 1, \dots, 1)\|}$  that has locally converged and is not displaced by any action will have its subproblem replaced with  $\lambda_i = (0, 0, i, \dots, 0, 0)$  (line 10 in Algorithm 1).

## V. NUMERICAL TESTS

In this section we test the proposed direct memetic approach on the solution of two simple test cases: the first test case is the Goddard Rocket problem and the second is the maximum energy orbit rise problem. For the former an analytical solution is available for a single objective optimisation problem and will be used to construct the exact Pareto front. For the latter the exact control law is known but an analytical expression for the Pareto front is not available although it can be empirically derived collecting the solutions from multiple runs. The algorithm was run 30 times on each problem to gather statistics on the quality of the Pareto front. The local NLP solver is the Matlab *fmincon* function.

### A. Goddard's rocket problem

The Goddard's rocket problem is to find an optimal ascent trajectory from a flat celestial body with no atmosphere to a prescribed altitude. The control variable is the thrust angle and both gravity and thrust accelerations are constant. The final

---

### Algorithm 2 Individualistic Action

---

```

1: Set  $\tilde{\mathbf{z}} = 2\mathbf{z} - \mathbf{z}^*_A$ 
2: if current agent is solving problem  $i$  only then
3:    $\lambda_j = (0, 0, i, \dots, 0, 0)$ 
4: else
5:    $\lambda_j = \frac{(1, 1, 1, \dots, 1)^T}{\|(1, 1, 1, \dots, 1)^T\|}$ 
6: end if
7: Pick a point  $(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  in  $B_j$ 
8: Run local search from  $(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  to solve Problem (11) and
   find solution  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j$ 
9: if  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j$  feasible then
10:  Return  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j, \tilde{\mathbf{J}}_j(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)$  and increase  $\rho_j < 1$ 
11: else
12:  if number of times  $\rho_j$  is reduced  $> \max\_contr\_ratio$ 
13:    then
14:      $\rho_j = 1$ 
15:  else
16:    Reduce  $\rho_j$  and return  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j = (\bar{\mathbf{x}}, \bar{\mathbf{p}})_j, \tilde{\mathbf{J}}_j = M$ 
17:  end if
18: end if

```

---



---

### Algorithm 3 Social Action

---

```

1: Select weight  $\lambda$ 
2: Select agents associated with  $\lambda$  and elements of the archive
    $A_g$ 
3: Apply DE operator to selected agents and elements of the
   archive and generate candidate solution  $\mathbf{u} = (\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$ 
4: Run inner level on  $\mathbf{u}$ 

```

---



---

### Algorithm 4 Inner level

---

```

1: Run local search from  $(\bar{\mathbf{x}}, \bar{\mathbf{p}}^c)$  to find solution  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j$ 
2: if  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j$  feasible then
3:  Return  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j, \tilde{\mathbf{J}}_j(\bar{\mathbf{p}}^*, \bar{\mathbf{x}}^*)$ 
4: else
5:  Return  $(\bar{\mathbf{x}}^*, \bar{\mathbf{p}}^*)_j = (\bar{\mathbf{x}}, \bar{\mathbf{p}})_j, \tilde{\mathbf{J}}_j = M$ 
6: end if

```

---

altitude is assigned and the final vertical component of the velocity has to be zero. The single objective optimal control formulation of the problem and its analytical solutions for either minimum time or maximum horizontal velocity can be found in [13], while a numerical solution with DFET can be found in [6].

In this paper, the problem is reformulated as follows, to consider the two objectives simultaneously:

$$\min_{t_f, u} (J_1 = t_f, J_2 = -v_x(t_f)) \quad (16)$$

subject to the dynamic constraints:

$$\begin{cases} \dot{x} &= v_x \\ \dot{v}_x &= a \cos u \\ \dot{y} &= v_y \\ \dot{v}_y &= -g + a \sin u \end{cases} \quad (17)$$

TABLE I  
MACSOC SETTINGS

$max\_fun\_eval$	10000
$pop\_size$	10
$\rho\_ini$	1
$F$	0.9
$CR$	0.9
$p\_social$	1
$max\_arch$	10
$max\_contr\_ratio$	5

TABLE II  
 $fmincon$  SETTINGS

$max\_eval$	100
$tol\_con$	1e-6

where  $g$  is the gravity acceleration,  $a$  the thrust acceleration,  $x$  and  $y$  are the components of the position vector,  $v_x$  and  $v_y$  the components of the velocity vector and  $u$  the control. The dynamics is integrated from time  $t = 0$  to time  $t = t_f$ . The boundary conditions are:

$$\begin{cases} x(0) = 0; & v_x(0) = 0 \\ y(0) = 0; & v_y(0) = 0 \\ y(t_f) = h; & v_y(t_f) = 0 \end{cases} \quad (18)$$

The parameters  $g$ ,  $a$  and  $h$  were respectively set to  $1.6 \cdot 10^{-3}$ ,  $4 \cdot 10^{-3}$  and 10. Following [6], the DFET method was applied splitting the time domain into 4 elements, with polynomials of order 6 for each control and state variable. The control angle was bounded between  $-\frac{\pi}{2}$  and  $\frac{\pi}{2}$ , while total mission time was bounded between 100 and 250. This gives a total of 29 optimisation variables. Table I summarises the settings of the optimiser:  $max\_fun\_eval$  the maximum number of objective functions evaluation,  $pop\_size$  the number of agents performing the search,  $\rho\_ini$  the initial radius of the local neighbourhood,  $F$  and  $CR$  the standard parameters for the Differential Evolution social actions,  $p\_social$  the ratio between agents performing only social actions and the total number of agents,  $max\_arch$  the number of solutions to be stored in  $A_g$ ,  $contr\_ratio$  contraction rate of the neighbourhood radius, and  $max\_contr\_ratio$  the maximum number of times  $\rho_j$  can contract before it is reset. Settings reported in Table II instead refer to the parameters of  $fmincon$ :  $max\_con\_eval$  is the maximum number of constraints evaluation (for each call to the objective functions) and  $tol\_con$  is the threshold under which the solution is considered to be feasible. All other  $fmincon$  settings are left as default.

Algorithm 1 was run 30 times to collect some statistics on its convergence behaviour (see Table III). The Generational Distance (GD) and Inverse Generational Distance (IGD) were used as accuracy metrics and were computed on a rescaled front in the interval  $[0, 1]$ . For the Goddard's rocket problem, GD and IGD were computed using the analytical solution of the minimum time problem for different maximum  $v_x$  while for the Orbit Rise case the cumulative front derived from 30 runs plus the maximum energy solutions for different times of

TABLE III  
CONVERGENCE AND SPREADING STATISTICS FOR THE TWO PROBLEMS

Problem	mean GD (variance)	mean IGD (variance)
Goddard	2.833e-2 (1.4232e-5)	2.9449e-2 (1.5498e-5)
Orbit	5.9444e-2 (1.96243e-4)	4.387e-2 (1.6173e-4)

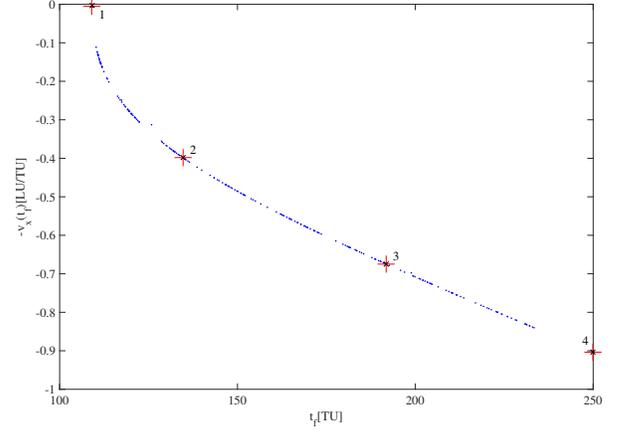


Fig. 1. Non dominated solutions of 30 different runs for the Goddard problem. Crosses indicate solutions for which trajectories, velocities and control law over time are also plotted. Circles indicate the objective values corresponding to the analytic solutions with the same time as the solutions marked with crosses

transfer was used as the reference front. Figure 1 shows the cumulative front from all 30 runs, along with 4 representative solutions (marked with crosses) and the analytic solutions with the same ascent time of the representative solutions (marked with circles). The crosses and circles are perfectly overlapping. The trajectories and time histories of the controls and velocities for the 4 representative solutions are plotted in figures 2 and 3 to 6 along with the single objective numerical solution and the analytic solution for the same ascent times. The solution obtained with the proposed approach is very close to both the numerical single objective and the analytic solutions. The discontinuities in the control laws are due to the discretisation scheme and to tolerance on the optimality of the solutions.

### B. Maximum Energy Orbit Rise

The original maximum energy orbit rise formulation and some solution strategies can be found in [14] and [6]. In this case, a spacecraft is orbiting around a celestial body, and it is required to increase its total energy by changing its altitude and velocity. The only control variable is again the thrusting angle, and the only other force affecting the spacecraft is gravity (in this case it is considered variable with altitude). The multi-objective extension, proposed in this paper, maximises the final energy and minimises the manoeuvre time:

$$\min_{t_f, u} \left( J_1 = t_f, J_2 = -\frac{(v_r^2(t_f) + v_t^2(t_f))}{2} + \frac{1}{r(t_f)} \right) \quad (19)$$

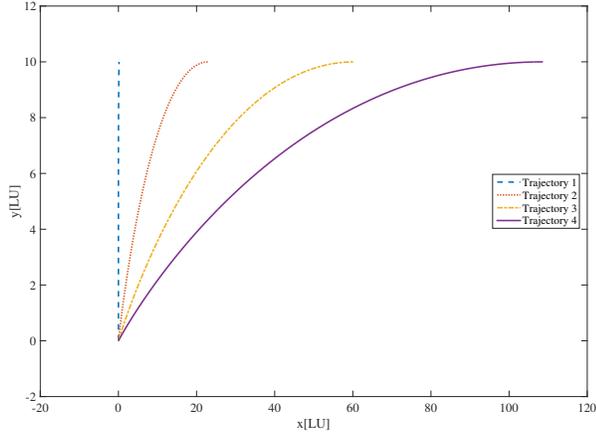
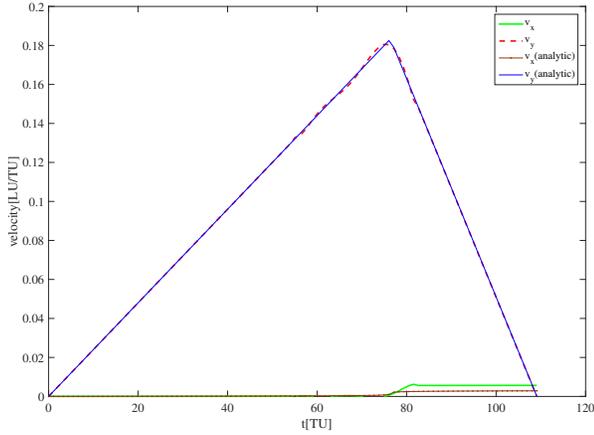
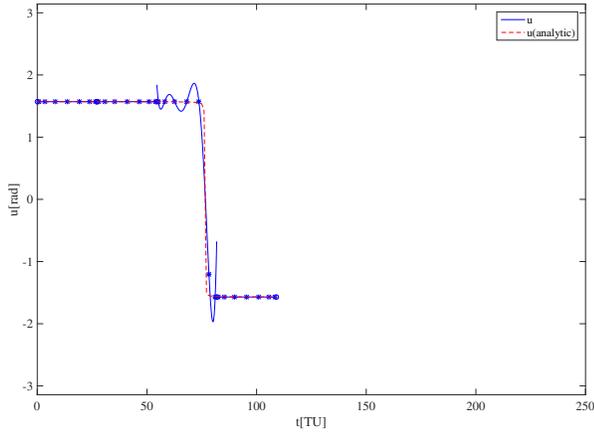


Fig. 2. Trajectories corresponding to the 4 selected points on the Pareto front



(a) Time history for the velocities

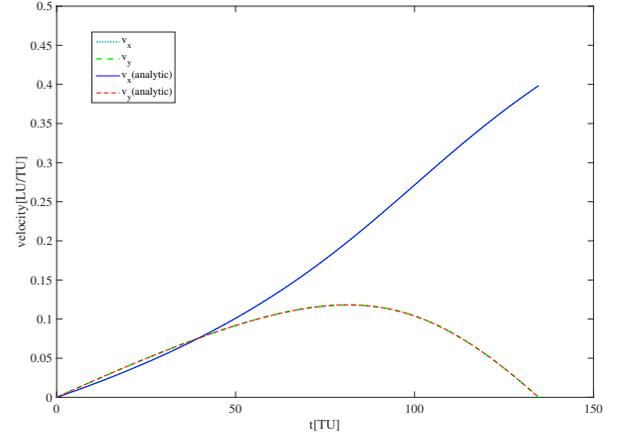


(b) Time history for the controls

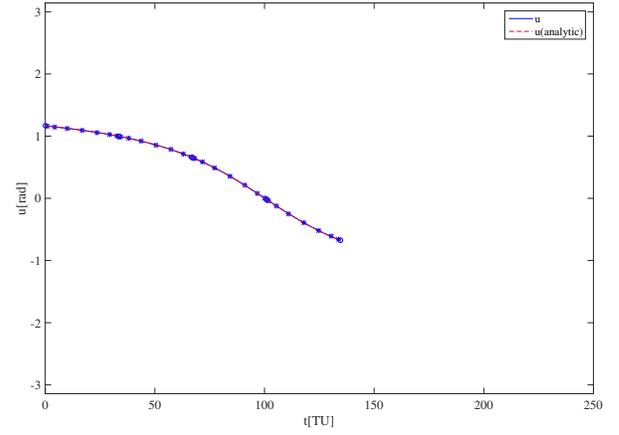
Fig. 3. Time history for velocities and controls, point 1 on the Pareto front

subject to the dynamic constraints:

$$\begin{cases} \dot{r} &= v_r \\ \dot{v}_r &= \frac{v_t^2}{r} - \frac{1}{r^2} + a \cos u \\ \dot{\theta} &= \frac{v_t}{r} \\ \dot{v}_t &= -\frac{v_t v_r}{r} + a \sin u \end{cases} \quad (20)$$



(a) Time history of the velocities



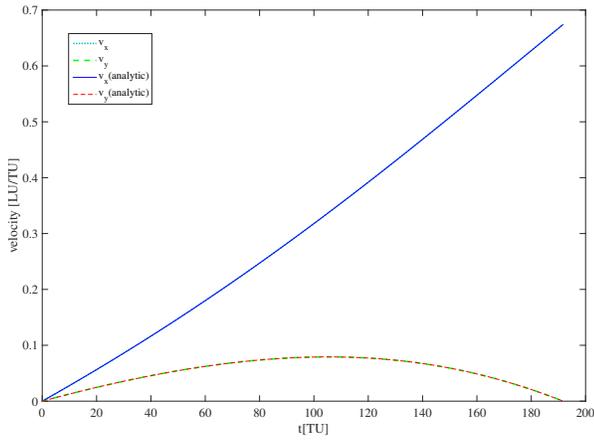
(b) Time history of the controls

Fig. 4. Time history of velocities and controls, point 2 on the Pareto front

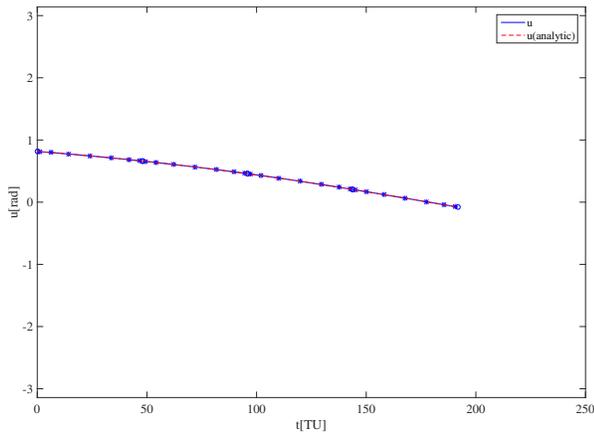
where  $r$  and  $\theta$  are the polar coordinates of the spacecraft,  $v_r$  and  $v_t$  are the radial and transversal velocities, and  $a$  is the magnitude of the control acceleration. In this work,  $a = 1e-2$ . The boundary conditions are:

$$\begin{cases} r(0) = 1.1; & v_r(0) = 0 \\ \theta(0) = 0; & v_t(0) = \frac{1}{\sqrt{1.1}} \end{cases} \quad (21)$$

Following [6], the time domain was subdivided into 30 elements of order 1 for each state and control variable. Control angles were bound between  $-\pi$  and  $\pi$ , while total mission time was bounded between 20 and 80. In total there are 61 optimisation variables. The transcribed problem was then optimised with MACSoc, with the same settings as in the previous case. The non dominated Pareto front of the combined 30 runs is reported in figure 7, while the GD and IGD are reported in Table III. In this case, we plot the comparison of 4 trajectories, and the corresponding control laws in figures 8 and 9 to 12 where one can see that the MACSoc control laws (dashed line in figures 9 to 12) closely follows the analytical solution for each  $t_f$  (continuous line in the same figures).

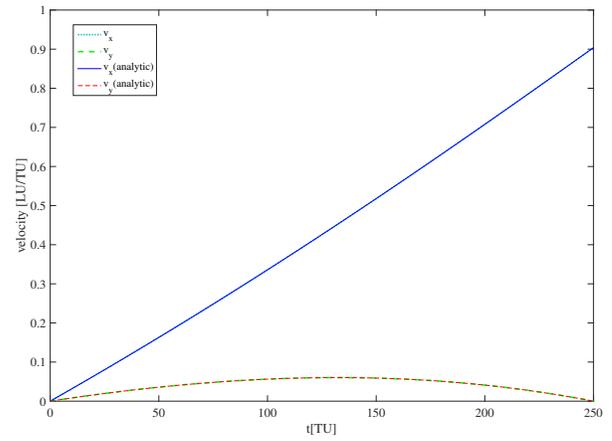


(a) Time history of the velocities

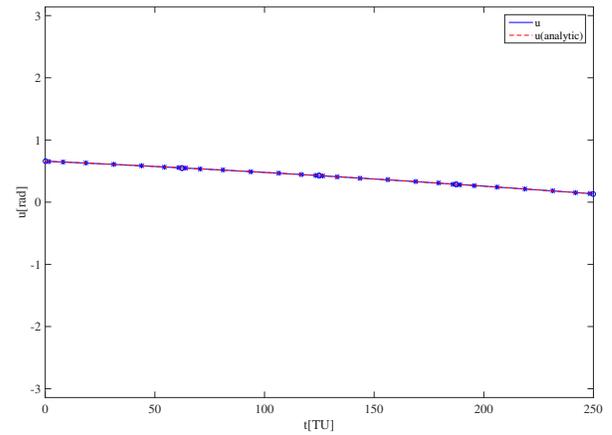


(b) Time history of the controls

Fig. 5. Time history of velocities and controls, point 3 on the Pareto front



(a) Time history of the velocities



(b) Time history of the controls

Fig. 6. Time history of velocities and controls, point 4 on the Pareto front

## VI. CONCLUSION

The paper proposed a novel memetic approach for the solution of multi-objective optimal control problems called MACSoc. The results on two standard optimal control problems with known control laws demonstrated that MACSoc can reliably converge to the Pareto front with good accuracy, good spreading of the solutions and a relatively low number of function evaluations. Further work is required to improve the treatment of infeasible solutions. The simple rejection mechanism is effective in the cases treated in this paper but can prevent progression towards the Pareto front in more complex problems with more difficult constraints.

## ACKNOWLEDGMENT

The first author gratefully acknowledges the support of the ESA NPI grant ref. TEC-ECN-SoW-20140806 and of Airbus Defence and Space.

## REFERENCES

- [1] V. Coverstone-Carroll, J. W. Hartmann and W. J. Mason, *Optimal multi-objective low-thrust spacecraft trajectories*, Computer methods in applied mechanics and engineering, volume 186, number 2, 2000, pp 387–402

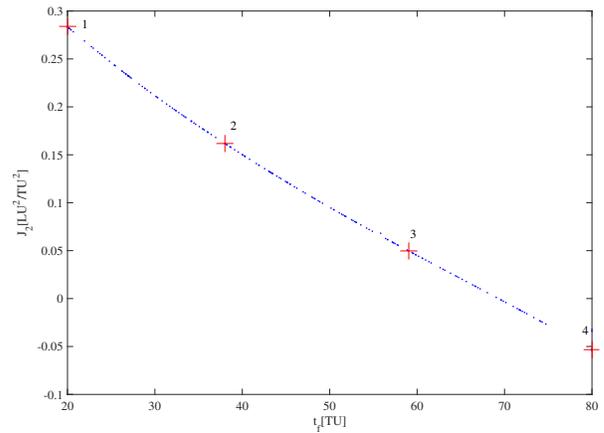


Fig. 7. Non dominated solutions of 30 different runs for the maximum energy rise problem. Crosses indicate four representative solutions along the Pareto front.

- [2] S. Ober-Blobaum, M. Ringkamp and G. zum Felde, *Solving multi-objective optimal control problems in space mission design using discrete mechanics and reference point techniques*, IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp 5711–5716
- [3] C.cn Kaya, H. Maurer, *A Numerical Method for Nonconvex Multi-objective Optimal Control Problems*, Computational Optimization and Applications, 2013.

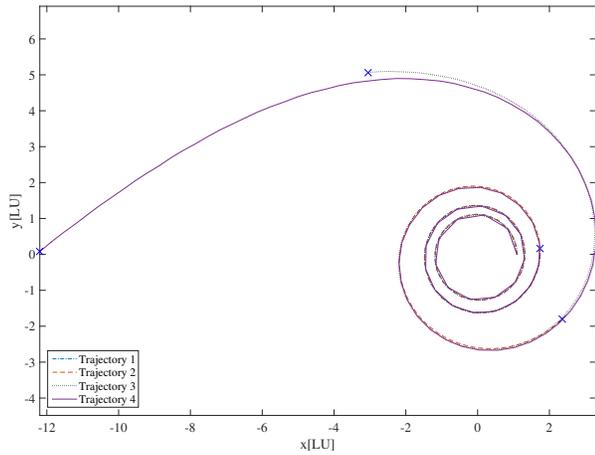


Fig. 8. Trajectories corresponding to the 4 selected points on the Pareto front

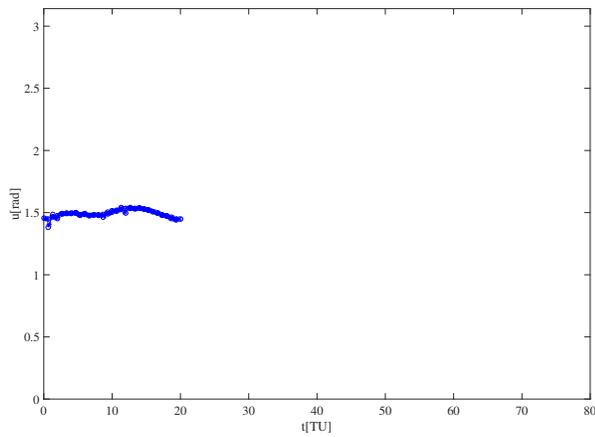


Fig. 9. Time history of the control angle for point 1 on the Pareto front

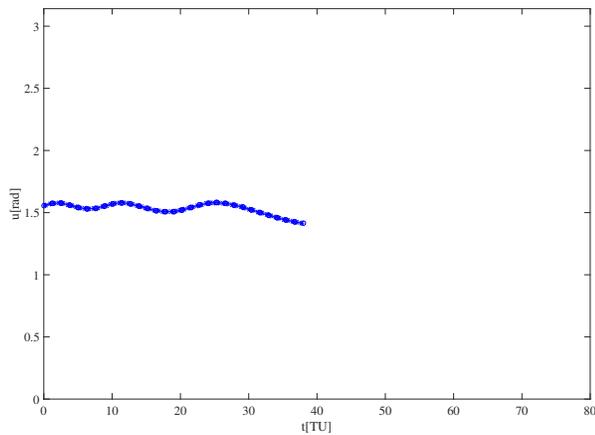


Fig. 10. Time history of the control angle point 2 on the Pareto front

- [4] J. A. Englander, M. A. Vavrina, A. R. Ghosh, *Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design*. AAS 15-227, 25th AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA.
- [5] Zuiani F. and Vasile M., *Multi Agent Collaborative Search Based on Tchebycheff Decomposition*, Computational Optimization and Applications September 2013, Volume 56, Issue 1, pp 189-208, DOI 10.1007/s10589-013-9552-9.

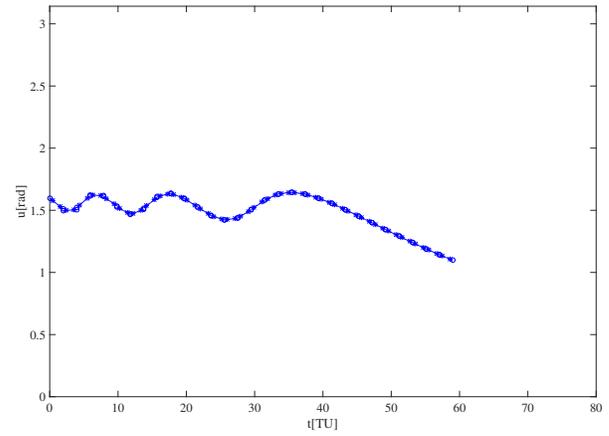


Fig. 11. Time history of the control angle for point 3 on the Pareto front

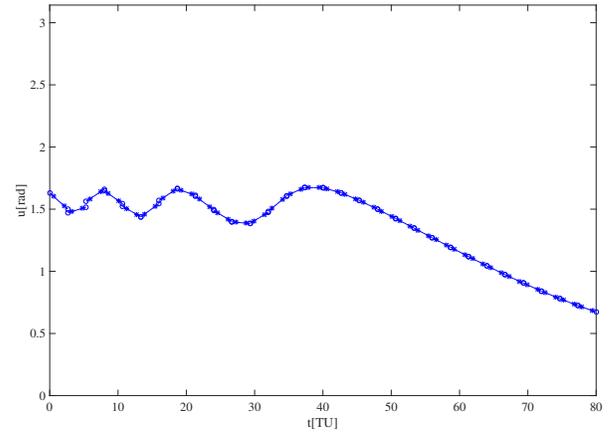


Fig. 12. Time history of the control angle for point 4 on the Pareto front

- [6] M. Vasile, *Finite elements in time: a direct transcription method for optimal control problems*, AIAA 2010-8275, AIAA/AAS Astrodynamics Specialist Conference, 2 - 5 August 2010, Toronto, Ontario Canada
- [7] M. Vasile and A. E. Finzi, *Direct lunar descent optimisation by finite elements in time approach*, International Journal of Mechanics and Control, volume 1, number 1, 2000.
- [8] H. D. Hodges and R. R. Bless, *Weak Hamiltonian finite element method for optimal control problems*, Journal of Guidance, Control, and Dynamics, volume 14, number 1, 1991, pp 148-156.
- [9] C. L. Bottasso and A. Ragazzi, *Finite element and Runge-Kutta methods for boundary-value and optimal control problems*, Journal of Guidance, Control, and Dynamics, volume 23, number 4, 2000, pp 749-751.
- [10] M. Vasile and F. Bernelli-Zazzera, *Optimizing low-thrust and gravity assist manoeuvres to design interplanetary trajectories*, The Journal of the astronomical sciences, volume 51, number 1, 2003, pp 13-35.
- [11] M. Vasile and F. Bernelli-Zazzera, *Targeting a heliocentric orbit combining low-thrust propulsion and gravity assist manoeuvres*, Operational Research in Space & Air, volume 79, 2003.
- [12] L. A. Ricciardi and M. Vasile, *Improved archiving and search strategies for Multi Agent Collaborative Search*, Eurogen 2015
- [13] A. E. Bryson, Y. Ho, *Applied optimal control: optimization, estimation and control, Revised printing*, CRC Press, 1975.
- [14] A. L. Herman and B. A. Conway, *Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules*, Journal of Guidance, Control, and Dynamics, volume 19, number 3, 1996, pp 592-599.