

# A Direct Multiple Shooting Approach to the Solution of Optimal Control Problems with Time-Delays\*

Carlo L. Bottasso<sup>1</sup>, Luca Riviello<sup>1</sup>, Massimo Ruzzene<sup>2</sup>, Francesco Scorcelletti<sup>1</sup>

<sup>1</sup> Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy. Phone: +39 02 2399 8315; Fax: +39 02 2399 8334; e-mail: [carlo.bottasso@polimi.it](mailto:carlo.bottasso@polimi.it)

<sup>2</sup> School of Aerospace Engineering, Georgia Institute of Technology, 270, Ferst Drive, 30332-0150 Atlanta, Georgia, USA

**Abstract** A method is presented for solving optimal control problems for dynamical systems governed by delay differential equations. The optimal control problem is transformed by the direct approach into a non-linear programming problem. Temporal discretization is performed by means of a multiple shooting method, which takes explicitly into account the dependence of the dynamic equations on the time delay. In particular, state continuity between two neighboring arcs is enforced on an overlapping time range using local shape functions. The effectiveness of the approach is demonstrated by a set of numerical examples, and by its application to the problem of trajectory optimization of supercavitating underwater vehicles.

**Key words** Optimal control, multiple shooting, delay differential equations, flight mechanics, trajectory optimization, supercavitation

## 1 Introduction

In modeling physical phenomena, it is generally assumed that all events are caused by earlier ones, this property being referred to as *causality*. To a very large extent, most mathematical models rely on the further assumption that the future state of a system is determined solely by its present state. In this case, the system response does not depend on its past history and the model is typically described by Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs). On the other hand, when the

---

\* Paper submitted to the *Journal of Optimization Theory and Applications*.

system dynamic response depends on its past history, time delays emerge in the modeling process and the behavior of the system is governed by Delay Differential Equations (DDEs) or Functional Differential Equations (FDEs) [9,10].

The reasons for including delay effects in a mathematical model can be probably categorized in at least two broad families. First, time delays frequently describe specific features of the physical phenomenon: typical examples of this kind can be found, for example, in population dynamics (reforestation, parasite growth and decay, etc.), where the rate of increase of the individuals depends on preceding events (e.g., the cut of the forest) or on the past dimension of the population (which, for example, determines the exposure to a parasite). In other cases, DDEs arise from the reduction of models governed by PDEs or ODEs through suitable substitution operations (see, for example, the simple example on lossless transmission lines in Reference [9, pp. 8–9]).

Similar models are sometimes used in fluid mechanics applications. As an example in the field of aeronautical engineering, consider the case of an aircraft with wing and aft horizontal stabilizer. A change in the circulation of the wing will shed vorticity in the wake which, traveling downstream at some finite velocity, will affect the angle of attack and hence the lift on the horizontal stabilizer with some time delay. Clearly, this effect can be modeled directly by using PDEs; for example, we may use the Navier-Stokes equations which, after spatial discretization, are turned into a large set of ODEs which is then integrated numerically in time. However, we can also model the same phenomenon using DDEs by considering a time delay between the change in lift on the wing and the variation in the relative velocity at the tail. In a general sense, the resulting DDE model can be thought of as a reduction of the former ODE one: we can imagine that all spatial fluid dynamic degrees of freedom are solved for in terms of the past history of the state, and eliminated from the system. This way the aerodynamic forces (in particular, those acting on the tail) now depend on the past, making the new model governed by a DDE instead of an ODE.

In this work we present a procedure to numerically solve optimal control problems for dynamical systems governed by DDEs. The interest in systems with delay stems here from the study of optimal maneuvers for a class of unconventional mechanical systems: supercavitating underwater vehicles. These vehicles possess a substantially reduced hydrodynamic drag with respect to fully wetted ones, being in contact with water only (i) at the nose, where a “cavitator” generates a cavity that completely envelops the body, (ii) at the fins, which pierce the cavity and are used to control the system, and (iii) on the vehicle after-body [12]. This unique configuration has led to a number of studies on the flight mechanics characteristics of this class of vehicles, for example for synthesizing suitable control laws [8], or analyzing the vehicle behavior during maneuvers [4]. Clearly, possible complex vehicle–cavity interactions make the problem of modeling supercavitating vehicles in unsteady conditions particularly challenging. Moreover, memory

effects associated with the advection of disturbances downstream from the cavitator are often mathematically modeled using DDEs [12].

The specialized literature proposes several techniques for solving optimal control problems for ODEs [3]. However, the DDE case has received very limited attention, and only for special simplified (e.g., linear) cases [6]. It appears that no well established solution method is currently available to optimize the motion of a generic, non-linear system with delay. In this work, we propose a general technique for solving optimal control problems for DDE models of arbitrary complexity. More precisely, we use a direct approach to recast the original optimal control problem into a non linear programming (NLP) one. In fact, the direct approach to optimal control consists in first discretizing the system states, controls and dynamic equations, and then optimizing the resulting algebraic problem. This procedure readily allows for the solution of the optimal control problem using off-the-shelf optimization tools [2]. This is in contrast with the indirect approach, where optimality is first explicitly enforced at the level of the infinite dimensional problem through the Lagrange multiplier technique, followed by discretization of the resulting boundary value problem [3]. The indirect approach may lead to extremely complex derivations even for ODE models, and would be a daunting task for most DDE cases except the simplest academic examples.

In this work, discretization of the problem is handled using a multiple shooting technique. This choice enables the treatment of problems characterized by very fast dynamic components in the model response compared to the overall analysis time span, which would otherwise not be manageable with direct-transcription-type methods [3]. In conventional multiple shooting methods for ODEs, the tail of each arc is glued to the head of the next one by enforcing point continuity conditions; such constraints are here substituted by continuity conditions on a line, given the fact that DDEs require line initial conditions instead of the usual point ones of ODEs. To make the problem tractable at the discretized level, shape functions are employed locally to represent the solution over the gluing line, yielding a finite number of constraints to render the continuity condition.

The remainder of the paper is organized as follows. First, Section 2 presents the relevant notation by describing time-delayed models and optimal control problems for this class of systems. Next, Section 3 describes in detail the proposed direct multiple shooting technique. Section 4 first presents the validation of the proposed algorithm on two simple examples, and subsequently illustrates the applicability of the method to a problem of engineering relevance involving the trajectory optimization of supercavitating underwater vehicles. Finally, Section 5 summarizes the main results of the work and reports on the conclusions of the present study.

## 2 Optimal Control Problems for Time-Delayed Systems

### 2.1 Time-Delayed Dynamical Systems

In this work, we consider dynamical systems governed by the following set of delay differential equations:

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \mathbf{y}(t - \tau_1), \dots, \mathbf{y}(t - \tau_M), \mathbf{u}(t)), \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^{n_y}$  and  $\mathbf{u} \in \mathbb{R}^{n_u}$  denote the state and control vectors, respectively, while function  $\mathbf{f} : \mathbb{R}^{n_y} \times \mathbb{R}^{M n_y} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$  depends upon current and delayed states. In equation (1), this dependency is represented through the positive time delays  $\tau_i \in \mathbb{R}_+$ ,  $i = 1, \dots, M$ , which may be time-invariant or may change over time. We assume that, at each time instant  $t$ , the  $M$  delays can be implicitly defined by the following set of  $M$  non-linear non-singular algebraic equations

$$\mathbf{g}(\mathbf{y}(t), \mathbf{y}(t - \tau_1), \dots, \mathbf{y}(t - \tau_M)) = 0, \quad (2)$$

where  $\mathbf{g} : \mathbb{R}^{n_y} \times \mathbb{R}^{M n_y} \rightarrow \mathbb{R}^M$ . For future reference, we define an upper bound  $\tau$  for the time delays as

$$\tau = \max_{t \in \Omega} \{\tau_1, \dots, \tau_M\} \geq 0, \quad (3)$$

where  $\Omega = [T_0, T] \subset \mathbb{R}$  is the time domain.

The integration in time of equation (1) over  $\Omega$  clearly requires the knowledge of the inputs applied to the system over the time interval, for example

$$\mathbf{u}(t) = \mathbf{u}^*(t), \quad t \in \Omega, \quad (4)$$

where  $\mathbf{u}^*(t)$  is some assigned control time history. Moreover, for a unique solution to exist [9], a *line* initial condition is necessary, namely

$$\mathbf{y}(T_0 - \eta) = \mathbf{y}_0(\eta), \quad \eta \in [0, \tau], \quad (5)$$

where  $\mathbf{y}_0 : [0, \tau] \rightarrow \mathbb{R}^{n_y}$  is a function providing the required information on the states prior to time instant  $T_0$ .

The ODE case is readily recovered by considering equation (1) with  $\tau_i = 0$ ,  $i = 1, \dots, M$ . From equation (5), the initial conditions required to solve an initial value problem (IVP) for a system of ODEs are

$$\mathbf{y}(T_0) = \mathbf{y}_0, \quad (6)$$

with  $\mathbf{y}_0 \in \mathbb{R}^{n_y}$ . In other words, a *point* initial condition must be specified when dealing with ODEs, in contrast with the *line* condition of the general DDE case. This is coherent with the distinction made in Section 1 between ODEs and DDEs, regarding the assumption of causality: for a DDE model, future states depend on a portion of the past system history, while for an ODE model they depend only on the current system state. Section 3.2 will specifically address the formulation of suitable state continuity conditions which are directly related to the line initial conditions (5).

## 2.2 Optimal Control of Time-Delayed Systems

We consider the problem of determining the optimal time histories of the control inputs  $\mathbf{u}_{\text{opt}}(t)$  and the associated states  $\mathbf{y}_{\text{opt}}(t)$  which minimize an assigned functional, subjected to a given set of constraints. The cost function can be written in general as

$$J = \phi^b(\mathbf{y}, \mathbf{u}, t)|_{\Gamma} + \int_{\Omega} L(\mathbf{y}, \mathbf{u}, t) dt, \quad (7)$$

where the first term accounts for possible initial and/or terminal costs, with  $\Gamma = \{T_0, T\}$  denoting the boundary of the domain  $\Omega$ , while the second term is the integral cost.

The optimal solution must satisfy a number of constraints, and in particular the system dynamic equations

$$\dot{\mathbf{y}}(t) - \mathbf{f}(\mathbf{y}(t), \mathbf{y}(t - \tau_1), \dots, \mathbf{y}(t - \tau_M), \mathbf{u}(t)) = 0, \quad (8a)$$

$$\mathbf{g}(\mathbf{y}(t), \mathbf{y}(t - \tau_1), \dots, \mathbf{y}(t - \tau_M)) = 0, \quad (8b)$$

possible initial and/or final line conditions

$$\mathbf{y}(T_0 - \eta) \in [\mathbf{y}_0^{\min}(\eta), \mathbf{y}_0^{\max}(\eta)], \quad (9a)$$

$$\mathbf{y}(T - \eta) \in [\mathbf{y}_T^{\min}(\eta), \mathbf{y}_T^{\max}(\eta)], \quad (9b)$$

with  $\eta \in [0, \tau]$ , non-linear constraints on states and controls

$$\mathbf{z}(\mathbf{y}, \mathbf{u}, t) \in [\mathbf{z}^{\min}, \mathbf{z}^{\max}], \quad (10)$$

integral conditions

$$\int_{\Omega} \mathbf{h}(\mathbf{y}, \mathbf{u}, t) dt \in [\mathbf{h}^{\min}, \mathbf{h}^{\max}], \quad (11)$$

and lower and upper bounds on states and controls

$$\mathbf{y} \in [\mathbf{y}^{\min}, \mathbf{y}^{\max}], \quad (12a)$$

$$\mathbf{u} \in [\mathbf{u}^{\min}, \mathbf{u}^{\max}]. \quad (12b)$$

For generality, conditions (9-12) are expressed as inequality constraints in the form  $x \in [x^{\min}, x^{\max}]$ , i.e.  $x^{\min} \leq x \leq x^{\max}$ . Equality constraints are readily enforced by letting  $x^{\min} = x^{\max}$ .

### 3 Numerical Solution of Optimal Control Problems for Time-Delayed Systems

#### 3.1 Overview

For the solution of generic, non-linear optimal control problems, there are two main different approaches [3]. The *indirect approach* amounts to augmenting the cost (7), by adjoining the governing equations (8) and all other constraint conditions (9–12) with a set of Lagrange multipliers. The first order stationarity conditions for the augmented cost yield a new set of equations with their associated boundary conditions, which govern the optimal control problem [5]. Whenever the problem can not be solved analytically, the resulting boundary value problem is discretized introducing a time grid and using some suitable numerical scheme.

On the other hand, in the *direct approach*, instead of first optimizing and then discretizing, one first discretizes the system states and controls and the model equations by means of a numerical method, and then the resulting algebraic problem is optimized. Cost function and constraints are readily evaluated in terms of the discrete variables by using appropriate interpolation and quadrature rules. This discretization process defines a discrete parameter optimization problem [7], which can be written in general as

$$\min_{\mathbf{x}} K(\mathbf{x}), \quad (13a)$$

$$\text{s.t.: } \phi(\mathbf{x}) \in [\phi^{\min}, \phi^{\max}]. \quad (13b)$$

The vector of optimization parameters is noted  $\mathbf{x}$ ,  $K(\mathbf{x})$  is the discrete version of the cost functional  $J$  given in equation (7), and  $\phi(\mathbf{x})$  is a vector function gathering together all the problem equality and inequality constraints. The specific form of the vector of algebraic unknowns  $\mathbf{x}$  and of the constraints  $\phi(\mathbf{x})$  depends on the method used for performing the discretization.

Similarly to the indirect approach, necessary conditions for a constrained optimum for problem (13) can now be obtained by adjoining the constraints  $\phi(\mathbf{x})$  to the objective  $K$  through the use of Lagrange multipliers, and imposing the stationarity of the resulting augmented objective function. When dealing with complex non-linear systems, the formal derivation of the optimal control governing equations, which is necessary in the indirect case, can be a nontrivial task, especially if time delays are present. On the contrary, the direct approach does not require the manipulation of the system equations of motion, and all that is necessary is the evaluation of the discretized cost function and constraints.

In the specialized literature, two rather different direct techniques emerge as the most commonly employed ones, namely *direct shooting* and *direct transcription*.

In the direct *single shooting* method, the vector of NLP unknowns typically contains the initial and the final system states, together with degrees

of freedom which describe the control variables. Given a guess of the initial conditions and of the controls, one can integrate the system equations forward in time, evaluating both cost function and constraints. This procedure has the effect of eliminating the internal states from the problem, except for the state values at the initial and terminal boundaries, which are retained as unknowns so as to be able to constrain them explicitly in order to enforce the boundary conditions. The direct *multiple* shooting technique simply consists in partitioning the time domain into several segments, applying single shooting to each one of them and enforcing suitable gluing conditions on states and controls at each interface between two segments. To this end, additional state unknowns at each segment interface are introduced in the optimization variables.

Direct transcription, on the other hand, is based on the idea of discretizing the equations of motion on each single time step; these discrete equations are then appended as constraints to the optimization problem. The vector of NLP variables is in this case represented by the values of states and controls on the discretization grid. This way the problem is much larger than in the multiple shooting case, but very sparse and typically with greater stability and robustness; furthermore, the problem is characterized by having many constraints which are active at convergence.

In spite of these two conceptually different ways of generating the final NLP problem, both direct multiple shooting and direct transcription reduce the original problem to a discrete parameter optimization problem. The choice of one method over the other, more often than not, is made based on the specific features of the application being dealt with. Both methods are commonly used for dynamic systems governed by ODEs, although they can be conceptually extended to cases in which delays are present.

In view of the applications considered in this work, it is worth noticing that with multiple shooting the size of the resulting NLP problem does not depend on the temporal resolution which is necessary so as to integrate the equations of motion while satisfying stability and accuracy requirements; in fact, in the case of shooting the choice of the time step does not affect the number of optimization variables. On the contrary, with the direct transcription technique, the number of NLP unknowns grows with the number of time steps taken to cover the temporal domain. In this work, we consider typical delays  $\tau$  which can be very significantly smaller than the simulation time span  $\Delta T = T - T_0$ . Furthermore, for accuracy and stability reasons, time marching of the equations of motion is typically conducted with time steps  $h$  which are much smaller than the typical delay  $\tau$ . Therefore, the numerical solution of the optimal control problems considered here is such that

$$h \ll \tau \ll \Delta T,$$

which potentially leads to a very large number of integration/discretization steps. In practice, the use of direct transcription would lead to extremely large NLP problems with prohibitive computational costs. This discussion

motivates the selection of the direct multiple shooting technique as the method of choice for the present study.

### 3.2 Multiple Shooting for Delay Differential Equations

We consider a partition of the time domain  $\Omega$  given by  $0 = t_0 < t_1 < \dots < t_{N_a} = T$  with  $\Omega^i = [t_i, t_{i+1}]$ ,  $i = 0, \dots, N_a - 1$ , where each  $\Omega^i$  is a shooting segment. Here and in the following, quantities associated with the generic vertex between segments  $i$  are indicated using the subscript  $(\cdot)_i$ , while quantities associated with the generic segment  $j$  are labeled with the superscript  $(\cdot)^j$ . On each shooting segment  $\Omega^i$ , the control vector is discretized as  $\mathbf{u}^i(t) = \sum_{j=1}^{N_c^i} s_j(t) \mathbf{u}_j^i$ , where  $s_j(t)$  are basis functions, in particular cubic splines in this work, and  $\mathbf{u}_j^i$  are  $N_c^i$  discrete control vectors. The present implementation allows for a different number of control degrees of freedom (DOFs) on each segment  $i$ , as indicated by the notation  $N_c^i$ .

Constraints need to be enforced for the state variables between arcs. Since we are dealing with DDEs, these gluing conditions are line constraints covering a time interval whose length is equal to the delay. For this purpose, in this work we use local expansions to approximate the state variables over the overlap intervals  $[t_i - \tau, t_i]$ . Notice that the overlap is defined in terms of the maximum delay  $\tau$  (equation (3)). Although implementations considering different overlaps for the different state variables based on their associated delays are certainly possible, here the marginal efficiency gains were not considered worth the considerable extra complexity of the implementation.

The local interpolations in the overlap regions can be expressed as follows:

$$\mathbf{y}_i(\xi) = \mathbf{y}(t_i - \tau\xi) = \sum_{k=1}^K \Phi_k(\xi) \mathbf{a}_{ik}, \quad i = 0, \dots, N_a, \quad \xi \in [0, 1], \quad (14)$$

where  $\mathbf{y}_i$  indicates a (state) vector function associated with the generic vertex  $i$  between two segments,  $\xi = \eta/\tau$  defines a dimensionless delay,  $\mathbf{a}_{ik} \in \mathbb{R}^{n_y}$  are interpolation coefficient vectors, while  $\Phi_k(\xi)$ ,  $k = 1, \dots, K$  are the interpolation basis functions. It is convenient to rewrite these relationships in the following compact vectorial form:

$$\mathbf{y}_i(\xi) = \mathbf{A}_i \mathbf{n}(\xi), \quad i = 0, \dots, N_a, \quad \xi \in [0, 1], \quad (15)$$

where  $\mathbf{A}_i \in \mathbb{R}^{n_y \times K}$  is the  $i$ th interpolation coefficient matrix defined as

$$\mathbf{A}_i = [\mathbf{a}_{i1} | \mathbf{a}_{i2} | \dots | \mathbf{a}_{iK}], \quad (16)$$

and  $\mathbf{n}(\xi) \in \mathbb{R}^K$  is the vector containing all basis functions:

$$\mathbf{n}(\xi) = (\Phi_1(\xi), \Phi_2(\xi), \dots, \Phi_K(\xi))^T. \quad (17)$$

In the present formulation, the optimization variables of the NLP problem are represented by the coefficients of the matrices  $\mathbf{A}_i$  at each gluing

line interface between two shooting segments, the control DOFs  $\mathbf{u}_j^i$  on each shooting segment, and possibly the final time  $T$  when this is unknown, i.e.

$$\mathbf{x} = \left( \dots, \mathbf{a}_{ik}^T, \dots, \mathbf{u}_j^{iT}, \dots, T \right)^T, \quad i = 0, \dots, N_a, \\ j = 1, \dots, N_c^i, k = 1, \dots, K. \quad (18)$$

The coefficients of matrix  $\mathbf{A}_{i-1}$  are used for approximating the initial line data for the integration over the  $i$ th arc, with  $i = 1, \dots, N_a$ . In other words, given matrix  $\mathbf{A}_{i-1}$ , the simulation time  $T$  and the control parameters  $\mathbf{u}_j^i$ , one can integrate numerically the equations of motion over the  $i$ th arc,  $i = 1, \dots, N_a$ , to obtain the system response along that arc.

Next, the tail of each arc needs to be glued with the head of the next one, in order to enforce the continuity of the states between subsequent integration arcs, as shown in Figure 1. The current estimate of the state time history along the  $i$ th arc  $[t_{i-1}, t_i]$ ,  $i = 1, \dots, N_a$ , is denoted here  $\widehat{\mathbf{y}}_i(t)$ , and is obtained by forward numerical integration on that arc. This time history overlaps with the interval  $[t_i - \tau, t_i]$ , which is the time range related to the initial data for the next arc  $i + 1$ . Therefore, the line continuity conditions between the two arcs can be expressed as

$$\widehat{\mathbf{y}}_i(t_i - \tau\xi) = \mathbf{y}_i(\xi) = \sum_{k=1}^K \Phi_k(\xi) \mathbf{a}_{ik}, \quad i = 1, \dots, N_a, \quad \xi \in [0, 1]. \quad (19)$$

Here again it is convenient to write equation (19) in compact form as follows:

$$\widehat{\mathbf{y}}_i(t_i - \tau\xi) = \mathbf{A}_i \mathbf{n}(\xi), \quad i = 1, \dots, N_a, \quad \xi \in [0, 1]. \quad (20)$$

The integration process is based on a numerical discretization scheme, and therefore, in the overlap region  $[t_i - \tau, t_i]$ , the states are known at the discrete time instants corresponding to the  $N_i$  integration mesh points, i.e

$$\widehat{\mathbf{y}}_i(t_i - \tau\xi_n), \quad n = 1, \dots, N_i, \quad i = 1, \dots, N_a. \quad (21)$$

Figure 2 shows a detail of the overlap region  $[t_i - \tau, t_i]$  and of the corresponding integration points. It should be noted that, in general,  $N_i$  may vary with the overlap region  $i$  considered, as indicated by the notation, especially if an adaptive integrator is used. If equations (20) are evaluated at the integration mesh points, they become

$$\widehat{\mathbf{y}}_i(t_i - \tau\xi_n) = \mathbf{A}_i \mathbf{n}(\xi_n), \quad n = 1, \dots, N_i, \quad i = 1, \dots, N_a, \quad (22)$$

or, in compact form

$$\widehat{\mathbf{Y}}_i = \mathbf{A}_i \mathbf{D}_i, \quad i = 1, \dots, N_a, \quad (23)$$

where the matrix of known coefficients  $\widehat{\mathbf{Y}}_i \in \mathbb{R}^{n_y \times N_i}$  is defined as

$$\widehat{\mathbf{Y}}_i = \left[ \widehat{\mathbf{y}}_i(t_i - \tau\xi_1) \mid \widehat{\mathbf{y}}_i(t_i - \tau\xi_2) \mid \dots \mid \widehat{\mathbf{y}}_i(t_i - \tau\xi_{N_i}) \right], \quad (24)$$

while  $\mathbf{D}_i \in \mathbb{R}^{K \times N_i}$  is the matrix of basis functions evaluated at the integration points

$$\mathbf{D}_i = [\mathbf{n}(\xi_1) | \mathbf{n}(\xi_2) | \dots | \mathbf{n}(\xi_{N_i})]. \quad (25)$$

Transposing equation (23), we obtain

$$\mathbf{D}_i^T \mathbf{A}_i^T = \widehat{\mathbf{Y}}_i^T, \quad i = 1, \dots, N_a, \quad (26)$$

which, for a given  $i$ , can be considered as a matrix linear system whose unknowns are the coefficients of matrix  $\mathbf{A}_i$ . In other words, the problem corresponds to a set of  $n_y$  linear systems, each having  $N_i$  equations and  $K$  unknowns. Since typically  $N_i \gg K$ , the linear system is over-constrained, and it can be readily solved through a least-squares approach. Pre-multiplication of equation (26) by  $\mathbf{D}_i$  gives

$$\mathbf{D}_i \mathbf{D}_i^T \mathbf{A}_i^T = \mathbf{D}_i \widehat{\mathbf{Y}}_i^T, \quad i = 1, \dots, N_a, \quad (27)$$

which enforces the state continuity condition along the overlap region  $[t_i - \tau, t_i]$ ,  $i = 1, \dots, N_a$  in a least-squares sense. Matrix system (27) is not explicitly solved in our implementation, rather it provides a set of constraints to the parametric optimization problem, i.e. it is included in the vector  $\boldsymbol{\phi}(\mathbf{x})$  of equation (13b).

It is worth noting that an alternative approach to enforcing continuity of the states would consist in imposing equality conditions at each integration point over the overlap region. In this case, the states at the integration nodes along the overlap regions  $\mathbf{y}_i(\xi_n)$  would have to be treated as optimization variables, instead of the interpolation coefficients  $\mathbf{a}_{ik}$ . The continuity conditions would be simply written as

$$\widehat{\mathbf{y}}_i(t_i - \tau \xi_n) = \mathbf{y}_i(\xi_n), \quad n = 1, \dots, N_i, \quad i = 1, \dots, N_a. \quad (28)$$

This approach leads to a simple formulation, which however suffers from two major drawbacks. First, these equality conditions require  $n_y \times N_i$  variables and constraints for each overlap region. On the contrary, the least-squares approach yields a number of local unknowns and constraints equal to  $n_y \times K$ . Since typically  $N_i \gg K$ , the point constraint approach implies a larger dimension of the resulting NLP problem. Second, the least-squares method is well suited when an adaptive time step integrator is employed, for which the number of integration points  $N_i$  in the overlap region is not known upfront. On the contrary, the dimension of the NLP problem obtained by imposing the point-wise equality conditions depends on the parameters  $N_i$ ,  $i = 1, \dots, N_a$ , which would need to be assigned *a priori*.

Finally, it is also useful to be able to enforce some degree of continuity on the controls across segment boundaries. In this case the gluing conditions are simply point constraints, as opposed to the line constraints of the state variables, which are easily expressed in terms of the control DOFs and included among the constraints of the NLP problem.

The discretization of the original optimal control problem with the method described in this section results in a non-linear discrete parameter optimization problem, formally described by equation (13). In particular, the vector function  $\phi(\mathbf{x})$  includes the state continuity conditions at the overlap regions expressed by relations (27), the continuity conditions on the controls at the segment boundaries, initial and final conditions and all constraints and bounds on states and control inputs. Note that constraints on the states can be explicitly enforced only in terms of the state optimization unknowns, which however are defined only at the interfaces between two segments. Internal state constraints on a segment need to be enforced through approximate ad-hoc procedures, whose treatment goes beyond the scope of this work.

NLP problem (13) can be solved efficiently by Sequential Quadratic Programming (SQP) methods [2], which is the approach considered in this work, or through the application of Interior Point (IP) algorithms [14].

## 4 Numerical Examples

In this section we test and demonstrate the proposed direct multiple shooting method by considering three numerical examples.

### 4.1 Single State System

The first example considers a DDE-governed optimal control problem whose analytical solution is described in [13]. The accuracy of the numerical solution can be thus evaluated through its comparison with the analytical one. Consider the following dynamical system characterized by a single constant delay:

$$\dot{y}(t) = ky(t - \tau) + u(t), \quad (29)$$

with  $k = 2$  and  $\tau = 1$ . Equation (29), although extremely simple, is a special case of the general class of systems given in equation (1). Referring to equation (29), let us consider an optimal control problem over the domain  $\Omega = [0, T]$ , consisting in the minimization of the cost function

$$J = \frac{1}{2}ay^2(T) + \frac{1}{2} \int_0^T u^2(t) dt, \quad (30)$$

with  $T = 2$  and  $a = 10^5$ , and subjected to the initial condition

$$y(0 - \eta) = 1, \quad \eta \in [0, 1]. \quad (31)$$

It can be shown [13] that the optimal control policy for this problem is

$$u_{\text{opt}}(t) = \begin{cases} -\frac{21}{16}(3 - 2t), & 0 \leq t < 1, \\ -\frac{21}{16}, & 1 \leq t \leq 2, \end{cases} \quad (32)$$

while the associated system response is

$$y_{\text{opt}}(t) = \begin{cases} 1 - \frac{31}{16}t + \frac{21}{16}t^2, & 0 \leq t < 1, \\ -\frac{25}{8} + \frac{115}{16}t - \frac{73}{16}t^2 + \frac{7}{8}t^3, & 1 \leq t \leq 2. \end{cases} \quad (33)$$

Considering the simplicity of the exact solution, it is straightforward to construct the NLP problem without introducing approximations. By using a linear interpolation of the control time history and an integration scheme which can exactly represent a cubic state response, no errors are introduced by the discretization process, and the exact solution can be obtained with the proposed numerical method, up to the tolerance used for arresting the iterations of the SQP algorithm [2]. Therefore we select linear control shape functions and  $N_u = 3$  parameters, and we use the integration scheme described in Reference [15], which achieves third-order accuracy by means of the local extrapolation technique [1]. Also, the time domain is broken into two equal shooting arcs.

Figures 3 and 4 represent the results of the numerical optimization problem in terms of state and control solutions, respectively. In Figure 3, the dotted line is the exact solution according to equation (33), circles represent the numerical solution, while solid lines indicate the overlapping regions used to enforce state continuity. For clarity, only a portion of the line initial condition to the left of  $t = 0$  is shown; the solid line between  $t = 0$  and  $t = 1$  indicates the overlap area between the first and second arc. Notice that there are no terminal boundary conditions in this problem, so no interpolation is needed on the tail of the second arc. The overlap area is treated using a third order polynomial to guarantee an exact agreement between the local expansion and the analytical solution. In Figure 4, the control time history is shown: here again, the dotted line represents the exact solution according to equation (32), while triangles indicate the three control parameters. For both states and controls there is perfect agreement between the analytical and numerical solutions. As explained before, this is due to the absence of approximations when constructing the discrete NLP problem. Indeed, possible differences between numerical and analytic results are only due to the tolerance used for arresting the SQP iterations, which however can be set to very strict values in this case.

#### 4.2 Spring-mass System

Goal of this second simple example is to show that the introduction of local expansions in the overlap areas between shooting segments does not affect the computed solution. We consider a simple, single DOF system composed of a mass and a non-conventional spring, which at time  $t$  generates a force proportional to the displacement at time  $t - \tau$ . The governing equation of the system can be written as follows:

$$m\ddot{x}(t) + kx(t - \tau) = \theta(t), \quad (34)$$

where  $x(t)$  describes the position of the mass, the function  $\theta(t)$  is an external force acting on the system as a control variable,  $m$  denotes the mass,  $k$  is the elastic constant and  $\tau$  the spring delay.

Equation (34) is readily transformed into a canonical system of first order equations in the general form (1) by writing

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{y}(t) + \begin{bmatrix} 0 & 0 \\ -k/m & 0 \end{bmatrix} \mathbf{y}(t - \tau) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \mathbf{u}(t), \quad (35)$$

where  $\mathbf{y}(t) = (x(t), v(t))^T$  and  $\mathbf{u}(t) = \theta(t)$ . The line initial conditions are

$$x(T_0 - \eta) = x_0(\eta), \quad \eta \in [0, \tau], \quad (36a)$$

$$v(T_0) = -\frac{dx_0}{d\eta}(0), \quad (36b)$$

while for the external forcing function we have

$$\theta(t) = \theta^*(t), \quad t \in \Omega, \quad (37)$$

$\theta^*(t)$  being a known control time history.

Figure 5 presents the results obtained by simulating the free response of the system for varying values of the delay  $\tau$ , and with initial conditions

$$x(T_0 - \eta) = 1, \quad \eta \in [0, \tau], \quad (38a)$$

$$v(T_0) = 0. \quad (38b)$$

The dash-dotted line, corresponding to  $\tau = 0$ , shows the simple harmonic oscillations characterizing the response of a classical spring-mass ODE system. On the other hand, it appears that the system is unstable in the presence of a delay  $\tau > 0$ . The amount of delay influences the instability of the system by increasing the rate of growth of the response amplitude. This unstable behavior can compromise the shooting optimization process; in general, when operating with unstable systems, an effective although somewhat heuristic solution is to break the temporal domain into several arcs by using multiple shooting.

The proposed technique is applied to the solution of an optimal control problem with cost function

$$J = \int_{T_0}^T \theta^2(t) dt + \frac{1}{\rho} \int_{T_0}^T \dot{\theta}^2(t) dt, \quad (39)$$

defined over a known duration  $\Delta T = T - T_0 = 3$ . In equation (39), the first integral term seeks to minimize the actuation force, while the second term is introduced to penalize excessive rates of the control input and to

avoid bang-bang solutions. The tuning parameter  $\rho$  is used to set the relative importance between the two integral terms. The initial and terminal boundary conditions are

$$x(T_0 - \eta) = 0, \quad \eta \in [0, \tau], \quad (40a)$$

$$v(T_0) = 0, \quad (40b)$$

$$x(T - \eta) = 1, \quad \eta \in [0, \tau], \quad (40c)$$

$$v(T) = 0. \quad (40d)$$

The system has unit mass and unit spring constant, and it is characterized by a delay  $\tau = 0.1$ . Also, the tuning parameter in equation (39) is set to  $\rho = 10$  in order to assign a greater weight to the force minimization objective.

The problem is solved for a number of control variables set to  $N_u = 40$ , and an increasing number of shooting arcs, namely  $N_a = 1, \dots, 4$ . Results are shown in Figures 6 and 7, which show that the computed solution does not depend on the number of arcs. This is clearly a desirable behavior, since multiple arcs should only be introduced to effectively deal with complex and unstable problems, but should not affect the accuracy of the numerical results. The plots of position and velocity time histories of Figures 6 and 7 also show the overlap areas used for the least-squares enforcement of state continuity, depicted using solid lines. These results were obtained by using local spline interpolations, while time marching was performed with the same Runge-Kutta integration scheme of the previous example. Finally, the time history of the force acting on the system is shown in Figure 8, illustrating how the force is always acting along the positive direction of motion, until settling on a constant unit value in order to compensate the elastic restoring effect of the spring.

#### 4.3 Trajectory Optimization of Supercavitating Underwater Vehicles

When a body moves through water at sufficient speed, the fluid pressure drops locally below the level that sustains the liquid phase, and a low-density gaseous cavity forms. Flows exhibiting cavities entirely enveloping the moving body, as shown in Figure 9, are termed *supercavitating* flows. In such regime, the liquid phase does not contact the moving body over most of its length, thus making the skin friction drag almost negligible. Specifically, in supercavitating conditions, the vehicle is in contact with the cavity just in a few localized parts: at the nose, where a specific device called the cavitator initiates the cavity formation, at the fins which are used for control, and on a portion of the vehicle after-body. Several new and projected supercavitating underwater vehicles exploit supercavitation as a means to achieve extremely high submerged speeds and low drag [12]. The size of existing or notional supercavitating high-speed bodies ranges from that of projectiles to heavy-weight full-scale torpedoes.

A six DOFs rigid body model describes the dynamic behavior of a supercavitating vehicle of the kind depicted in Figure 10. The vehicle is controlled by a one-degree-of-freedom cavitator, by four fins arranged according to the cruciform configuration shown in Figure 11, and by the propulsion thrust. Additional forces acting on the vehicle include gravity and possible interactions with the cavity. It is assumed that the two horizontal fins are mechanically interconnected and have a common rotation, while the two vertical fins can be controlled independently. The resulting flight mechanics model is governed by the following set of equations:

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \mathbf{y}(t - \tau_r), \mathbf{y}(t - \tau_l), \mathbf{y}(t - \tau_u), \mathbf{y}(t - \tau_d), \mathbf{y}(t - \tau_t), \mathbf{u}(t)), \quad (41a)$$

$$\mathbf{g}(\mathbf{y}(t), \mathbf{y}(t - \tau_r), \mathbf{y}(t - \tau_l), \mathbf{y}(t - \tau_u), \mathbf{y}(t - \tau_d), \mathbf{y}(t - \tau_t)) = 0, \quad (41b)$$

where details of the mathematical model of the cavitator force, the hydrodynamic actions on the fins and the body-cavity interactions can be found in Reference [4], and are here omitted for the sake of brevity.

The model is characterized by five delays:  $\tau_t$  is the delay related to contact forces in the vehicle afterbody, while  $\tau_r, \tau_l, \tau_u, \tau_d$  are the delays related to the rightmost, leftmost, upper and lower fins, respectively. In practice, the forces acting on the fins and the contact interactional forces depend on the shape of the cavity, which in turn is influenced by the past history of the system state. More precisely, according to Logvinovich's independence principle [11], the model assumes that each cavity section expands independently from adjacent ones. The expansion of each section is determined by the state of motion of the cavitator when the cavity is generated, and further distorted by other effects such as buoyancy. As a result, perturbations of the cavitator state of motion produce cavity distortions which propagate towards the after-body according to the vehicle history of motion, and influence hydrodynamic cavity/vehicle interactions with certain time lags which are (slightly) different for each point of contact.

The state vector is composed of 12 variables, namely

$$\mathbf{y} = (x, y, z, \phi, \theta, \psi, u, v, w, p, q, r)^T, \quad (42)$$

where  $x, y, z$  denote the position of a reference point  $P$  of the body with respect to the inertial frame,  $\phi, \theta, \psi$  are the Euler angles (in the 3-2-1 sequence) representing the orientation of a vehicle-attached frame with respect to the inertial frame, while  $u, v, w$  and  $p, q, r$  are the body-attached components of the linear and angular velocity vectors of the vehicle, respectively. The control vector is composed of 5 quantities,

$$\mathbf{u} = (\delta_s, \delta_c, \delta_e, \delta_{vu}, \delta_{vl})^T, \quad (43)$$

where  $\delta_s$  denotes the throttle,  $\delta_c$  and  $\delta_e$  the rotations of the cavitator and elevator, respectively, and  $\delta_u$  and  $\delta_l$  the deflections of the upper and lower vertical fins which as rudders and also provide roll control.

Using the proposed trajectory optimization procedure, we study a family of turn maneuvers for varying heading changes. Specifically, the vehicle, initially in leveled trim at 85 m/s, is required to turn in minimum time and return to the initial trim setting after the desired change in heading. Solutions are found for assigned final heading angle ranging from 10 to 90 degrees. The boundary conditions further specify that the initial and final depths should be the same. The optimization cost function is written as

$$J = T^2 + \frac{1}{\rho} \int_{T_0}^T \dot{\mathbf{u}} \cdot \dot{\mathbf{u}} dt, \quad (44)$$

where  $T$  denotes the unknown final time, while the second term of the cost includes the control velocities  $\dot{\mathbf{u}}$ , and  $\rho = 10$  is a weighting factor.

Figure 12 shows the computed optimal trajectories. The results for increasing required heading change were obtained using a continuation technique, whereby the solution for a given heading is used as a starting guess for the subsequent maneuver. The first turn corresponding to a 10 degree change, is calculated using as a starting guess the steady state and the control values corresponding to the initial trim condition. Each solution is characterized by a single shooting arc and 20 equally spaced DOFs per control input.

Figure 13 shows the time histories of the vehicle Euler angles for some representative turns. The results show that the vehicle undergoes significant roll and pitch during the maneuvers, while following significantly three dimensional trajectories. In particular, Figure 13(a) shows that all maneuvers imply maximum roll angles which are close to 90 degrees. This very pronounced roll is in fact needed for the cavitator to generate sufficient lift to counterbalance the centripetal force during turning [8]. A more sophisticated vehicle, with additional controllable degrees of freedom, may be considered in order to reduce or eliminate the need for such a pronounced roll. The variation of the pitch angle presented in Figure 13(b) confirms the three-dimensional nature of the turns.

Finally, the optimal control inputs are presented in Figure 14. All of the controls, with the exception of the throttle, tend to gradually return to the trim values associated to the flight velocity of 85 m/s with an approximately zero rate condition.

It is important to note that in all the maneuvers considered in this section the vehicle does not come into contact with the cavity. The absence of vehicle/cavity contacts was not specifically enforced by means of constraints, and can be attributed to the fact that impact phenomena imply an increase in drag as well as an increase in the control rates in response to the contact itself. Both effects correspond to an unfavorable increase of the cost function (44), since the additional drag will slow down the vehicle and hence increase the flight time. In general, however, and particularly for highly aggressive maneuvers, the absence of contacts with the cavity can be imposed as part of the definition of the maneuver itself, with the objective

to avoid, or minimize the occurrence of drag increase, high control rates, vibrations, noise, etc.

## 5 Conclusions

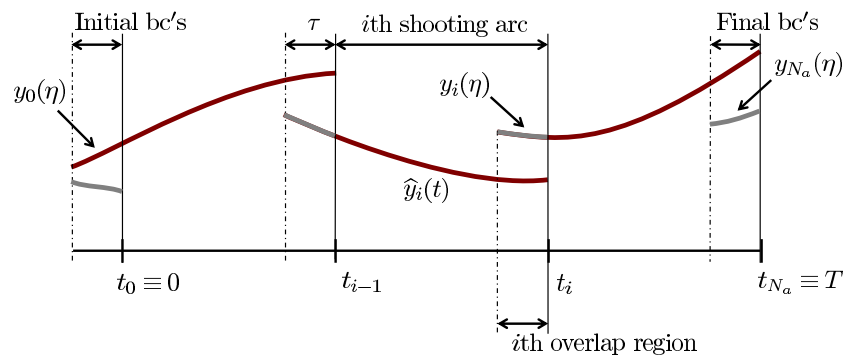
Motion planning of mechanical systems governed by DDEs creates a need for general and efficient numerical methods to solve optimal control problems with time delays. Here, we have proposed a direct multiple shooting approach, specifically adapted to handle DDEs. The proposed algorithm has been successfully employed to solve two simple test cases, and to optimize maneuvers for a state-of-the-art supercavitating vehicle model. From this study, the following conclusions can be drawn:

- In using direct methods for the solution of this class of problems, multiple shooting should be preferred over transcription (collocation) methods whenever the time step necessary for capturing the faster components of interest of the solution is much smaller than the time delay and, even more so, if the delay is in turn much smaller than the temporal domain. In those cases in fact, the use of transcription-type methods would lead to extremely large numbers of unknowns and constraints, and hence to overwhelming computational costs. On the contrary, in the multiple shooting case the dimension of the optimization problem is independent from the time resolution necessary to march forward in time the model equations, so that problems can be solved efficiently at affordable computational costs.
- Between two consecutive shooting arcs, DDEs induce state continuity conditions defined on a time line. Although equality conditions at each integration point of the overlap region can be considered, we argued that the use of variable step integrators makes this option unfeasible and justifies the enforcement of these conditions in a least-squares sense. A direct and welcome byproduct of this approach is a reduced number of local constraints and unknowns.
- The presented algorithm can be fruitfully applied to solve complex nonlinear problems, and it can be easily programmed starting from generic direct multiple shooting software for ODE optimal control problems.

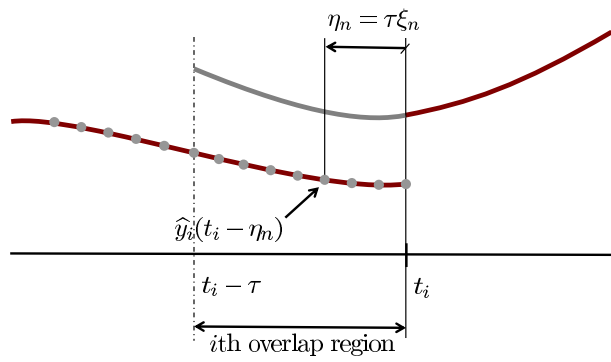
## References

1. Ascher, U.M., Petzold, L.R.: Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1998)
2. Barkley, A., Gill, P.E., Rosen, J.B.: SQP methods and their application to numerical optimal control. Report NA 97-3, Department of Mathematics, University of California, San Diego, CA (1997)
3. Betts, J.T.: Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics*, **21**(2), 193–207 (1998)

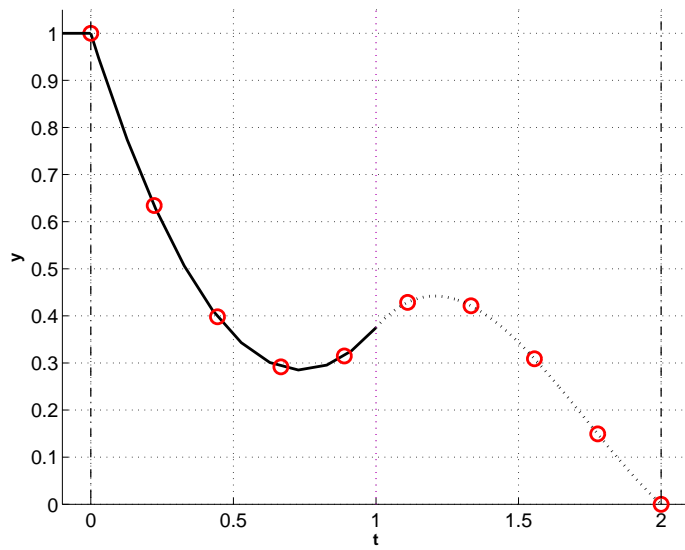
4. Bottasso, C.L., Scorcelletti, F., Ruzzene, M., Ahn, S.S.: Trajectory optimization for DDE models of supercavitating underwater vehicles. *ASME Journal of Dynamic Systems, Measurement, and Control*, to appear (2008)
5. Bryson, A.E., Ho, Y.C.: *Applied Optimal Control*. Wiley, New York (1975)
6. Eller, D.H., Aggarwal, J.K., Banks, H.T.: Optimal control of linear time-delayed systems. *IEEE Transaction on Automatic Control*, **AC-14**(6), 678–687 (1969)
7. Gill, P.E., Murray, W., Wright, M.H.: *Practical Optimization*. Academic Press, London and New York (1981)
8. Kirschner, I.N., Kring, D.C., Stokes, A.W., Fine, N.E., Uhlman, J.S.: Control strategies for supercavitating vehicles. *Journal of Vibration and Control*, **8**, 219–242 (2002)
9. Kuang, Y.: *Delay Differential Equations with Applications in Population Dynamics*. Academic Press, San Diego, CA (1993)
10. Hale, J.K., Verduyn Lunel, S.M.: *Introduction to Functional Differential Equations*. Springer (1993)
11. Logvinovich, G.V.: Hydrodynamics of free-boundary flow. U.S. Department of Commerce, Washington, DC, Tech. Rep., translated from Russian (NASA-TT-F-658) (1972)
12. Miller, D.: Going to war in a bubble. *International Defense Review*, (1995)
13. Mueller, T.E.: Optimal control of linear systems with time lag. University of Illinois, Urbana, Coordinated Science Lab. Rept. R-254, (1965)
14. Renegar, J.A.: *Mathematical View of Interior Point Methods in Convex Optimization*. SIAM, Philadelphia (2001)
15. Shampine, L.F., Thompson, S.: Solving DDEs in Matlab. *Applied Numerical Mathematics*, **37**, 441–458 (2001)



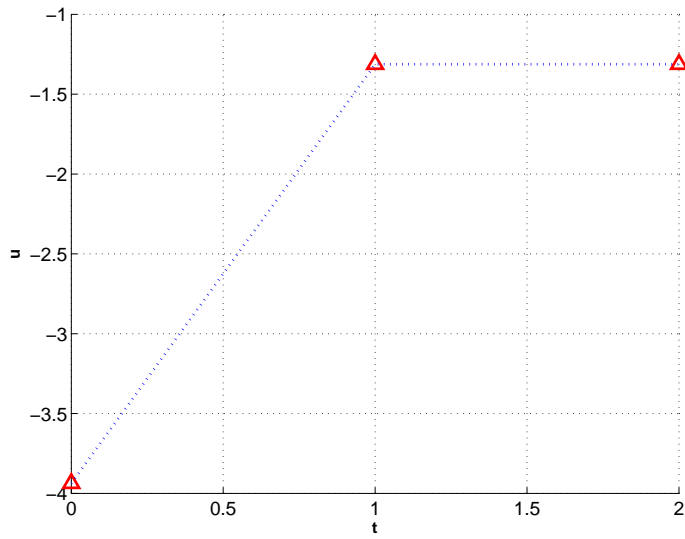
**Fig. 1** Line continuity conditions for the state time history of a DDE model using multiple shooting



**Fig. 2** Detail of the overlap region between two shooting segments, showing the discretized solution for the states obtained by forward time marching of the equations of motion



**Fig. 3** State time history for the single state DDE example. Circles: numerical solution. Dotted line: analytical solution. Solid line: overlapping regions for enforcement of state continuity conditions



**Fig. 4** Control time history for the single state DDE example. Dotted line: exact solution. Triangles: discrete numerical solution

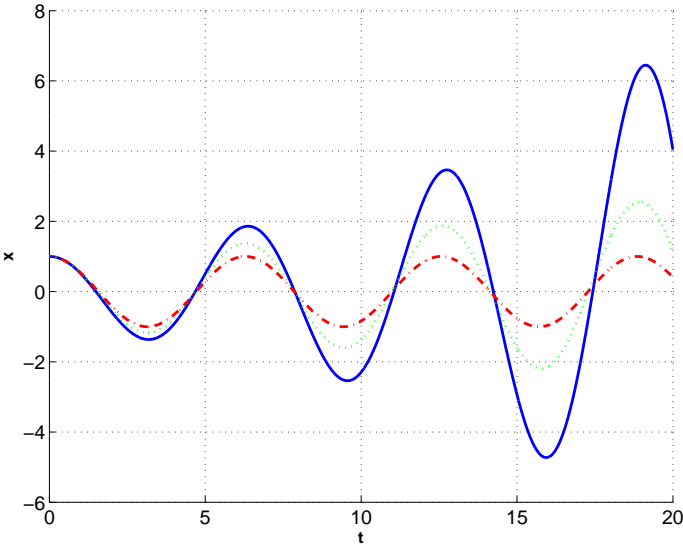
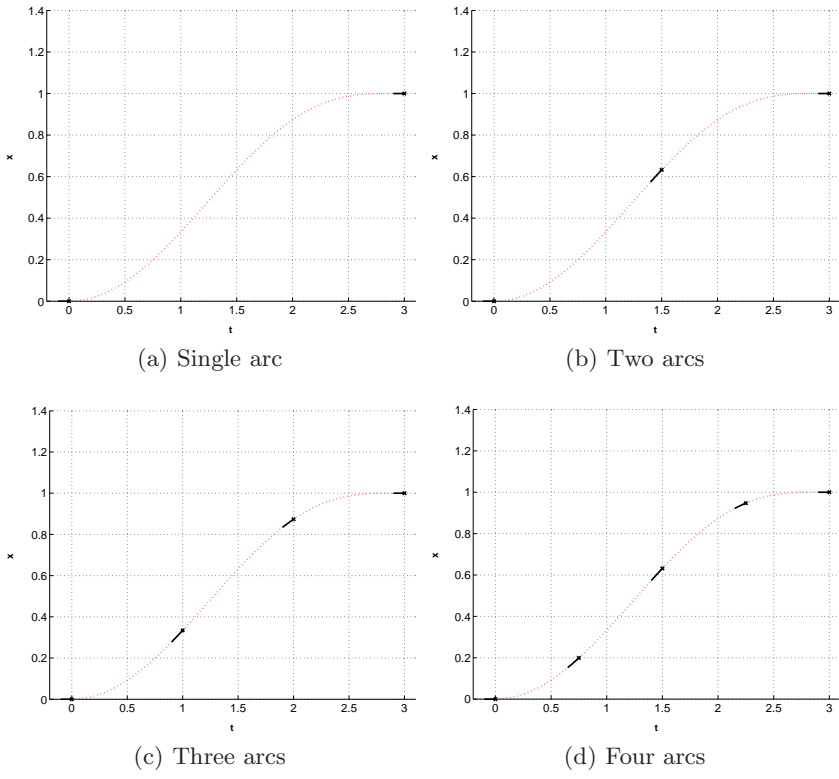
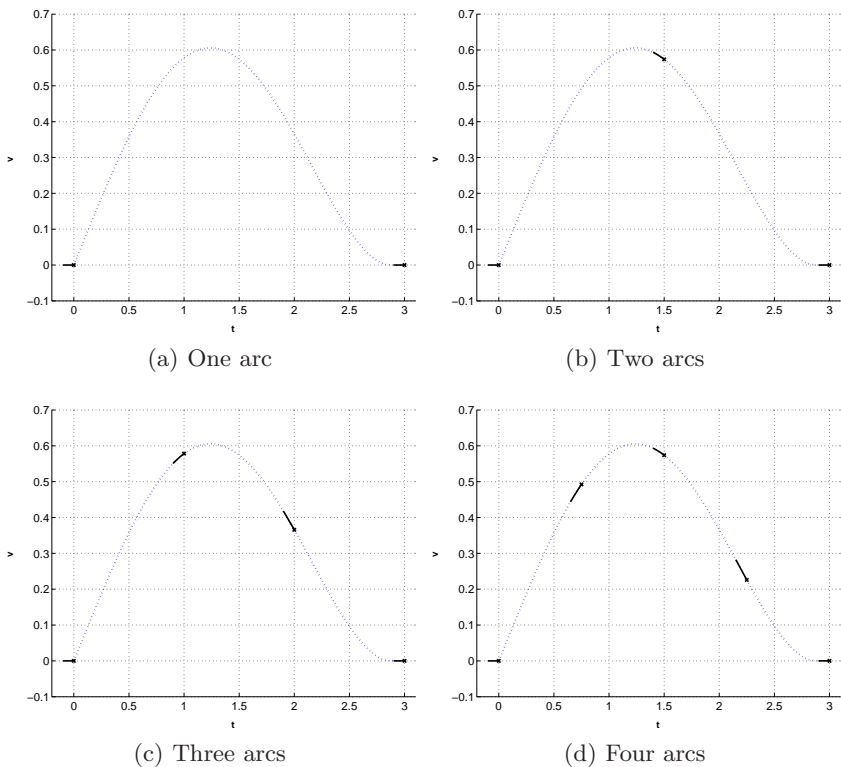


Fig. 5 Spring-mass system: free response for several values of the delay



**Fig. 6** Computed optimal position response for different numbers of integration arcs. Solid line: state interpolation on overlapping regions



**Fig. 7** Computed optimal velocity response for different numbers of integration arcs. Solid line: state interpolation on overlapping regions

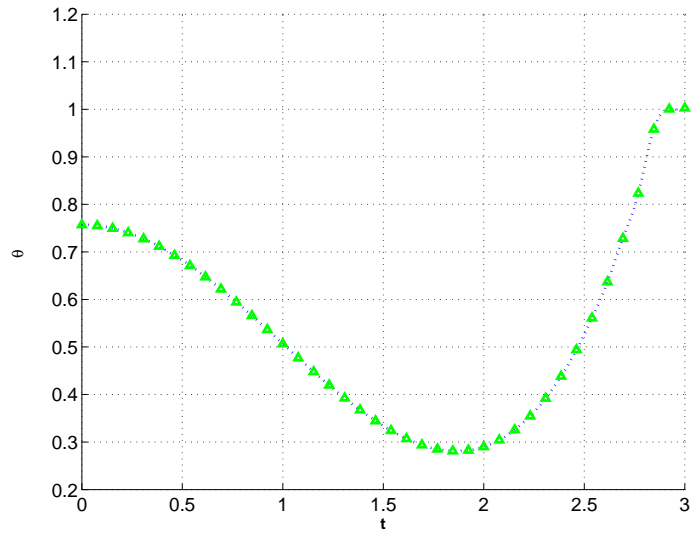
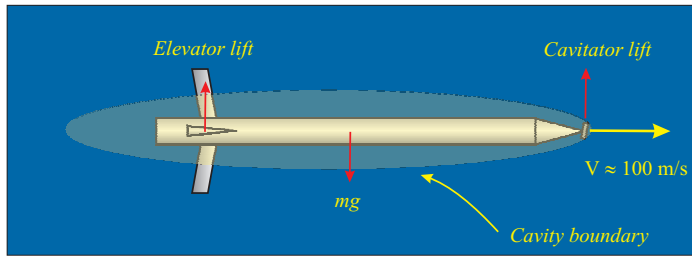
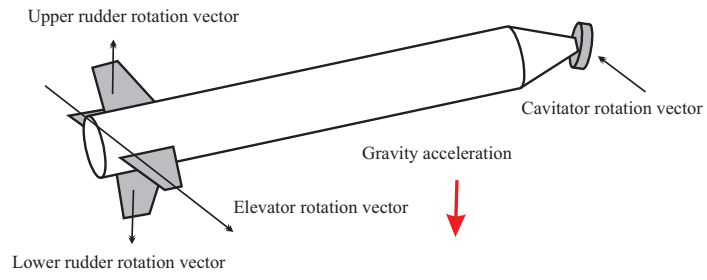


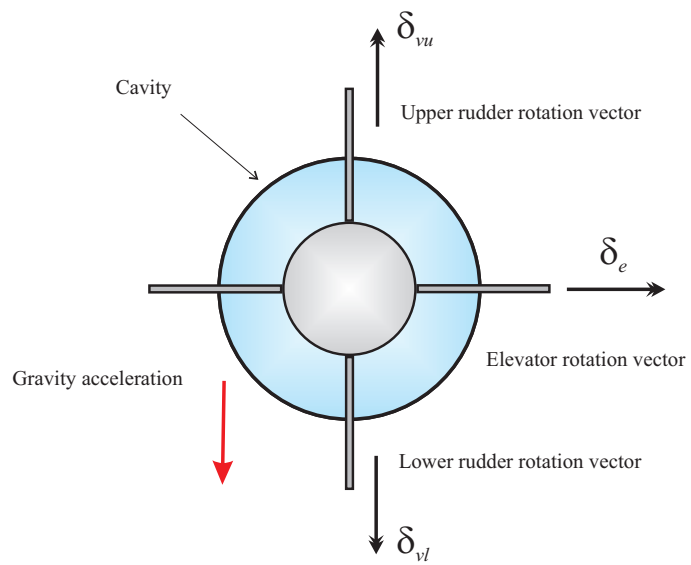
Fig. 8 Computed optimal input time history



**Fig. 9** Underwater projectile in supercavitating regime



**Fig. 10** Schematic view of vehicle configuration



**Fig. 11** Cruciform configuration of vehicle fins

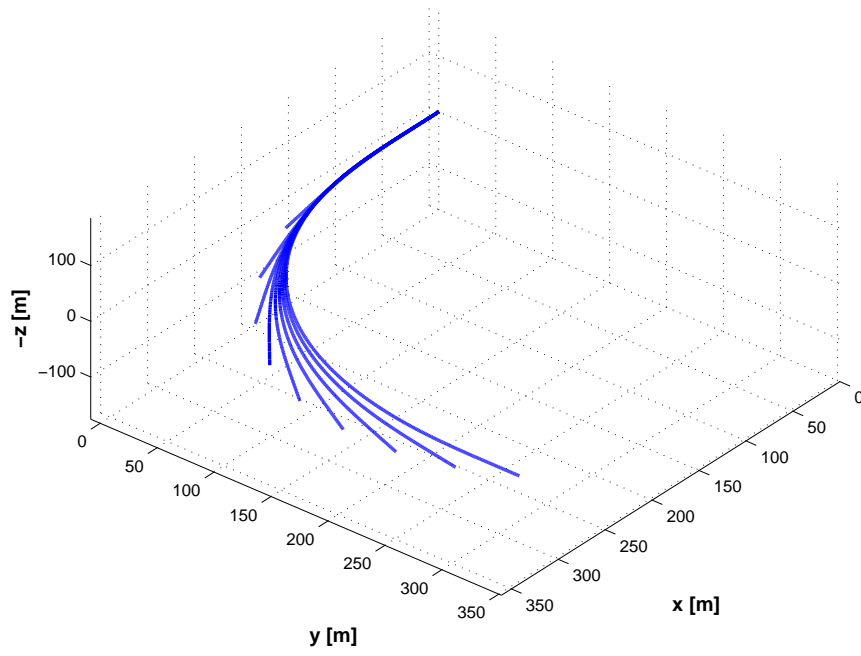
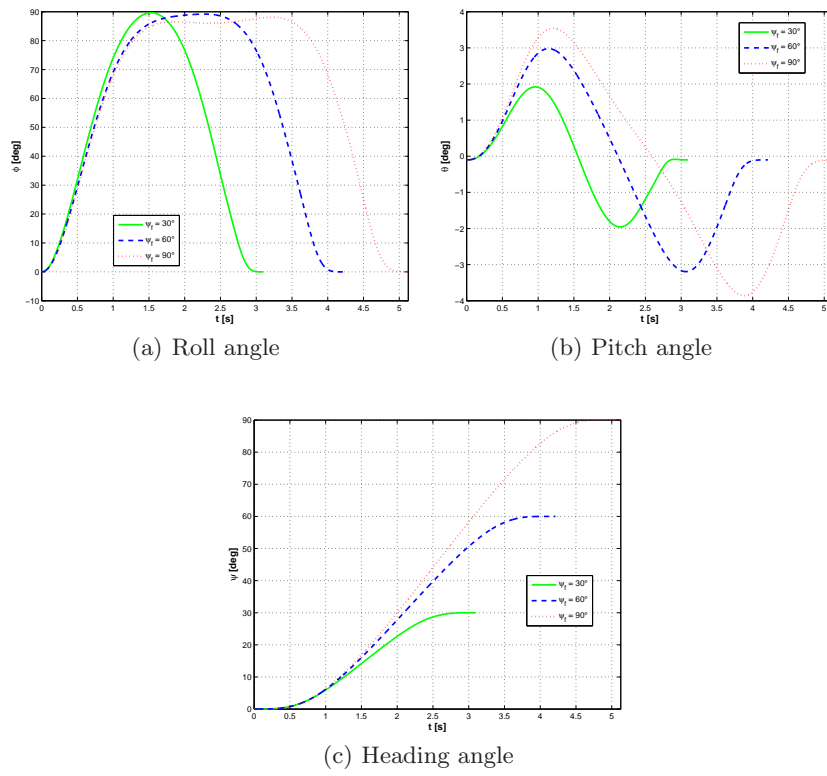
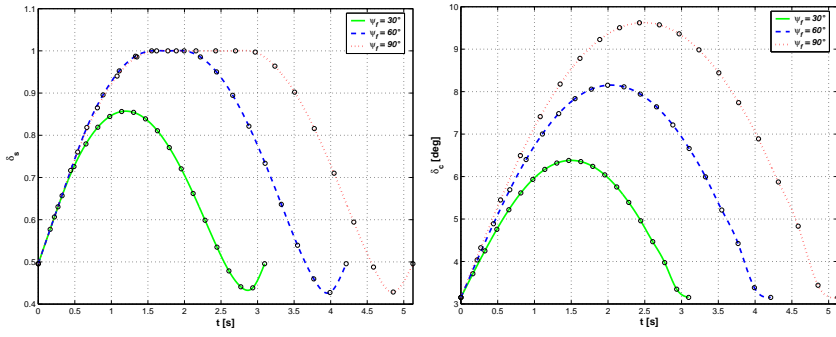


Fig. 12 Family of optimal minimum time turns for varying heading changes

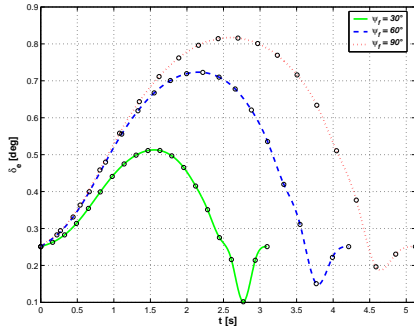


**Fig. 13** Euler angles for representative minimum time turns

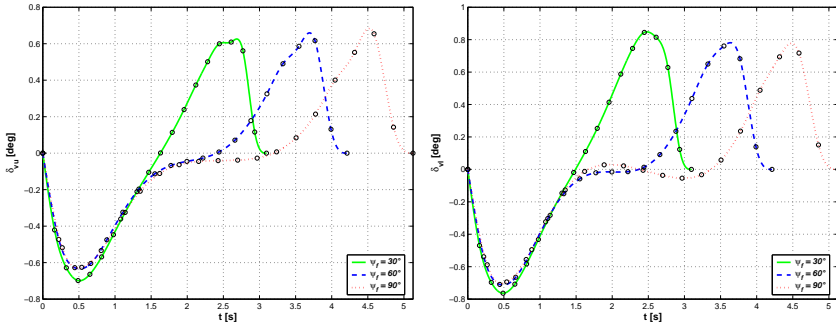


(a) Throttle

(b) Cavitator



(c) Elevator



(d) Upper rudder

(e) Lower rudder

Fig. 14 Control inputs for representative minimum time turns