MULTIBODY DYNAMICS 2003
Jorge A.C. Ambrósio(Ed.)
IDMEC/IST, Lisbon, Portugal, July 1-4 2003

SYMBOLIC-NUMERIC INDIRECT METHOD FOR SOLVING OPTIMAL CONTROL PROBLEMS FOR LARGE MULTIBODY SYSTEMS: THE TIME-OPTIMAL RACING VEHICLE EXAMPLE

E. Bertolazzi, F. Biral, and M. Da Lio

- Department of Mechanical and Structural Engineering,
- University of Trento
Via Mesiano 77, 38050 Trento, Italy
Tel. +39 461 88 2504, Fax: +39 461 88 2599.
e-mail: francesco.biral@ing.unitn.it, website: http://www.dinamoto.it/

Abstract. *This work presents a methodological framework, based on an indirect approach, for the automatic generation and numerical solution of Optimal Control Problems (OCP) for mechatronic systems, described by a system of Differential Algebraic Equation (DAEs). The equations of the necessary condition for optimality were derived exploiting the DAEs structure, according to the Calculus of Variation Theory. A collection of symbolic procedures was developed within a general-purpose Computer Algebra Software. Those procedures are general and make it possible to generate both OCP equations and their jacobians, once any DAE mathematical model, objective function, boundary conditions and constraints are given. Particular attention has been given to the correct definition of the boundary conditions especially for models described with set of dependent coordinates. The non-linear symbolic equations, their jacobians with the sparsity patterns, generated by the procedures above mentioned, are translated into a C++ source code. A numerical code, based on a Newton Affine Invariant scheme, was also developed to solve the BVPs as generated by such procedures. The software and methodological framework here presented were successfully applied to the solution of the minimum-lap time problem of a racing motorcycle.*

1 INTRODUCTION

The main objective of this paper is to present a combination of a symbolic-numeric indirect method to solve efficiently in a semi-automatic way Optimal Control Problems for large multibody systems.

In the last decade the applications of the Optimal Control Problems (sometimes called also Dynamic Optimization) have spread over many research and industrial fields as proved by the bibliography (for a review see[1-2-3].). However, even if the theoretical framework (namely

Calculus of Variation, Dynamic Programming, Non Linear Program (NLP)) has been quite clearly defined, its application to industrial or real problems is still difficult for the complexity and dimensionality of the system of equations to deal with. Two groups of techniques are commonly employed to solve this kind of problem. The first group involves the so called *direct methods*, which instead of solving the necessary condition discretize the problem in order to obtain a NLP, which may be solved by means of consolidated numerical schemes, such as Initial Value Solver (IVS) and Sequential Quadratic Programming (SQP). In this sense they are easier to implement, and probably this is the main reason why they are widely employed. The second group consists of *indirect methods*, which are based on the solution of the necessary condition of optimality, as derived by the Calculus of Variation. Indirect Methods usually produce very accurate solution and are quite sensitive to design parameters variations, since they solve directly the equations of first necessary condition. However, to fully exploit the Indirect Method one has to produce the symbolical expressions of the necessary conditions of optimality. In other words, according to the calculus of variations, the adjoint equations have to be calculated hopefully along with their jacobians for the numerical solution of the arising Two Point Boundary Value Problem (TPBVP). In addition, the number of adjoint equations (also called co-state equations) is equal to the original problem dimension, thus the indirect approach at least doubles the problem dimensionality. Actually, adjoint equations should be solved also in the direct methods. However, in many cases, this is done as a post processing proof and adjoint variables can be obtained by a post-optimal calculation using the Lagrange multipliers of the resulting nonlinear optimization problem.

In addition the difficulty of the problem is worsened by the fact that the mathematical models of multibody and mechatronic systems have been growing in complexity, to better simulate and study prototypes and industrial products. They are composed of many bodies connected each others by various kind of constraints. In particular they include subsystems that reproduce non-linear behavior of different elements or entities such as tyre, control blocks, aerodynamic forces, etc. Thus, these models, very often, consists of hundreds of higly non-linear algebraic differential equations. It is clear that the application of the indirect method to such large systems requires massive symbolic manipulations that has discouraged researchers of this field to adopt this technique, as emerges from the bibliography[4-5]. Moreover, the use of commercial multibody software is commonly preferred because it is easier to build the multibody model, they are integrated with FEM and CAD software and can be used as IVS in the NLP approach.

Anyway, in both cases (direct or indirect method) it is essential to find a solution of the TPBVP in reasonably fast time. Thus, in dealing with such large problems, the numerical algorithm plays a major. The most important numerical schemes are three. *Single Shooting*, which suffers from high sensitivity to guessed values and the fact that some equations (typically the co-state equations) may be very-fast diverging. *Multiple shooting* that follows the same idea as single shooting, but the domain is divided in smaller subintervals reducing sensitivity to guessed values and divergence specially when combined with direct collocation. Finally, there are the *discretization* or *global methods*, which are reckoned to be the most stable, and the solution is obtained simultaneously for the whole domain. An hybrid solution were presented by Burlish et al.[6] where a direct method is used to find a quite good guess

solution for indirect method whose equations (symbolically derived) are solved with a Multiple Shooting technique, to obtain an accurate solution.

The approach presented in this work wants to exploit the advances made in the symbolic manipulation tools to fully use an indirect approach for solving OCP. It is shown how the first necessary condition of optimality can be derived symbolically exploiting the structure of the DAEs of the multibody system even if they are very large. The obtained system of equations is solved with a global method based on a Newton Affine Invariant scheme. To simplify the generation of equations, a collection of symbolic procedures were developed in a general-purpose Computer Algebra Software. Those procedures are general and automatically generate the equations for the OCP and the jacobian blocks for its finite difference discretization, once any DAE system, objective function, boundary conditions and trajectory constraints are given. The obtained equations are translated into C++ source code and the BVPs is solved by means of a software based on an affine invariant Newton scheme.

In particular a new formulation for the boundary conditions is presented, being them critical for numerical method convergence. This formulation is intended for large multibody system described with set of dependent coordinates. In this work was tested on the motorcycle model by M. Da Lio et al.[3] for future application to a more complex model. An example of solution a minimum-lap time problem for a real circuit is also shown.

2 EFFICIENT SYMBOLIC GENERATION OF NECESSARY CONDITION EQUATIONS OF OPTIMALITY AND THEIR JACOBIANS

2.1 The formulation of Optimal Control Problem for a multibody systems

The general variational formulation of an OCP consists in finding the controls $\boldsymbol{u}(s) \in \mathfrak{R}^m$ that minimizes the functional:

$$J[\mathbf{x},\mathbf{u}] = \int_{s_i}^{s_f} f(\mathbf{x}(s),\mathbf{u}(s))ds \ , \tag{1}$$

under differential and algebraic constraints. Here, the symbol denotes the state vector $\boldsymbol{x}(s) \in \mathfrak{R}^n$ and s is the independent variable (which can have physical meaning or as the considered case it is a re-parametrization of the time domain). The minimum is constrained and it is assumed to satisfy the following ODE (or DAE):

$$\mathbf{a}(\mathbf{x}(s),\dot{\mathbf{x}}(s),\mathbf{u}(s)) = \mathbf{0}, \qquad s \in \left(s_i, s_f\right) \tag{2}$$

where $\mathbf{a}(\mathbf{x},\dot{\mathbf{x}},\mathbf{u}) = \left(a_i(\mathbf{x},\dot{\mathbf{x}},\mathbf{u})\right)_{i=1}^{n}$. Equation (2) represents multibody or mechatronic systems dynamics. Algebraic constraints are also present and are divided in two groups, equality and inequality ones. The equality constraints are pointwise and are used to set initial and final boundary conditions as follows:

$$\mathbf{c}(\mathbf{x}(s_i),\mathbf{x}(s_f),\boldsymbol{\rho}) = \mathbf{0}, \tag{3}$$

where $\mathbf{c}(\mathbf{x}(s_i),\mathbf{x}(s_f),\boldsymbol{\rho}) = \left(c_i(\mathbf{x}(s_i),\mathbf{x}(s_f),\boldsymbol{\rho})\right)_{i=1}^{p}$. Vector $\boldsymbol{\rho}$ is a list of parameters used to "set" unknown or cyclic boundary conditions. Boundary condition aspects will be discussed in more details in section 2.3.
Inequality constraints are set along the trajectory on state variables and controls and are written as follows:

$$\mathbf{d}(\mathbf{x}(s),\mathbf{u}(s)) \le \mathbf{0}, \qquad s \in \left(s_i, s_f\right) \tag{4}$$

where $\mathbf{d}(\mathbf{x},\mathbf{u}) = \left(d_i(\mathbf{x},\mathbf{u})\right)_{i=1}^{q}$. These inequalities are used to describe the domain of the solution and limitation of the control variable $\boldsymbol{u}$(s).

The OCP based on equations (1)—(4) is formulated in a constrained variational way, and can be transformed into an unconstrained formulation. Inequality constraints are approximated by means of penalty functions, while differential and equality constraints can be eliminated by introducing a set of suitable Lagrangian multipliers ($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$). This formulation is possible at the cost of *n* more unknowns for the elimination of the differential constraints an *p* more unknowns for the eliminations of the equality constraints. The resulting functional is the following:

$$J\left[\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}\right] = \boldsymbol{\mu}\cdot\mathbf{c}\left(\mathbf{x}\left(s_i\right),\mathbf{x}\left(s_f\right),\boldsymbol{\rho}\right) + \int_{s_i}^{s_f}\left[\, f_p(\mathbf{x}(s),\mathbf{u}(s)) + \boldsymbol{\lambda}(s)\cdot\mathbf{a}(\mathbf{x}(s),\dot{\mathbf{x}}(s),\mathbf{u}(s))\right] ds\,, \tag{5}$$

where:

$$f_p(\mathbf{x}(s),\mathbf{u}(s)) = f(\mathbf{x}(s),\mathbf{u}(s)) + \sum_{i=1}^{q} p_i\left(d_i\left(\mathbf{x},\mathbf{u}\right)\right),$$

and p_i is the penalty function associated to the *i-th* components of inequalities (4) and takes the form desired.

2.2 The necessary condition of Optimality for multibody system

The Theory of the Calculus of Variations states that the necessary condition of optimality of functional (5) is a system of equations, which comes from the stationary condition of its first variation. The theory is well established and in many papers and textbooks it is possible to find the equation derivation[7-8]. However, in deriving these equations, some simplifications can still be introduced if the particular structure of the multibody DAE system is considered. Those simplifications reduce computation time of the symbolic manipulations, which can be an advantage with multibody systems described with hundreds of state variables.

Typically the equation of motions of multibody system (2) can be reduced to a system of differential equations linear in $\dot{\mathbf{x}}$. Thus:

$$\mathbf{a}(\mathbf{x},\dot{\mathbf{x}},\mathbf{u}) = \mathbf{A}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{b}(\mathbf{x},\mathbf{u})\,, \tag{6}$$

where $\mathbf{A}(\mathbf{x}) = \left(A_{ij}(\mathbf{x})\right)_{i,j=1}^{n}$ and $\mathbf{b}(\mathbf{x}) = \left(\mathbf{b}_i(\mathbf{x})\right)_{i=1}^{n}$. If the system is described with a minimal set of coordinates, then matrix $\mathbf{A}(\mathbf{x})$ is not singular. However, the majority of multibody systems are described with set of dependent coordinates, such as natural coordinates, Euler parameters, etc[9]. This approach lets consider each body as an independent entity connected to others by means of constraints. The resulting kinematics relationships and equations of motion are simpler compared to models described with a minimal set of coordinates especially when the kinematic chain is quite long. As a drawback one gets that the number of equations is greater. Constraints that have to be added can be both differential and algebraic equations and the final system is a differential algebraic system (DAEs) that leads to a singular matrix $\mathbf{A}(\mathbf{x})$ in (6). However, it is again possible to transform the DAEs into a ODEs by means of some stabilization technique (Baumgarte, penalty formulation for example[9]). In both cases from the assumption (6) follows some simplifications when deriving the equations of the first necessary condition of optimality. Taking the first variation of (5) and setting it to zero, (see Appendix I) one obtain the following boundary value problem (BVP) written in a vector form:

$$\mathbf{M}\left(\mathbf{y}(s)\right)\dot{\mathbf{y}}(s) + \mathbf{n}\left(\mathbf{y}(s),\mathbf{u}(s)\right) = \mathbf{0}, \qquad s \in (s_i, s_f) \tag{7A}$$

$$\mathbf{h}\left(\mathbf{y}(s_i),\mathbf{y}(s_f),\boldsymbol{\rho}\right) = \mathbf{0} \tag{7B}$$

$$\mathbf{g}\left(\mathbf{y}(s),\mathbf{u}(s)\right) = \mathbf{0}, \quad s \in (s_i,s_f) \tag{7C}$$

The expressions of the previous vectors are:

$$\mathbf{M}(\mathbf{y}) = \begin{pmatrix} \mathbf{T}(\mathbf{x},\boldsymbol{\lambda}) & -\mathbf{A}(\mathbf{x})^T \\ \mathbf{A}(\mathbf{x}) & \mathbf{0} \end{pmatrix}, \quad \mathbf{n}(\mathbf{y},\mathbf{u}) = \begin{pmatrix} \dfrac{\partial d(\mathbf{x},\boldsymbol{\lambda},\mathbf{u})}{\partial \mathbf{x}} \\ \dfrac{\partial d(\mathbf{x},\boldsymbol{\lambda},\mathbf{u})}{\partial \boldsymbol{\lambda}} \end{pmatrix}, \qquad \mathbf{g}(\mathbf{y},\mathbf{u}) = \frac{\partial d(\mathbf{x},\boldsymbol{\lambda},\mathbf{u})}{\partial \mathbf{u}}, \tag{7D}$$

where:

$$\mathbf{y} = \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix}, \quad d(\mathbf{x},\boldsymbol{\lambda},\mathbf{u}) = f_p(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T\mathbf{b}(\mathbf{x},\mathbf{u}) \text{ and } \mathbf{T}(\mathbf{x},\boldsymbol{\lambda}) = \frac{\partial\left(\mathbf{A}(\mathbf{x})^T\boldsymbol{\lambda}\right)}{\partial \mathbf{x}} - \frac{\partial\left(\mathbf{A}(\mathbf{x})^T\boldsymbol{\lambda}\right)^T}{\partial \mathbf{x}},$$

and the form of $\mathbf{h}\left(\mathbf{y}(s_i),\mathbf{y}(s_f),\boldsymbol{\rho}\right)$ is discussed forward.

The equation system composed of (7A)—(7C) is derived from first variation necessary condition. Vector form (7A) is the ODEs (DAEs if $\mathbf{A}(\mathbf{x})$ is singular), which includes the initial problem system of equations (6) plus the adjoint equations. In fact, if one substitutes the structure of matrix $\mathbf{M}(\mathbf{y})$ and vector $\mathbf{n}(\mathbf{y},\mathbf{u})$ into (7A), it can be rewritten as follows:

$$\mathbf{T}(\mathbf{x},\boldsymbol{\lambda})\dot{\mathbf{x}} - \mathbf{A}(\mathbf{x})^T\dot{\boldsymbol{\lambda}} + \frac{\partial}{\partial \mathbf{x}}\left[f_p(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T\mathbf{b}(\mathbf{x},\mathbf{u})\right] = \mathbf{0}, \tag{8A}$$

$$\mathbf{A}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{b}(\mathbf{x},\mathbf{u}) = 0, \tag{8B}$$

where (8B) is the starting multibody system of equation (6) and (8A) are the adjoint equations.

It is clear that simplifications come from (6) in the generation of the adjoint equations. Only $\mathbf{T}(\mathbf{x},\boldsymbol{\lambda})$ and $\mathbf{n}(\mathbf{y},\mathbf{u})$ must be calculated while the other terms are already available. It is also interesting to point out that the multibody equations have to be kept with their implicit form since the explicit one requires the inversion of matrix $\mathbf{A}(\mathbf{x})$, which for large system is impractical and can end to subsequent numerical ill-conditioning.

The vector form (7B) represents boundary conditions and (7C) is the vector form of system of algebraic equations, which give the optimal inputs as function of ***y***. Usually (7C) is a system of non-linear equations in controls ***u*** and becomes linear if the controls are quadratically penalized.

2.3 Boundary conditions formulation

The boundary conditions are a very important aspect in the OCP. They can deeply change or even compromise the OCP solution. This is the reason for dedicating a whole paragraph to this issue. By means of boundary conditions it is possible to fix the whole multibody initial and final motion or more often only part of it. In principle it is also possible to leave all state variable values unset (also referred as "free") being this case, for example, meaningful for a minimum lap time problem. This means that equation (3) is empty and initial and final system conditions will be computed according the calculus of variations in order to be the best possible ones that minimize the functional J in equation (5). However, while it is practicable the case with all final states unset, it seems that it is quite difficult to do the same with initial states unless some precautions are taken. The reasons lay in the influence of trajectory constraints that should also act on the boundary conditions and actually they act in the continuous formulation, but this is not any more true when the problem is discretized. This fact it is partially hidden by the constraint elimination via penalties, which includes the constraints in the integral of (5) and not on the boundary conditions with μ Lagrange multipliers. As a consequence equation (7E) does not contains trajectory constraints.

Thus, it is necessary to include them at the boundary values in the discretized formulation. This is especially critical for the initial conditions since the solution at the final boundary it is a result of the past motion, while the initial one has no memory of what was before. Thus, the best initial motion can be anyone that respects system equations (2) and eventually equation (3). Very often the highest values for control forces are choosen and the initial motion can end up in a non physical solution (if constraints on the initial states are not enforced on boundaries) or worst in a numerical instability, when high state variable rates are involved. In the following figure a numerical oscillation induced by high force rate at the initial boundary is shown. (It is referred to *case 2* introduce in section 4).

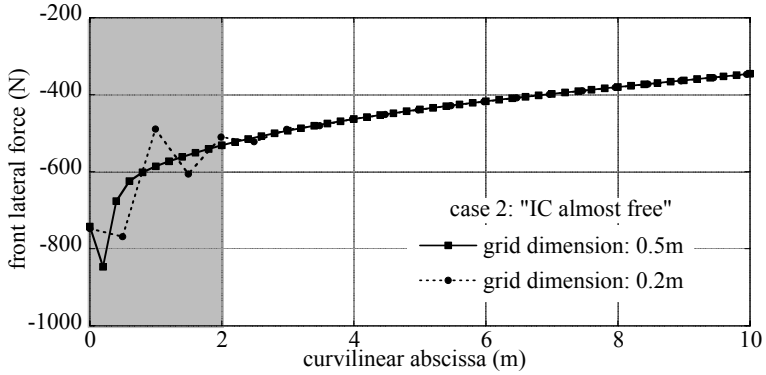

Figure 1: influence of grid dimension close to boundary conditions

Thickening mesh point close to boundaries may cure numerical instability. However to prevent non-physical solution trajectory constraints have to be enforce on boundary conditions.

Anyway in many practical cases one might desire to put some conditions on initial and final states. While this is straightforward with mathematical models described with set of independent coordinates, which have direct physical meaning, especially if they correspond to system's degree of freedoms, it is not so simple if mathematical models are described with set of dependent coordinates. In the last case a map between independent coordinates and dependent ones has to be supplied, in order to easily put boundary conditions on state variables.

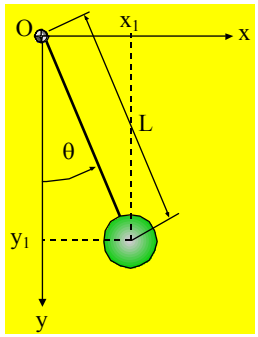

Figure 2: Pendulum; one degree of freedom (θ), two natural coordinates (x_1, y_1)

To let the reader better understand the concept a simple example will be used. Let us consider a simple pendulum (see figure 2). This system has one degree of freedom, the rotation about point O, indicated with variable θ. If the system is described by means of the variable θ, it is quite easy to put an initial condition on the pendulum position and velocity. In fact, in that case the initial conditions would be: $\theta(t=0)=\theta_0$ and $\dot{\theta}(t=0)=\dot{\theta}_0$. On the other hand, if the set of coordinates used to describe the pendulum are (x_1, y_1) (see figure 2), the

initial conditions would be: $\begin{cases} x_1(t=0) = L\sin(\theta_0) \\ y_1(t=0) = L\cos(\theta_0) \end{cases}$ and $\begin{cases} \dot{x}_1(t=0) = L\cos(\theta_0)\dot{\theta}_0 \\ \dot{y}_1(t=0) = -L\sin(\theta_0)\dot{\theta}_0 \end{cases}$. The last one represents the map introduced above.

Thus with large multibody systems which are commonly described with set of dependent coordinates (in many cases not directly linked to the system degree of freedom) a map that transforms a minimal set of physical coordinates into the system set of coordinates is required. Of course it is possible that not all physical coordinates are known or one may not desire to set a value for all of them.

We restrict the form of boundary conditions to the following form:

$$\mathbf{c}(\mathbf{x}_i, \mathbf{x}_f, \boldsymbol{\rho}) = \begin{pmatrix} \mathbf{x}_i - \mathbf{L}_i(\boldsymbol{\rho}) \\ \mathbf{x}_f - \mathbf{L}_f(\boldsymbol{\rho}) \end{pmatrix} \tag{9}$$

where $\boldsymbol{\rho}$ are the unknown or not-set physical coordinates. The idea of not distinguish between parameters for initial and final conditions, let one easily put cyclic boundary conditions. In fact going back to the simple pendulum example a cyclic condition for the angular position is:

$$\begin{cases} x_1(t=0) = L\sin(\theta_0) \\ y_1(t=0) = L\cos(\theta_0) \\ x_1(t=T) = L\sin(\theta_0) \\ y_1(t=T) = L\cos(\theta_0) \end{cases} \tag{10}$$

where only one parameter is used: θ_0.

2.4 Automatic symbolic generation of necessary conditions and their jacobians

The matrix formulation used for the necessary conditions (equations (7A), (7C) and (11)) and the assumption of equation (6) let one easily automatise the equation generation in a general-purpose Computer Algebra Software. Moreover, as one may see in (7D) the calculation of matrix $\mathbf{M}(\mathbf{y})$ means computing only the matrix $\mathbf{T}(\mathbf{x},\boldsymbol{\lambda})$, being matrix $\mathbf{A}(\mathbf{x})$ already available. This turns out in a saving of computation time when generating the equations. Symbolic computation time is a key issue. Then, since for the numerical solution of the BVP-DAE system the jacobian matrices of the equations are also necessary and the authors of this paper consider very important the availability of the jacobian symbolic expression for numerical method convergence. From this point of view, beign the symbolic jacobian calculation the heaviest part it is still possible to exploit the structure in order to reduce computation time. Finally being the jacobian matrices very sparse they are symbolically analyzed to extract the sparsity of non zero elements. Sparsity patterns and non-zero elements are stored to be exploited both to speed up numerical algorithm.

It is interesting to point out that these procedure for symbolic generation are general independent from the multibody system used or the mechatronic system, provided that it is given in the form of (6).

3 GLOBAL METHOD FOR THE FAST NUMERICAL SOLUTION OF THE MPBVP

3.1 Discretization of the BVP-DAE

The solution of the infinite dimensional problem (7A-C) is approximated by a finite dimensional one by discretizing it. The interval $[s_i, s_f]$ is split into N subintervals,

$s_i = s_0 < s_1 < .\cdots < s_N = s_f$

Equations (7C) are evaluated on the node $s_{k+1/2} = \frac{s_k + s_{k+1}}{2}$,

$$\mathbf{0} = \mathbf{g}(\mathbf{y}(s_{k+1/2}), \mathbf{u}(s_{k+1/2})) = \mathbf{g}\left(\frac{\mathbf{y}(s_k) + \mathbf{y}(s_{k+1})}{2}, \mathbf{u}(s_{k+1/2}) \right) + O(h^2) \tag{11}$$

where $h = \max\{s_k - s_{k-1}, k = 1,2,\ldots,N\}$. Equations (7A) are approximated by using the midpoint quadrature rule to average on $[s_k, s_{k+1}]$ and by using finite differences in place of the derivative terms:

$$\begin{aligned} \mathbf{0} &= \frac{1}{s_{k+1} - s_k} \int_{s_k}^{s_{k+1}} \mathbf{M}(\mathbf{y}(s))\mathbf{y}(s) + \mathbf{n}(\mathbf{y}(s), \mathbf{u}(s)) ds \\ &= \mathbf{M}\left(\frac{\mathbf{y}(s_{k+1}) + \mathbf{y}(s_k)}{2} \right) \frac{\mathbf{y}(s_{k+1}) - \mathbf{y}(s_k)}{s_{k+1} - s_k} + \mathbf{n}\left(\frac{\mathbf{y}(s_{k+1}) + \mathbf{y}(s_k)}{2}, \mathbf{u}(s_{k+1/2}) \right) + O(h^2) \end{aligned} \tag{12}$$

Using (12) and (11), and neglecting the truncation term of order $O(h^2)$ we obtain the following non-linear system:

$$\Phi(\mathbf{W}) = \begin{cases} \mathbf{M}\left(\frac{\mathbf{y}_{k+1} + \mathbf{y}_k}{2} \right) \frac{\mathbf{y}_{k+1} - \mathbf{y}_k}{s_{k+1} - s_k} + \mathbf{n}\left(\frac{\mathbf{y}_{k+1} + \mathbf{y}_k}{2}, \mathbf{u}_{k+1/2} \right) & k = 0,1,\ldots,N-1 \\ \mathbf{h}(\mathbf{y}_0, \mathbf{y}_N, \mathbf{q}) & k = N \\ \mathbf{g}\left(\frac{\mathbf{y}_{k-N} + \mathbf{y}_{k-N+1}}{2}, \mathbf{u}_{k-N-1/2} \right) & k = N+1, N+2, \ldots, 2N \end{cases} \tag{13}$$

where $\mathbf{W} = (\mathbf{y}_0, \mathbf{y}_1 \ldots . \mathbf{y}_N, \mathbf{q}, \mathbf{u}_{1/2}, \mathbf{u}_{3/2}, \ldots, \mathbf{u}_{N-1/2})$

3.2 Elimination of control

Map (13) can be simplified if controls equations $\mathbf{g}\left(\frac{\mathbf{y}_{k-N} + \mathbf{y}_{k-N+1}}{2}, \mathbf{u}_{k-N-1/2} \right) = \mathbf{0}$ are solved separately. In particular we assume that we are able to obtain a function $\mathbf{u}(\mathbf{y})$ such that

$\mathbf{g}(\mathbf{y},\mathbf{u}(\mathbf{y}))=\mathbf{0}$. With $\mathbf{u}(\mathbf{y})$ we can construct the reduced map $\Psi(\mathbf{Z})$ derived from (13) as follows:

$$\Psi(\mathbf{Z})=\begin{cases}\mathbf{M}\left(\frac{\mathbf{y}_{k+1}+\mathbf{y}_k}{2}\right)\frac{\mathbf{y}_{k+1}-\mathbf{y}_k}{s_{k+1}-s_k}+\mathbf{n}\left(\frac{\mathbf{y}_{k+1}+\mathbf{y}_k}{2},\mathbf{u}\left(\frac{\mathbf{y}_{k+1}+\mathbf{y}_k}{2}\right)\right) & k=0,1,\ldots,N-1\\ \mathbf{h}(\mathbf{y}_0,\mathbf{y}_N.\mathbf{q}) & k=N\end{cases} \tag{14}$$

where $\mathbf{Z}=(\mathbf{y}_0,\mathbf{y}_1.\ldots.\mathbf{y}_N,\mathbf{q})$. The function $\mathbf{u}(\mathbf{y})$ when necessary is evaluated by solving the problem $\mathbf{g}(\mathbf{y},\mathbf{u}(\mathbf{y}))=\mathbf{0}$ while derivative of $\mathbf{u}(\mathbf{y})$ are evaluated by using derivative of $\mathbf{g}(\mathbf{y},\mathbf{u})$ and implicit function theorem.

3.3 The solution algorithm

As a solution algorithm a variant of Newton-Raphson scheme is used. In particular affine invariant Newton scheme[10] is used to solve (14). The scheme read as:

- Let $\mathbf{Z}^0$ assigned
- For $k=0,1,...$ until convergence

  $\mathbf{Z}^{k+1}=\mathbf{Z}^k-\alpha_k\mathbf{J}_k^{-1}\Psi(\mathbf{Z}^k)$

  where $\mathbf{J}_k=\frac{\partial\Psi}{\partial\mathbf{Z}}(\mathbf{Z}^k)$ and α_k is chosen to satisfy

  $\left\|\mathbf{J}_k^{-1}\Psi(\mathbf{Z}^{k+1})\right\|\le C\left(1-\frac{\alpha_k}{2}\right)\left\|\mathbf{J}_k^{-1}\Psi(\mathbf{Z}^k)\right\|$

the constant C in the original algorithm is 1 but this enforce monoticity and produce stagnation for large residual. We choose to use $C>>1$ when residual are large in order to speed up the convergence process.

4 MINIMUM LAP TIME PROBLEM FOR A RACING MOTORCYCLE

4.1 Motorcycle model and problem definition: racing competitions

The application of Optimal Control to racing engineering problems my produce large improvements in vehicle performance. Competition among teams is getting harder and harder, and technological innovation together with engineering and mechanic experience is a dominant factor. It is well known that the result of a racing vehicle is a combination of driver skills and vehicle performance[11]. However, in practice, it is very difficult to estimate such different contributions because they are deeply connected and influence each other. The OC gives the possibility to estimate the vehicle performance simply by driving it in the best possible way. Once the vehicle is always operated in the best way, one can carry out analysis on its design parameters to improve its intrinsic performance. Few examples are present in bibliography[12-13], but what makes the difference in this field is the sensitivity to parameters variations.

The methodology proposed above, applied to a motorcycle model turned out to be capable of reproducing important issue of the vehicle gross motion, despite its simplicity (when compared to other literature models[14]) as the comparison with telemetry witnesses[3]. In this work the same model is used to test the new formulation of the boundary condition and investigate their influence on motorcycle maneuvers. An example of a minimum lap time with is also presented.

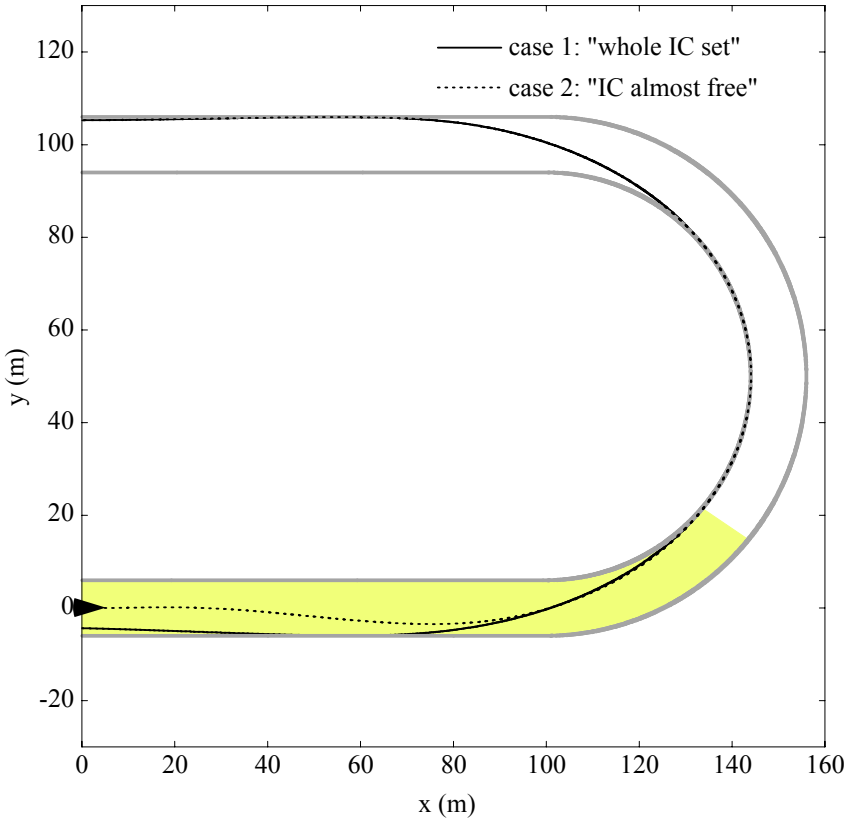

Figure 3: Influence of boundary conditions on motorcycle trajectories

4.2 The influence of boundary conditions: "U" shaped curve test

The initial and final boundary conditions can be fixed to a precise value or left free in order that the best conditions are found as solution (a desirable condition for simulating a real minimum lap time problem). The road geometry used in this test is a "U" shaped curve with curvature radius of 50m. This is a simple case that lets easily appreciate the influence of boundary conditions on motorcycle maneuvers. On this test two cases were considered: *case 1* that consider initial motorcycle motion completely fixed and *case 2* with initial motorcycle motion completely free (except vertical and longitudinal forces). Both had final motion

completely free. The following figures show that the maneuvers differ only on the first part of the U curve (as highlighted in figure in the grey area). Of course for *case 2* the best possible initial motion is found. Figure 3 compares the trajectories and shows that the initial lateral position of *case 2* is closer to the right road edge than *case 1*. Figure 4 shows that the initial lateral velocity of *case 2* is higher than *case 1* (which had it fixed). However being longitudinal forces for *case 2* fixed to zero, the initial velocity could have been higher if they were left free. Figure 5 and 6, shows respectively motorcycle roll and steering angles, and interestingly the so called "out tracking maneuver"[15] is not present in *case 2*. The reason is that the found initial motion is with the motorcycle already rolled. As a consequence it is not necessary to lean the motorcycle inside the curve with the usual stroke on the handle bar known as out tracking as did in case 1. The gain in term of time between the two cases is of 0.75s and is obtained completely on the first part of the curve for the reasons explained above.

4.3 Minimum lap time: the circuit of Adria

A minimum lap time on a real circuit was also solved. Computational mesh consists of 3010 points resulting in a non linear system of about 120000 equations. The Affine Invariant Newton scheme converged in 115 iterations, with an accuracy of 10^{-10} on scaled residual.

Figure 7 shows the computed motorcycle trajectory. As highlighted final position does not coincide with initial one since final boundary conditions were left free. Figure 8 shows motorcycle velocity for the whole track.

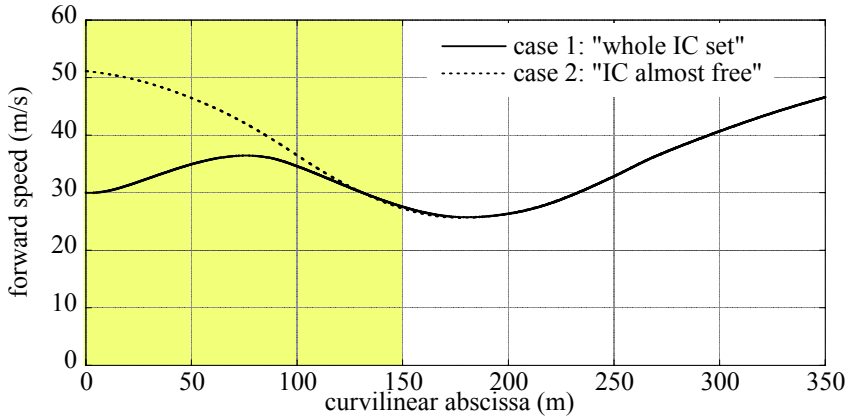

Figure 4: influence of boundary conditions on motorcycle velocity

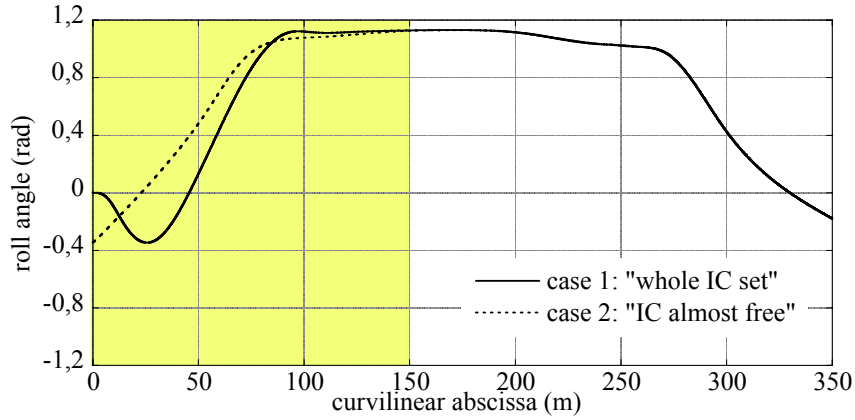

Figure 5: influence of boundary conditions on motorcycle roll angle

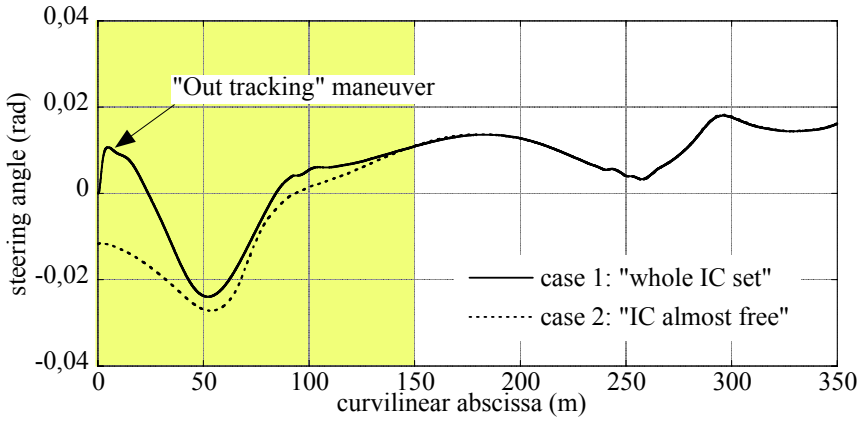

Figure 6: influence of boundary conditions on motorcycle steering angle

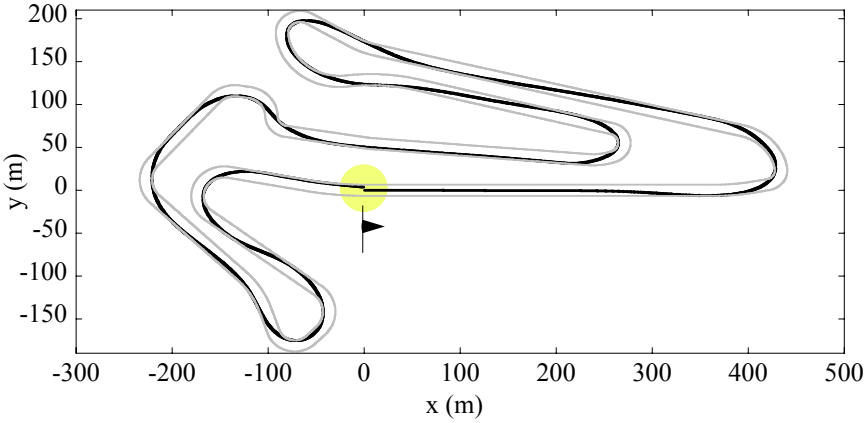

Figure 7: motorcycle trajectory on Adria circuit

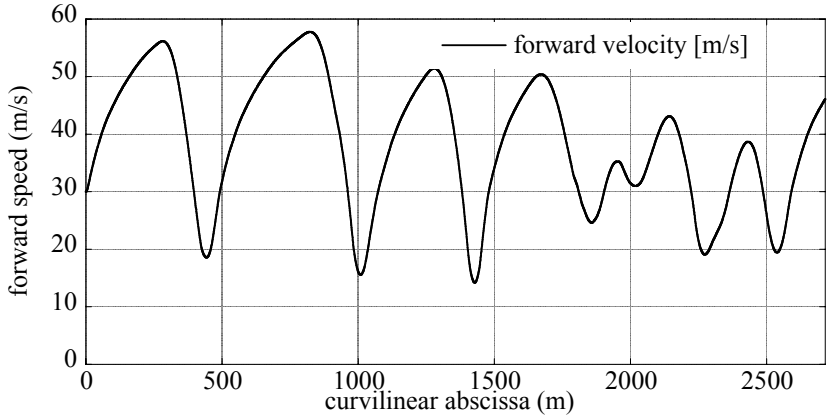

Figure 8: motorcycle velocity profile on Adria circuit

5 CONCLUSIONS

This paper presents a full direct approach technique for solving OCP applied to multibody and mechatronic systems. A collection of symbolic procedure were used to obtain the optimality necessary condition equations, included adjoint ones. In these procedures the particular structure of multibody system is exploited, to better handle problem with large number of equations. The resulting TPBVP is discretized by a finite difference scheme resulting nonlinear system of equations are automatically translated into high-level programming language along with their jacobian matrices. The solution algorithm is based on a modified version of the Affine Invariant Newton scheme. A new formulation for the boundary condition was introduced and it was intended to be further applied to a larger multibody system described with set of dependent coordinates.

Examples of the influence of boundary conditions on motorcycle maneuvers are shown on a U curve. A minimum lap time on a real circuit is also shown.

ACKNOWLEDGMENT

Methods for symbolic modelling of multibody systems described here have been developed partial with the financial support from the “Provincia Autonoma di Trento” (PAT) in the framework of project MEPROMEC (a research project for the years 2001-2004 aiming at the development of tools for modelling simulation and design of mechatronic systems).

REFERENCES

[1] L. Cervantes and L. T. Biegler, *Optimization strategies for dynamic systems*, in Encyclopedia of Optimization, C. Floudas and P.Pardalos, eds., vol. 4, Kluwer, 2001, pp. 216–227.

[2] R. W. H. Sargent, *Optimal control*, J. Comput. Appl. Math., 124 (2000), pp. 361–371. Numerical analysis 2000, Vol. IV, Optimization and nonlinear equations.
[3] E. Bertolazzi, F. Biral, M. Da Lio, *Symbolic-Numeric Efficient Solution of Optimal Control Problems for Multibody Systems*, submitted to Journal of Computational Methods in Science and Engineering, Collana scientifica DIMS n°22, 1-19 (2003).
[4] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: an SQP algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006 (electronic).
[5] R. Serbant and L. R. Petzold, *COOPT—a software package for optimal control of largescale differential-algebraic equation systems*, Math. Comput. Simulation, 56 (2001), pp. 187–203. Method of lines (Athens, GA, 1999).
[6] R. Bulirsch, E. Nerz, H. J. Pesch, and O. von Stryk, *Combining direct and indirect methods in optimal control: range maximization of a hang glider*, in Optimal control (Freiburg,1991), vol. 111 of Internat. Ser. Numer. Math., Birkh¨auser, Basel, 1993, pp. 273–288.
[7] J. L. Troutman, *Variational calculus and optimal control*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, second ed., 1996. With the assistance of William Hrusa, Optimization with elementary convexity.
[8] B. C. Fabien, *Numerical solution of constrained Optimal Control Problems with parameters*, Applied Mathematics and Computation, 80 (1996), pp. 43-62
[9] J. Garcia de Jalon and E. Bayo, *Kinematic and dynamic simulation of multibody systems*, Mechanical Engineering Series, Springer-Verlag, New York, 1994. The real-time challenge.
[10] P. Deuflhard, G. Heindl, *Affine invariant convergence theorems fro Newton's Method and extensions to related methods*, SIAM J. Numer. Anal. Vol. 16, 1979.
[11] Spackman, K.: *The Future of Formula One Driver Training*, Race Tech Magazine (Racecar Graphic Ltd, London, UK), Jan-Feb 2000.
[12] V. Cossalter, M. Da Lio, F. Biral, and L. Fabbri, *Evaluation of motorcycle manoeuvrability with the optimal manoeuvre method*, SAE Transactions - Journal of passenger cars, (2000). SAE paper 983022.
[13] D. Casanova, S. R. S., and P. Symonds, *Minimum time manoeuvring: The significance of yaw inertia*, Vehicle system dynamics, 34 (2000), pp. 77–115.
[14] V. Cossalter and R. Lot, *A motorcycle multi-body model for real time simulations based on the natural coordinates approach*, Vehicle System Dynamics, 37 (2002), pp. 423–447.
[15] Rice, R.S.: *Rider Skill Influences on Motorcycle Manoeuvring*, SAE paper No. 780312 (1978).

APPENDIX I: DERIVATION OF OPTIMALITY NECESSARY CONDITIONS AND BOUNDARY CONDITIONS

In this Appendix the optimality necessary conditions are derived. A lot had already been done in the past, and from this point of view there is nothing new. However, here the

multibody DAEs structure is exploited and more attention is give to the correct setting of boundary conditions.

Conventions are: lower case letter used for column vectors, upper case for matrix. Variables vector $\mathbf{x},\mathbf{u},\boldsymbol{\lambda}$ are function of s.

The OCP based on equations (1)—(4) and (6) formulated in a non-constrained functional is:

$$J\left[\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\rho}\right]=\boldsymbol{\mu}^T\mathbf{c}\left(\mathbf{x}_i,\mathbf{x}_f,\boldsymbol{\rho}\right)+q\left(\mathbf{x}_i\right)+q\left(\mathbf{x}_f\right)+\int_{s_i}^{s_f}\left[f_p(\mathbf{x},\mathbf{u})+\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\dot{\mathbf{x}}+\mathbf{b}\left(\mathbf{x},\mathbf{u}\right)\right]\right]ds \quad \text{(A1)}$$

where $q(\mathbf{x})$, is the penalties function applied to the initial and final state. Using the maps (9) to put the boundary conditions one obtains the following functional:

$$\begin{gathered}J\left[\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}\right]=\boldsymbol{\mu}_i\cdot\left(\mathbf{x}(s_i)-\mathbf{L}_i(\boldsymbol{\rho})\right)+\boldsymbol{\mu}_f\cdot\left(\mathbf{x}(s_f)-\mathbf{L}_f(\boldsymbol{\rho})\right)+q\left(\mathbf{x}(s_i)\right)+q\left(\mathbf{x}(s_f)\right)\\ \int_{s_i}^{s_f}\left[f_p(\mathbf{x},\mathbf{u})+\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\dot{\mathbf{x}}+\mathbf{b}\left(\mathbf{x},\mathbf{u}\right)\right]\right]ds\end{gathered} \quad \text{(A2)}$$

Performing the first variation and setting it to zero (stationary condition for the functional) variables and using the shortcut $\mathbf{z}_i=\mathbf{z}(s_i)$, $\mathbf{z}_f=\mathbf{z}(s_f)$ for a generic function $\mathbf{z}(s)$:

$$\begin{gathered}\delta J=\delta\boldsymbol{\mu}_i^T\left(\mathbf{x}_i-\mathbf{L}_i(\boldsymbol{\rho})\right)+\boldsymbol{\mu}_i^T\left(\delta\mathbf{x}_i-\frac{\partial\mathbf{L}_i(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\delta\boldsymbol{\rho}_i+\frac{\partial q\left(\mathbf{x}_i\right)}{\partial\mathbf{x}_i}\delta\mathbf{x}_i\right)\\ +\delta\boldsymbol{\mu}_f^T\left(\mathbf{x}_f-\mathbf{L}_f(\boldsymbol{\rho})\right)+\boldsymbol{\mu}_f^T\left(\delta\mathbf{x}_f-\frac{\partial\mathbf{L}_f(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\delta\boldsymbol{\rho}_f+\frac{\partial q\left(\mathbf{x}_f\right)}{\partial\mathbf{x}_f}\delta\mathbf{x}_f\right)\\ +\int_{s_i}^{s_f}\frac{\partial f_p(\mathbf{x},\mathbf{u})}{\partial\mathbf{x}}\delta\mathbf{x}ds+\int_{s_i}^{s_f}\frac{\partial f_p(\mathbf{x},\mathbf{u})}{\partial\mathbf{u}}\delta\mathbf{u}ds\\ +\int_{s_i}^{s_f}\delta\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\dot{\mathbf{x}}+\mathbf{b}\left(\mathbf{x},\mathbf{u}\right)\right]ds+\int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left[\left(\frac{\partial\mathbf{A}(\mathbf{x})}{\partial\mathbf{x}}\delta\mathbf{x}\right)\dot{\mathbf{x}}\right]ds+\int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\delta\dot{\mathbf{x}}\right]ds\\ +\int_{s_i}^{s_f}\left[\boldsymbol{\lambda}^T\left(\frac{\partial\mathbf{b}\left(\mathbf{x},\mathbf{u}\right)}{\partial\mathbf{x}}\delta\mathbf{x}\right)\right]ds+\int_{s_i}^{s_f}\left[\boldsymbol{\lambda}^T\left(\frac{\partial\mathbf{b}\left(\mathbf{x},\mathbf{u}\right)}{\partial\mathbf{u}}\delta\mathbf{u}\right)\right]ds=\mathbf{0}\end{gathered} \quad \text{(A4)}$$

Integrating by parts the following:

$$\begin{gathered}\int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\delta\dot{\mathbf{x}}\right]ds=\left[\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\delta\mathbf{x}\right]\right]_{s_i}^{s_f}-\int_{s_i}^{s_f}\left(\frac{d}{ds}\boldsymbol{\lambda}^T\mathbf{A}(\mathbf{x})\right)\delta\mathbf{x}ds\\ =\boldsymbol{\lambda}_f^T\mathbf{A}(\mathbf{x}_f)\delta\mathbf{x}_f-\boldsymbol{\lambda}_i^T\mathbf{A}(\mathbf{x}_i)\delta\mathbf{x}_i-\int_{s_i}^{s_f}\dot{\boldsymbol{\lambda}}^T\mathbf{A}(\mathbf{x})\delta\mathbf{x}ds-\int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left(\frac{\partial\mathbf{A}(\mathbf{x})}{\partial\mathbf{x}}\dot{\mathbf{x}}\right)\delta\mathbf{x}ds\end{gathered} \quad \text{(A5)}$$

Substituting and collecting the variations:

$$
\begin{aligned}
\delta J = {} & \delta\boldsymbol{\mu}_i^{\,T}\left(\mathbf{x}_i - \mathbf{L}_i(\boldsymbol{\rho})\right) + \delta\boldsymbol{\mu}_f^{\,T}\left(\mathbf{x}_f - \mathbf{L}_f(\boldsymbol{\rho})\right) \\
& + \left[-\boldsymbol{\lambda}_i^{\,T}\mathbf{A}(\mathbf{x}_i) + \boldsymbol{\mu}_i^{\,T} + \frac{\partial q(\mathbf{x}_i)}{\partial \mathbf{x}_i}\right]\delta\mathbf{x}_i + \left[\boldsymbol{\lambda}_f^{\,T}\mathbf{A}(\mathbf{x}_f) + \boldsymbol{\mu}_f^{\,T} + \frac{\partial q(\mathbf{x}_f)}{\partial \mathbf{x}_f}\right]\delta\mathbf{x}_f \\
& - \boldsymbol{\mu}_i^{\,T}\frac{\partial \mathbf{L}_i(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\delta\boldsymbol{\rho} - \boldsymbol{\mu}_f^{\,T}\frac{\partial \mathbf{L}_f(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\delta\boldsymbol{\rho} + \int_{s_i}^{s_f}\frac{\partial f_p(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\delta\mathbf{x}ds - \int_{s_i}^{s_f}\dot{\boldsymbol{\lambda}}^T\mathbf{A}(\mathbf{x})\delta\mathbf{x}ds \\
& + \int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left[\left(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial \mathbf{x}}\delta\mathbf{x}\right)\dot{\mathbf{x}}\right]ds - \int_{s_i}^{s_f}\boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}}\right)\delta\mathbf{x}ds + \int_{s_i}^{s_f}\left[\boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{b}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\delta\mathbf{x}\right)\right]ds \\
& + \int_{s_i}^{s_f}\left[\frac{\partial f_p(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{b}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right]\delta\mathbf{u}ds + \int_{s_i}^{s_f}\delta\boldsymbol{\lambda}^T\left[\mathbf{A}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{b}(\mathbf{x},\mathbf{u})\right]ds = \mathbf{0}
\end{aligned}
\tag{A6}
$$

where, it is possible to rewrite the following term in a more compact way, as indicated below:

$$
\begin{aligned}
& \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial \mathbf{x}}\delta\mathbf{x}\right)\dot{\mathbf{x}} - \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}}\right)\delta\mathbf{x} = \sum_{i,j,k}\lambda_i\dot{x}_j\left(\frac{\partial A_{ik}(\mathbf{x})}{\partial x_j}\delta x_k\right) - \sum_{i,j,k}\lambda_i\delta x_j\left(\frac{\partial A_{ik}(\mathbf{x})}{\partial x_j}\dot{x}_k\right) \\
& = \sum_{i,j,k}\left(\frac{\partial A_{ik}(\mathbf{x})}{\partial x_j} - \frac{\partial A_{ij}(\mathbf{x})}{\partial x_k}\right)\lambda_i\dot{x}_j\delta x_k = \sum_{i,j,k}T_{ik}(\mathbf{x},\boldsymbol{\lambda})\dot{x}_j\delta x_k = \mathbf{T}(\mathbf{x},\boldsymbol{\lambda})^T\dot{\mathbf{x}}\delta\mathbf{x}
\end{aligned}
\tag{A7}
$$

Finally the equations of necessary conditions are:

- the original multibody system of dynamics equations:

$$\mathbf{A}(\mathbf{x}(s))\dot{\mathbf{x}}(s) + \mathbf{b}\left(\mathbf{x}(s),\mathbf{u}(s)\right) = \mathbf{0} \tag{A8a}$$

- the adjoint (or co-state) euqations:

$$\mathbf{T}(\mathbf{x}(s),\boldsymbol{\lambda}(s))^T\dot{\mathbf{x}}(s) - \dot{\boldsymbol{\lambda}}^T(s)\mathbf{A}(\mathbf{x}(s)) + \boldsymbol{\lambda}^T(s)\frac{\partial \mathbf{b}\left(\mathbf{x}(s),\mathbf{u}(s)\right)}{\partial \mathbf{x}} + \frac{\partial f_p(\mathbf{x}(s),\mathbf{u}(s))}{\partial \mathbf{x}} = \mathbf{0} \tag{A8b}$$

- the optimal control $\mathbf{u}(s)$ equations:

$$\frac{\partial f_p(\mathbf{x}(s),\mathbf{u}(s))}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T(s)\frac{\partial \mathbf{b}\left(\mathbf{x}(s),\mathbf{u}(s)\right)}{\partial \mathbf{u}} = 0 \tag{A8c}$$

- the boundary condition equations:

$$\begin{pmatrix} \mathbf{x}_i - \mathbf{L}_i(\rho) \\ \mathbf{x}_f - \mathbf{L}_f(\rho) \\ \boldsymbol{\mu}_i - \mathbf{A}(\mathbf{x}_i)^T \boldsymbol{\lambda}_i + \dfrac{\partial q(\mathbf{x}_i)}{\partial \mathbf{x}_i}^T \\ \boldsymbol{\mu}_f + \mathbf{A}(\mathbf{x}_f)^T \boldsymbol{\lambda}_f + \dfrac{\partial q(\mathbf{x}_f)}{\partial \mathbf{x}_f}^T \\ \dfrac{\partial L_i(\rho)}{\partial \rho}^T \boldsymbol{\mu}_i + \dfrac{\partial L_f(\rho)}{\partial \rho}^T \boldsymbol{\mu}_f \end{pmatrix} = \mathbf{0} \qquad \text{(A8d)}$$

where: $c(\mathbf{x}_i, \mathbf{x}_f, \rho) = \begin{pmatrix} \mathbf{x}_i - \mathbf{L}_i(\rho) \\ \mathbf{x}_f - \mathbf{L}_f(\rho) \end{pmatrix}$

Solving the third and fourth rows of (A8d) with respect to $\boldsymbol{\mu}_i^T$, $\boldsymbol{\mu}_f^T$, it is possible to re-manipulate the boundary conditions as follow, eliminating the langrange multipliers $\boldsymbol{\mu}$

$$\mathbf{h}(\mathbf{y}_i, \mathbf{y}_f, \rho) = \begin{pmatrix} \mathbf{x}_i - \mathbf{L}_i(\rho) \\ \mathbf{x}_f - \mathbf{L}_f(\rho) \\ \dfrac{\partial L_i(\rho)}{\partial \rho}^T \left(\mathbf{A}(\mathbf{x}_i)^T \boldsymbol{\lambda}_i - \dfrac{\partial q(\mathbf{x}_i)}{\partial \mathbf{x}_i}^T \right) - \dfrac{\partial L_f(\rho)}{\partial \rho}^T \left(\mathbf{A}(\mathbf{x}_f)^T \boldsymbol{\lambda}_f + \dfrac{\partial q(\mathbf{x}_f)}{\partial \mathbf{x}_f}^T \right) \end{pmatrix} \qquad \text{(A8d)}$$