Technical Report

IRI-TR-21-02

Second Order Collocation

Enric celaya

December, 2021

Institut de Robòtica i Informàtica Industrial

Abstract

Collocation methods for optimal control commonly assume that the system dynamics is expressed as a first order ODE of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$, where $\mathbf{x}$ is the state and $\mathbf{u}$ the control vector. However, in many cases, the dynamics involve the second order derivatives of the coordinates: $\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}, t)$, so that, to preserve the first order form, the usual procedure is to introduce one velocity variable for each coordinate and define the state as $\mathbf{x} = [\mathbf{q}, \mathbf{v}]^\mathsf{T}$, where $\mathbf{q}$ and $\mathbf{v}$ are treated as independent variables. As a consequence, the resulting trajectories do not fulfill the mandatory relationship $\mathbf{v} = d\mathbf{q}/dt$ except at the collocation points, where it is explicitly imposed.

We propose a formulation for Trapezoidal and Hermite-Simpson collocation methods adapted to deal directly with second order dynamics without the need to introduce $\mathbf{v}$ as independent from $\mathbf{q}$, and granting the consistency of the trajectories for $\mathbf{q}$ and $\mathbf{v}$.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)
`http://www.iri.upc.edu`

Corresponding author:

E. Celaya
tel: +34 93 401 5787
`celaya@iri.upc.edu`
`http://www.iri.upc.edu/staff/celaya`

1 Introduction

Direct methods for the numerical solution of trajectory optimization problems involve the transcription of a continuous-domain optimal control problem into a finite-dimensional nonlinear programming (NLP) problem [2]. The transcription process consists in partitioning the time history of the control and state variables into N intervals delimited by $N+1$ knot points $t_i, i = 0 \dots N$. The system dynamics is introduced by imposing the dynamic equations at a set of M collocation points, which may coincide, or not, with the set of knot points. The cost function is approximated with a function of the values taken by the variables at the collocation points, and the NLP problem is then formulated using them. Once the NLP problem is solved for the discrete set of points, a continuous solution is built with an interpolating function that fulfills the system dynamics at all collocation points.

The general formulation of most collocations methods assumes that the system dynamics is expressed as a first order ODE of the form [4]:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{1}$$

where $\mathbf{x} \in \Re^n$ is the state vector and $\mathbf{u} \in \Re^m$ is the vector of control variables. However, in Robotics, as in Mechanics in general, the evolution of the system is commonly governed by a second order ODE for the generalized coordinates q_i and their first temporal derivatives $\dot{q}_i \equiv v_i$:

$$\frac{d^2\mathbf{q}}{dt^2} = \mathbf{g}(\mathbf{q}, \mathbf{v}, \mathbf{u}, t), \tag{2}$$

To cast the second order ODE (2) into the first order form (1), the state vector is defined by stacking the generalized coordinates and their temporal derivatives: $\mathbf{x} = [\mathbf{q}, \mathbf{v}]^\mathsf{T}$, and the dynamic equation becomes:

$$\frac{d\mathbf{x}}{dt} = \frac{d}{dt}\begin{bmatrix}\mathbf{q}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ \mathbf{g}(\mathbf{q}, \mathbf{v}, \mathbf{u}, t)\end{bmatrix}. \tag{3}$$

Formulated in this way, the number of variables of the NLP problem is increased from $(n+m)M$ to $(2n+m)M$ [3]. In addition, a consistency issue is raised: since $\mathbf{v}$ is treated as a variable independent of $\mathbf{q}$, their functional relationship is only granted at the collocation points, where it is explicitly imposed. When the temporal evolution of both $\mathbf{v}$ and $\mathbf{q}$ are modeled with polynomials of the same degree, the necessary condition

$$\mathbf{v} = \frac{d\mathbf{q}}{dt} \tag{4}$$

is not preserved in general. Not fulfilling (4) prevents the possibility to reach a correct solution, since, even if the control function $\mathbf{u}(t)$ produces the expected trajectory for $\mathbf{v}(t)$, its integration will not coincide with the function obtained for $\mathbf{q}(t)$, i.e., the state trajectory $\mathbf{x}(t)$ is inconsistent. In this work, we present modified versions of the trapezoidal and Hermite-Simpson collocation methods specifically addressed to second order systems, with the dynamics given by (2). The new formulation grants that condition (4) is fulfilled all along the trajectory.

2 First Order Dynamics Equations: Trapezoidal and Hermite-Simpson Collocation

Two of the most widely used collocation methods are Trapezoidal and Hermite-Simpson. Before developing the second order version of these methods, we briefly recall how they approximate the dynamics equations and build the interpolating polynomials for the state.

2.1 Trapezoidal collocation

In trapezoidal collocation, the state trajectories are approximated by quadratic polynomials. If, for each interval $[t_k, t_{k+1}]$, we define $\tau = t - t_k$, we can write the polynomial approximation for a component x of the state and its temporal derivative as:

$$x(\tau) = a\tau^2 + b\tau + c \tag{5}$$
$$\dot{x}(\tau) = 2a\tau + b. \tag{6}$$

The polynomial coefficients can be expressed it terms of three parameters: the value of the polynomial at the initial time of the interval and the values of its derivative at the initial and final times:

$$\begin{aligned} x(0) &= x_k \\ \dot{x}(0) &= \dot{x}_k \\ \dot{x}(h) &= \dot{x}_{k+1}, \end{aligned}$$

where $h = t_{k+1} - t_k$. Using (5) and (6), and solving for a, b, c, we get

$$\begin{aligned} a &= \frac{\dot{x}_{k+1} - \dot{x}_k}{2h} \\ b &= \dot{x}_k \\ c &= x_k. \end{aligned}$$

Substituting in (5) and taking $\tau = h$, we can express the value of x at the end of the interval as:

$$x(h) = x_k + \frac{h}{2}(\dot{x}_{k+1} + \dot{x}_k). \tag{7}$$

The collocation constraints impose that the dynamics equation (1) must hold at the knot points, so that

$$\begin{aligned} \dot{x}_k &= f(\mathbf{x}_k, \mathbf{u}_k, t_k) \equiv f_k \\ \dot{x}_{k+1} &= f(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, t_{k+1}) \equiv f_{k+1}, \end{aligned}$$

which, together with the continuity condition $x_{k+1} = x(h)$, results in the dynamics constraint

$$x_{k+1} = x_k + \frac{h}{2}(f_{k+1} + f_k), \tag{8}$$

which is the well known trapezoidal rule [1]. The interpolation formula (5) becomes

$$x(\tau) = x_k + f_k\tau + \frac{\tau^2}{2h}(f_{k+1} - f_k). \tag{9}$$

Continuity between polynomials of consecutive intervals is granted until the first derivative since, by construction, the value of $\dot{x}_{k+1} = f_{k+1}$ imposed at the end of interval k is taken as the initial value for interval $k + 1$.

2.2 Hermite-Simpson collocation

In Hermite-Simpson collocation, the state trajectories are approximated by cubic polynomials:

$$x(\tau) = a\tau^3 + b\tau^2 + c\tau + d, \tag{10}$$
$$\dot{x}(\tau) = 3a\tau^2 + 2b\tau + c. \tag{11}$$

In this case, the polynomial coefficients can be expressed in terms of four parameters: the values of the polynomial and its derivative at the initial and final times of the interval:

$$\begin{aligned} x(0) &= x_k \\ x(h) &= x_{k+1} \\ \dot{x}(0) &= \dot{x}_k \\ \dot{x}(h) &= \dot{x}_{k+1}. \end{aligned}$$

Introducing them in (10), (11), and solving for a, b, c, d, we get

$$\begin{aligned} a &= \frac{\dot{x}_{k+1} + \dot{x}_k}{h^2} - \frac{2(x_{k+1} - x_k)}{h^3} \\ b &= -\frac{\dot{x}_{k+1} + 2\dot{x}_k}{h} + \frac{3(x_{k+1} - x_k)}{h^2} \\ c &= \dot{x}_k \\ d &= x_k. \end{aligned}$$

In order to determine a fourth degree polynomial, four conditions have to be imposed, and the Hermite-Simpson collocation method makes this by fixing its value at the beginning of the interval and imposing the dynamics constraints at the two bounding knot points and the midpoint between them. The value of the polynomial and its derivative at the interval midpoint can be expressed in terms of the four parameters used above, that is, substituting a, b, c, d in (10) and (11), and taking $\tau = h/2$, we get, respectively:

$$x_c = \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}(\dot{x}_k - \dot{x}_{k+1}) \tag{12}$$

$$\dot{x}_c = \frac{3}{2h}(x_{k+1} - x_k) - \frac{1}{4}(\dot{x}_k + \dot{x}_{k+1}), \tag{13}$$

where $x_c = x(h/2)$.

The collocation constraints are introduced by imposing the fulfillment of the dynamics equations (1) at the three points:

$$\begin{aligned} \dot{x}_k &= f(\mathbf{x}_k, \mathbf{u}_k, t_k) \equiv f_k \\ \dot{x}_c &= f(\mathbf{x}_c, \mathbf{u}_c, t_c) \equiv f_c \\ \dot{x}_{k+1} &= f(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, t_{k+1}) \equiv f_{k+1}, \end{aligned}$$

so that substituting in equation (13), and isolating x_{k+1}, we obtain the dynamics constraint

$$x_{k+1} = x_k + \frac{h}{6}(f_k + 4f_c + f_{k+1}), \tag{14}$$

while the value at the midpoint is obtained from (12):

$$x_c = \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}(f_k - f_{k+1}).$$

Finally, the interpolation formula is obtained by substituting (14) into (10):

$$x(\tau) = x_k + f_k\tau - \frac{\tau^2}{2h}(3f_k - 4f_c + f_{k+1}) + \frac{\tau^3}{3h^2}(2f_k - 4f_c + 2f_{k+1}). \tag{15}$$

Note that, despite the polynomial approximation into each interval between consecutive knot points is of third degree, continuity through knots is only imposed on the state trajectory and its first derivative.

3 Modified Trapezoidal Collocation for 2^{nd} Order Dynamics

When the dynamics is governed by a second order system of the form

$$\frac{d^2\mathbf{q}}{dt^2} = \mathbf{g}(\mathbf{q}, \mathbf{v}, \mathbf{u}, t), \tag{16}$$

applying the same strategy as in the trapezoidal method will consist in imposing constraint (16) at each knot point. However, since in this case constraints are applied to the second derivative of the state, when $\mathbf{q}(t)$ is a second degree polynomial, its second derivative will be constant, and we could not impose two different collocation constraints, one at each interval bound. As a consequence, the state must be approximated by polynomials of degree 3 at least. So, we have

$$\begin{aligned} q(\tau) &= a\tau^3 + b\tau^2 + c\tau + d & (17)\\ \dot{q}(\tau) &= 3a\tau^2 + 2b\tau + c & (18)\\ \ddot{q}(\tau) &= 6a\tau + 2b. & (19) \end{aligned}$$

Since four parameters are required to determine a third degree polynomial, we will need a further condition in addition to the initial value and the two second derivatives at the interval bounds. Note that for a cubic polynomial, no more than two independent conditions can be fulfilled by its second derivative, so that imposing the dynamics at the midpoint of the interval as in the Hermite-Simpson method is not possible here. The obvious choice is clearly the value of the first derivative v_k at the initial point. Thus we will use as parameters:

$$\begin{aligned} q(0) &= q_k \\ \dot{q}(0) &= v_k \\ \ddot{q}(0) &= \ddot{q}_k \\ \ddot{q}(h) &= \ddot{q}_{k+1}. \end{aligned}$$

Substituting in (17), (18), (19) and solving for the coefficients a, b, c, d, we obtain the following expression for the interpolation polynomial (17):

$$q(\tau) = q_k + v_k\tau + \ddot{q}_k\frac{\tau^2}{2} + \frac{\tau^3}{6h}(\ddot{q}_{k+1} - \ddot{q}_k). \tag{20}$$

Taking $\tau = h$ we get:

$$q(h) = q_k + v_k h + \frac{h^2}{6}(\ddot{q}_{k+1} + 2\ddot{q}_k). \tag{21}$$

Imposing the dynamics equation (2) so that

$$\begin{aligned} \ddot{q}_k &= g(\mathbf{q}_k, \mathbf{v}_k, \mathbf{u}_k, t_k) \equiv g_k & (22)\\ \ddot{q}_{k+1} &= g(\mathbf{q}_{k+1}, \mathbf{v}_{k+1}, \mathbf{u}_{k+1}, t_{k+1}) \equiv g_{k+1}, & (23) \end{aligned}$$

and the continuity condition $q(h) = q_{k+1}$, we obtain the dynamics constraint

$$q_{k+1} = q_k + v_k h + \frac{h^2}{6}(g_{k+1} + 2g_k). \tag{24}$$

Similarly, substituting the obtained coefficients a, b, c in (18) and taking $\tau = h$, we get

$$\dot{q}(h) = v_k + \frac{h}{2}(\ddot{q}_{k+1} + \ddot{q}_k), \tag{25}$$

so that using $\dot{q}(h) = v_{k+1}$ and the dynamics constraints (22)-(23), we obtain

$$v_{k+1} = v_k + \frac{h}{2}(g_{k+1} + g_k), \tag{26}$$

which is, in fact, the trapezoidal rule, that in this case is only valid for the velocity, but not for the state itself, that is given by (24). Finally, the interpolation function (20) is written as

$$q(\tau) = q_k + v_k\tau + \frac{\tau^2}{2}g_k + \frac{\tau^3}{6h}(g_{k+1} - g_k). \tag{27}$$

It is worth noting that, here, the continuity between polynomials at knot points is of second order, since the collocation constraints impose the coincidence of the second derivative of $q(t)$.

4 Modified Hermite-Simpson Collocation for 2^{nd} Order Dynamics

Our purpose here is to impose the second order dynamics on the two knots and the midpoint of each interval, in similarity with the conventional Hermite-Simpson method. Clearly, if we want to impose three conditions to the second derivative of a polynomial, it must be of degree 2 at least, what implies that the degree of the polynomials approximating the configuration must be at least 4. Thus, we propose to approximate $q(\tau)$, and its derivatives, by

$$\begin{aligned} q(\tau) &= a\tau^4 + b\tau^3 + c\tau^2 + d\tau + e && (28)\\ \dot{q}(\tau) &= 4a\tau^3 + 3b\tau^2 + 2c\tau + d && (29)\\ \ddot{q}(\tau) &= 12a\tau^2 + 6b\tau + 2c. && (30) \end{aligned}$$

Since 5 parameters are needed to determine the 5 coefficients of the polynomial, we will use, in addition to the three accelerations $\ddot{q}_k, \ddot{q}_c, \ddot{q}_{k+1}$, the values of the state q_k and its derivative v_k at the initial point:

$$\begin{aligned} q(0) &= q_k\\ \dot{q}(0) &= v_k\\ \ddot{q}(0) &= \ddot{q}_k\\ \ddot{q}(h/2) &= \ddot{q}_c\\ \ddot{q}(h) &= \ddot{q}_{k+1}. \end{aligned}$$

Solving for the coefficients, we obtain the following expression for the interpolating polynomial:

$$q(\tau) = q_k + v_k\tau + \frac{\tau^2}{2}\ddot{q}_k + \frac{\tau^3}{6h}(-3\ddot{q}_k + 4\ddot{q}_c - \ddot{q}_{k+1}) + \frac{\tau^4}{6h^2}(\ddot{q}_k - 2\ddot{q}_c + \ddot{q}_{k+1}). \tag{31}$$

Evaluating this expression for $\tau = h$ and $\tau = h/2$ we get, respectively:

$$\begin{aligned} q_{k+1} &= q_k + v_k h + \frac{h^2}{6}(\ddot{q}_k + 2\ddot{q}_c) && (32)\\ q_c &= q_k + \frac{h}{2}v_k + \frac{h^2}{96}(7\ddot{q}_k + 6\ddot{q}_c - \ddot{q}_{k+1}), && (33) \end{aligned}$$

and imposing (2), so that

$$\begin{aligned} \ddot{q}_k &= g(\mathbf{q}_k, \mathbf{v}_k, \mathbf{u}_k, t_k) \equiv g_k\\ \ddot{q}_c &= g(\mathbf{q}_c, \mathbf{v}_c, \mathbf{u}_c, t_c) \equiv g_c\\ \ddot{q}_{k+1} &= g(\mathbf{q}_{k+1}, \mathbf{v}_{k+1}, \mathbf{u}_{k+1}, t_{k+1}) \equiv g_{k+1}, \end{aligned}$$

we obtain the dynamics constraint:

$$q_{k+1} = q_k + v_k h + \frac{h^2}{6}(g_k + 2g_c), \tag{34}$$

with

$$q_c = q_k + \frac{h}{2}v_k + \frac{h^2}{96}(7g_k + 6g_c - g_{k+1}). \tag{35}$$

The expression for $v_{k+1} = \dot{q}(h)$ is obtained by substituting $\tau = h$ in (29):

$$v_{k+1} = v_k + \frac{h}{6}(g_k + 4g_c + g_{k+1}), \tag{36}$$

which we can recognize as the Simpson quadrature formula. An expression for $v_c = \dot{q}(h/2)$ is obtained by substituting $\tau = h/2$ in (29):

$$v_c = v_k + \frac{h}{24}(5g_k + 8g_c - g_{k+1}), \tag{37}$$

however, since q_c and v_c are to be used in the evaluation of g_c, we want to avoid expressing them in terms of g_c. For this, we isolate g_c from (36) and substitute the result in (35) and (37), respectively, to yield:

$$q_c = q_k + \frac{h}{32}(13v_k + 3v_{k+1}) + \frac{h^2}{192}(11g_k - 5g_{k+1}) \tag{38}$$

$$v_c = \frac{1}{2}(v_k + v_{k+1}) + \frac{h}{8}(g_k - g_{k+1}). \tag{39}$$

Written in this form, they can be used to formulate the problem in compressed form, i.e., eliminating the need to introduce variables q_c, v_c, g_c. The interpolating polynomial (31) becomes:

$$q(\tau) = q_k + v_k\tau + \frac{\tau^2}{2}g_k + \frac{\tau^3}{6h}(-3g_k + 4g_c - g_{k+1}) + \frac{\tau^4}{6h^2}(g_k - 2g_c + g_{k+1}). \tag{40}$$

In this case, the continuity across knot points is also of second order due to the coincidence of the second derivative imposed by the collocation constraints.

5 Test Cases

The performance of the proposed 2nd order methods are next evaluated and compared with the corresponding usual 1st order ones. For the ease of comparison, two examples proposed in [2] will be used, namely, the block-move and the cart-pole swing-up problems.

5.1 Block-Move problem

This problem consists in finding how to move a block of mass m between two points in a horizontal plane without friction, starting and finishing at rest, in a fixed amount of time T, so as to minimize the cost function defined as the integral of control effort squared:

$$J = \int_0^T u^2(\tau)d\tau, \tag{41}$$

where $u(t)$ is the force applied to the block. The analytic solution for this problem is easily derived and can be used to compare the results with the true optimal trajectory, which, for a total time of 1s and mass $m = 1$, is given by:

$$\begin{aligned} u(t) &= 6 - 12t \\ x(t) &= 3t^2 - 2t^3, \end{aligned}$$

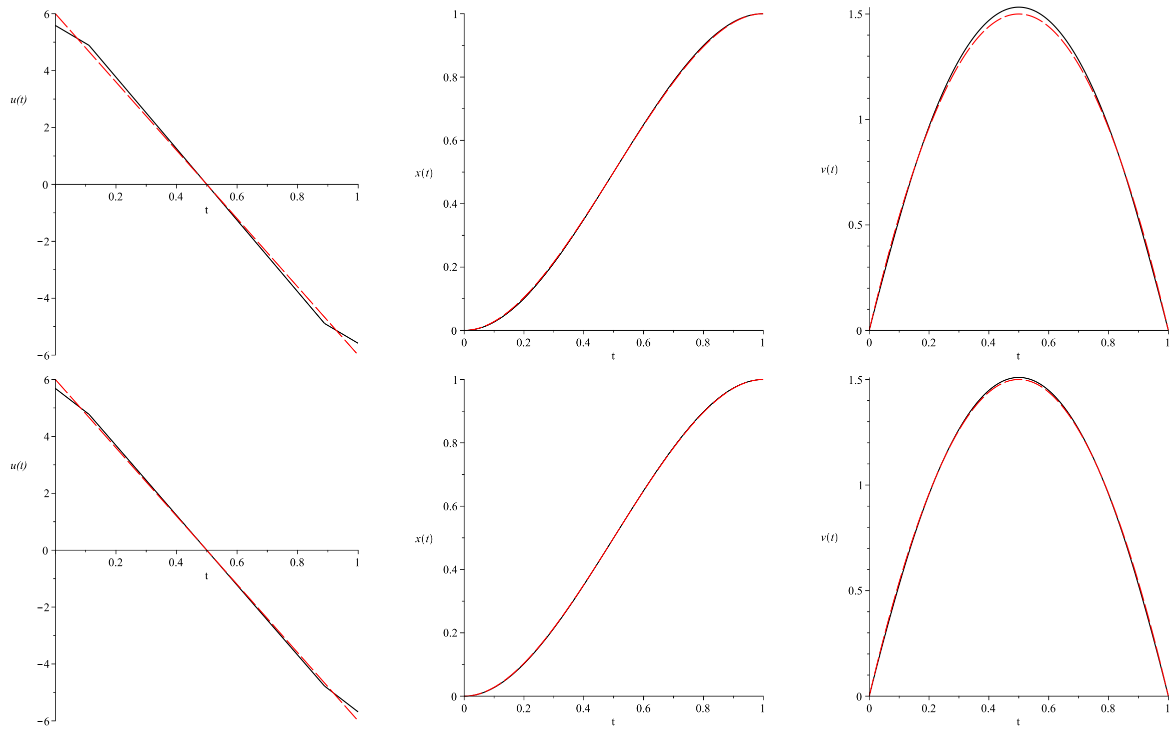

Figure 1: Block-Move trajectories. Top row: regular trapezoidal collocation. Bottom row: modified trapezoidal. Dashed red line shows the exact solution.

with a total cost for the trajectory $J = 12.0$. Comparing the results obtained by the 1st and 2nd order trapezoidal methods with 10 collocation points, we get an improvement in the total cost from 12.569 for the first to 12.281 for the second, so that the difference with the optimal value is reduced to the half. Figure 1 shows the trajectories obtained by each method comparing them with the exact solution, evidencing the improved accuracy of the 2nd order method.

For this specific problem, the first order Hermite-Simpson method already provides the exact optimal solution, since it approximates the control trajectory $u(t)$ linearly and the trajectory $x(t)$ with a cubic spline, which are the right degrees for them. Thus, the comparison between the results of the original and modified Hermite-Simpson methods shows no difference between them.

5.2 Cart-Pole Swing-Up problem

The cart-pole system comprises a cart that travels along a horizontal track and a pendulum that hangs freely from the cart. A motor drives the cart forward and backward along the track. Starting with the pendulum hanging below the cart at rest at a given position, the goal is to reach a final configuration in a certain amount of time T with the pendulum stabilized at a point of inverted balance and the cart staying at rest at a distance d from the initial position, minimizing the cost function J given by (41).

We will compare the performance of the 1st and 2nd order Hermite-Simpson collocation methods on this problem. The dynamic equations and model parameters are taken from [2]. The solution presented here has a cost $J = 57.939$ for the 1st order method and $J = 57.925$ for the 2nd order one, and corresponds to a different local minimum than that given in [2], whose cost is a little worst: $J = 58.805$. Figure 2 shows the trajectories obtained in both approaches, which are practically identical except for minor deviations visible near the extrema of the control trajectory $u(t)$.

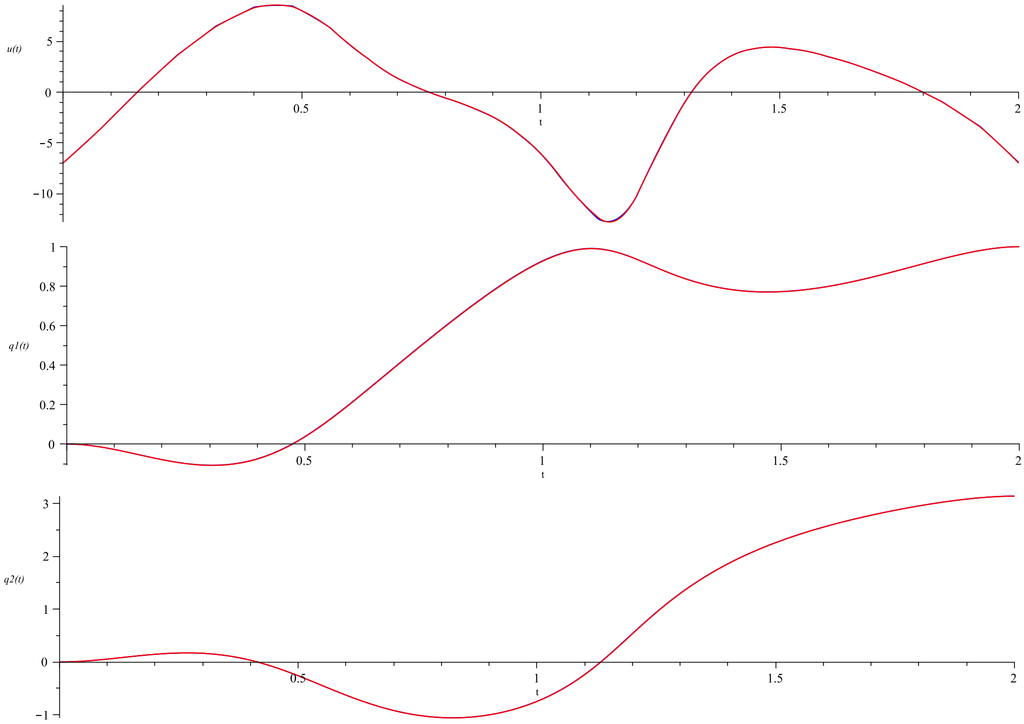

Figure 2: Control and state trajectories for Cart-Pole Swing-Up problem. Blue: 1st order Hermite-Simpson. Red: 2nd order Hermite-Simpson.

In this case, the optimal solution is not available, so that, to make a comparison, we will compute the dynamic error produced by each method. For a second order system governed by (2) we define second order dynamic error for component i of the state as $\epsilon(t) = \ddot{q}_i(t) - g_i(\mathbf{q}, \mathbf{v}, \mathbf{u}, t)$, which is more relevant than the first order dynamic errors used in [2]. Figure 3 compares the second order dynamic errors of both approaches. It can be appreciated that the error is significantly smaller for the 2nd order method. Also note that the dynamic error at the collocation points is exactly 0, something that is not satisfied by the 1st order method. This is due to the fact that the 1st order method does not grant the consistency between the generalized coordinates and their derivatives (eq. (4)).

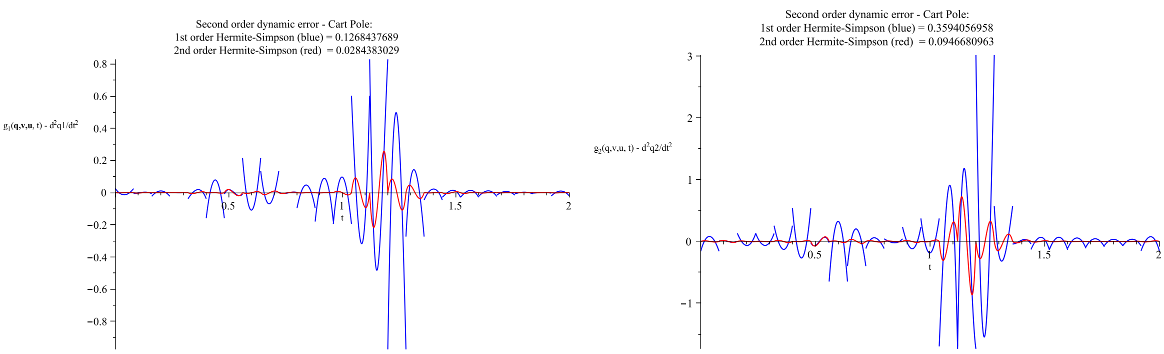

Figure 3: Second order dynamic error for Cart-Pole Swing-Up problem.

The same results are presented in Figure 4, showing the integration of the absolute value of the second order dynamic error along each interval.

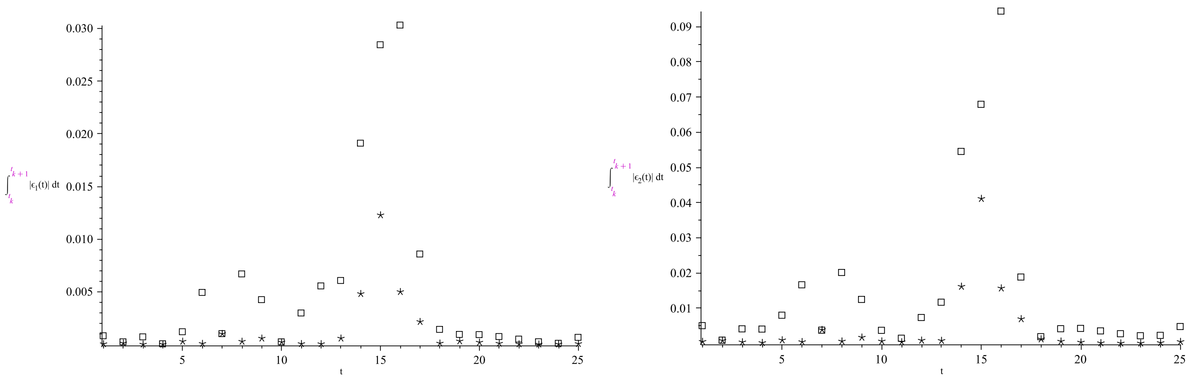

Figure 4: Integrated absolute second order dynamic error for Cart-Pole Swing-Up problem. Boxes: 1st order method. Asterisks: 2nd order method.

6 Conclusions

Trapezoidal and Hermite-Simpson collocation methods are very popular in the robotics community. However, they disregard the fact that the dynamics are most often second order. Directly imposing the second order constraints at the same collocation points as the original algorithms requires increasing the degree of the polynomials for coordinates approximation while keeping the same degree for velocity and actuation approximations. The proposed second order algorithms grant the concordance between the coordinates and their velocities not only at the collocation points, but all along the trajectory.

In the case of second order systems, the first order dynamic errors provided i.e. in [2], comparing the velocity trajectories with those of the coordinate derivatives, give an incomplete view of the accuracy of the results. Such dynamic error is eliminated with the methods proposed in this work. A more relevant performance measure, in this case, is the difference between the acceleration and the second derivative of the coordinate trajectories, since this is, in fact, what the optimization problem tries to minimize. Test cases comparing the second order methods with the first order ones show the improvement obtained in the second order dynamic error as well as in the total cost of the resulting trajectories.

References

[1] J.T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* SIAM, Philadelphia, PA, 2010.

[2] M. Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 1996.

[3] I. Michael Ross, J. Rea, and F. Fahroo. Exploiting higher-order derivatives in computational optimal control. In *In Proc. 10th Mediterranean conference on control and Automation - MED2002*, 2002.

[4] R. Tedrake. *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832).* Downloaded on november 2021 from http://underactuated.mit.edu/, 2021.

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation through the project PID2020-117509GB-I00.

IRI reports

This report is in the series of IRI technical reports.
All IRI technical reports are available for download at the IRI website
`http://www.iri.upc.edu`.