**Article:**

Nie, Y. orcid.org/0000-0002-0688-413X and Kerrigan, E.C. orcid.org/0000-0002-3967-1544 (2025) Reliable solution to dynamic optimization problems using integrated residual regularized direct collocation. IEEE Control Systems Letters, 9. pp. 1063-1068. ISSN: 2475-1456

https://doi.org/10.1109/lcsys.2025.3580771

# Reliable Solution to Dynamic Optimization Problems using Integrated Residual Regularized Direct Collocation

Yuanbo Nie[1] and Eric C. Kerrigan[2]

***Abstract*— Direct collocation (DC) is a widely used method for solving dynamic optimization problems (DOPs), but its implementation simplicity and computational efficiency are limited for challenging problems. For DOPs involving singular arcs, DC solutions often exhibit significant fluctuations along the singular arc, accompanied by large residual errors between collocation points, where the dynamic constraints are enforced as equality constraints. In this paper, we introduce the direct transcription method of integrated residual regularized direct collocation (IRRDC). This approach enforces dynamic constraints using a combination of point-wise residual constraints (expressed as either equalities or inequalities) and a penalty term on the integrated residual error, which helps reduce errors between collocation points. IRRDC retains the implementation simplicity of DC while improving both solution accuracy and efficiency, particularly for challenging problem types. Through the examples, we demonstrate that for problems where traditional DC results in excessive fluctuations, IRRDC effectively suppresses fluctuations and yields solutions with greater accuracy — at least two orders of magnitude lower in various error measures in relation to the dynamic and path constraints.**

## I. Introduction

Optimization problems involving dynamical systems are crucial aspects of engineering. For practical dynamic optimization problems (DOPs), numerical approaches are commonly employed. The direct transcription method is widely used to convert the infinite-dimensional problems into finite-dimensional nonlinear programming problems (NLPs), known as the transcription process.

For DOPs in model predictive control (MPC), the transcription method of multiple shooting is very popular [1]. For longer horizon optimal control problems and DOPs with highly nonlinear path constraints, the transcription method of the direct collocation (DC) method is generally considered to be preferable thanks to its well-rounded balance between computational complexity and solution accuracy [2].

The direct collocation (DC) framework enforces dynamic constraints at selected points on a discretized mesh. Efficient software toolboxes [3], [4] have been developed based on this approach and are widely adopted within the engineering field. However, certain categories of DOPs, particularly those involving singular arcs and high-index differential-algebraic equations (DAEs), present challenges for DC [2, Sec 4.14].

The development of integrated residual methods (IRM), such as the Quadrature Penalty Method (QPM) [5]–[7] and Direct Alternating Integrated Residuals (DAIR) [8], present alternatives to traditional DC by focusing on an integrated residual error (IRE) rather than point-wise residual measures. IRMs tend to handle challenging problems more reliably. By reliability, we refer to the ability to obtain solutions without prior knowledge of potential problem difficulties, e.g. reducing the likelihood of spurious solutions with large fluctuations. However, this increased flexibility comes at a cost: IRMs typically require greater expertise and care to configure the transcription process appropriately.

Users familiar with DC may find it challenging to adopt IRM effectively. For instance, both QPM and DAIR can produce solutions that deviate significantly from those reported in the literature if the trade-offs between competing dynamic constraints are not properly specified. As a result, DC remains the go-to numerical method for solving a general DOP with limited prior knowledge.

Starting with an overview of existing approaches in Section II, this paper investigates how the core framework of DC can be largely preserved while integrating insights and techniques from IRM to better handle challenging problems. We introduce this hybrid approach, termed integrated residual regularized direct collocation (IRRDC), in Section III. In IRRDC, dynamic constraints are enforced through a combination of explicit point-wise constraints, inherited from DC, and a penalty term on the IRE, inspired by IRMs. We also provide a rationale for why this seemingly simple combination yields practical benefits. Section IV presents numerical examples that demonstrate the advantages of the proposed method.

## II. Numerical Solution of Dynamic Optimization

Optimization-based control often requires the solution of DOPs expressed in the general Bolza form:

$$\min_{\substack{x(\cdot),u(\cdot)\\t_0,t_f}} \; V(x(t_0),x(t_f),t_0,t_f)+\int_{t_0}^{t_f} \ell(x(t),u(t),t)\mathrm{d}t \quad \text{(1a)}$$

subject to

$$f(\dot{x}(t),x(t),u(t),t)=0, \quad \forall t\in[t_0,t_f], \quad \text{(1b)}$$

$$g(\dot{x}(t),x(t),u(t),t)\leq 0, \quad \forall t\in[t_0,t_f], \quad \text{(1c)}$$

$$b(x(t_0),x(t_f),u(t_0),u(t_f),t_0,t_f)\leq 0, \quad \text{(1d)}$$

with the state trajectory of the system $x:\mathbb{R}\rightarrow\mathbb{R}^n$ and the control input trajectory $u:\mathbb{R}\rightarrow\mathbb{R}^m$,. The initial and terminal time are $t_0$ and $t_f$. Other elements include the Mayer cost $V$, the Lagrange cost $\ell$, the dynamic constraint $f$ containing $n_f$ equality constraints, the path constraint $g$

[1]Yuanbo Nie is with the School of Electrical & Electronic Engineering, University of Sheffield, S1 3JD Sheffield, U.K. y.nie@sheffield.ac.uk

[2]Eric C. Kerrigan is with the Department of Electrical & Electronic Engineering and Department of Aeronautics, Imperial College London, London SW7 2AZ, U.K., e.kerrigan@imperial.ac.uk

containing $n_g$ inequality constraints, and the boundary constraint $b$.

### A. Direct Transcription Method

To yield a practical approach to solve the infinite-dimensional continuous-time DOP numerically, direct transcription methods transcribe the DOP into finite-dimensional NLPs. In this transcription process, the state and input trajectories $x(\cdot)$ and $u(\cdot)$ are approximated by parameterised approximation functions $\tilde{x}(\cdot)$ and $\tilde{u}(\cdot)$, respectively. A popular choice [2] is to define $\tilde{x}(\cdot)$ and $\tilde{u}(\cdot)$ as piece-wise functions based on a time mesh with $K$ intervals, with the $k^{\text{th}}$ interval denoted as $\mathbb{T}_k := [s_k, s_{k+1}]$. Time instances $s_k$ represent the interval boundaries between intervals of the time mesh $t_0 = s_1 < \cdots < s_{K+1} = t_f$.

Under this setting, the trajectory for state variables inside each interval $k$ can be approximated as

$$x^{(k)}(t) \approx \tilde{x}^{(k)}(t) := \sum_{i=1}^{N^{(k)}} \chi_i^{(k)} \beta_i^{(k)}(t), \tag{2}$$

with $\beta_i^{(k)}(\cdot)$ a basis function and $\chi_i^{(k)}$ the corresponding coefficient. A commonly adopted approach — also used in this work — is to employ interpolating polynomials as basis functions. A key advantage of this choice is that there exist $N^{(k)}$ time instances $s_k \leq d_1^{(k)} < \cdots < d_{N^{(k)}}^{(k)} \leq s_{k+1}$ at which $\chi_i^{(k)} = \tilde{x}^{(k)}(d_i^{(k)}) \in \mathbb{R}^n$ for all $i \in \mathbb{I}_{N^{(k)}}$, with the index set defined as $\mathbb{I}_{N^{(k)}} = \{1, \ldots, N^{(k)}\}$. In this case, the unknown coefficients $\chi_i^{(k)}$ are sampled points of the approximate state trajectory, enabling convenient implementation. The corresponding time instances $d_i^{(k)}$ are referred to as the *data sampling instances*. The input can be parameterized similarly with $\upsilon_i^{(k)} = \tilde{u}^{(k)}\left(d_i^{(k)}\right) \in \mathbb{R}^m$. Additionally, we let $\chi := [\chi^{(1)}, \ldots, \chi^{(K)}]^\top$, $\chi^{(k)} = [\chi_1^{(k)}, \ldots, \chi_{N^{(k)}}^{(k)}]^\top$, $\upsilon := [\upsilon^{(1)}, \ldots, \upsilon^{(K)}]^\top$, $\upsilon^{(k)} = [\upsilon_1^{(k)}, \ldots, \upsilon_{N^{(k)}}^{(k)}]^\top$ for ease of presentation. As $\chi^{(k)}$ and $\upsilon^{(k)}$ fully determine $\tilde{x}^{(k)}(\cdot)$ and $\tilde{u}^{(k)}(\cdot)$, they also define the values of the approximation functions at instances beyond the data sampling instances.

A general formulation for the NLP with static decision variables $(\chi, \upsilon, t_0, t_f)$ is therefore:

$$\min_{\chi,\upsilon,t_0,t_f} V\left(\chi_1^{(1)}, \chi_{N^{(K)}}^{(K)}, t_0, t_f\right) + \sum_{k=1}^{K} \sum_{i=1}^{Q^{(k)}} w_i^{(k)} \ell\left(\tilde{x}^{(k)}(q_i^{(k)}), \tilde{u}^{(k)}(q_i^{(k)}), q_i^{(k)}\right) \tag{3a}$$

subject to, for all $k \in \mathbb{I}_K$,

$$\psi\left(\chi^{(k)}, \upsilon^{(k)}, t_0, t_f\right) = 0, \tag{3b}$$

$$\gamma\left(\chi^{(k)}, \upsilon^{(k)}, t_0, t_f\right) \leq 0, \tag{3c}$$

and for some $k_i \in \mathbb{I}_K$ and $k_j \in \mathbb{I}_K$,

$$\phi\left(\chi^{(k_i)}, \chi^{(k_j)}, \upsilon^{(k_i)}, \upsilon^{(k_j)}, t_0, t_f\right) \leq 0. \tag{3d}$$

The functions $\psi$, $\gamma$ and $\phi$ contain discretized forms of $f$, $g$ and $b$, along with other transcription-specific constraints. For instance, continuity constraints for the state and input trajectories could also be incorporated into (3d). The formulation of (3b)–(3d) may vary depending on the transcription scheme used, with further details provided in next subsection for the handling of dynamic constraints. To approximate the integral of the Lagrange cost inside an interval $k$, numerical integration with a $Q^{(k)}$-point quadrature rule [2, Sec. 4.4] is used, with *quadrature weights* $w_i^{(k)}$ and *quadrature abscissae* $q_i^{(k)}$, for all $i \in \mathbb{I}_{Q^{(k)}}$.

Convergence of the NLP (3) to a solution does not fully resolve the numerical solution of the DOP (1). This is because, regardless of the chosen direct transcription method, the initial discretization mesh may not be suitable for achieving the desired accuracy. As a result, mesh refinement (MR) may be required as a subsequent step, based on an assessment of the errors in the approximate solution obtained from the NLP.

For a given NLP decision variable tuple $(\chi,\upsilon,t_0,t_f)$, which fully determines the approximate solution trajectory tuple $(\tilde{x}(\cdot),\tilde{u}(\cdot),t_0,t_f)$, commonly used error measures include

$$\eta := \max_{k\in\mathbb{I}_{N^{(k)}},\, \xi\in\mathbb{I}_{n_f}} \int_{\mathbb{T}_k} |r_\xi^{(k)}(t)| \, \mathrm{d}t, \tag{4}$$

$$\sigma := \max_{t\in[t_0,t_f],\, \zeta\in\mathbb{I}_{n_g}} g_\zeta(\dot{\tilde{x}}(t), \tilde{x}(t), \tilde{u}(t), t), \tag{5}$$

with $r_\xi^{(k)}$ the $\xi$th element in the residual (also referred to as the residual error)

$$r^{(k)}(t) := f(\dot{\tilde{x}}^{(k)}(t), \tilde{x}^{(k)}(t), \tilde{u}^{(k)}(t), t), \tag{6}$$

and $g_\zeta$ the $\zeta$th element in (1c). In this definition, the dependencies of $r^{(k)}$ on other variables are omitted for simplicity in notation. Measures (4) and (5) are referred to as the maximum *absolute local (AL) error* and the maximum *constraint violation (CV) error* respectively. A scaled version of (4) is known as the maximum *relative local (RL) error* [2]. Other forms of IRE exist, such as the mean integrated residual norm square error (MIRNS), and the mean integrated residual square error (MIRS), as discussed in [8]. These error measures provide valuable insights into the necessary modifications to the discretization mesh for improvements, such as guiding MR schemes to add mesh nodes at locations beneficial for error reduction [2].

### B. Handling of Dynamic Constraint

Both DC and IRM embed the approximate satisfaction of the dynamic constraints (1b) within the formulation of the NLP; however, the formulation details differ.

*1) Dynamic relationships as equality constraints:* From the weighted residuals method for the numerical solution of differential equations [9], the satisfaction of the dynamic equations requires

$$\int_{\mathbb{T}_k} \varpi^{(k)}(t) r_\xi^{(k)}(t) \, \mathrm{d}t = 0, \quad \forall \xi \in \mathbb{I}_{n_f}, \tag{7}$$

to hold for all *test functions* $\varpi^{(k)}$ that make (7) integrable, which can only be true if the residual is zero. In practice, the goal is to select finite sets of test functions that would approximately solve the differential equation, with different choices of test functions leading to various popular weighted residual methods, including the Galerkin, collocation, and least-squares methods. These different choices result in varying accuracy and computational complexity characteristics, making the selection of test functions problem-specific [9].

The collocation method uses Dirac delta functions as test functions $\varpi^{(k)}$ in (7). With the isolation property of Dirac delta functions [8], (7) will become a system of equations that requires the local residual (6) to be zero at the data sampling instances, also known as the *collocation points*. In DC, the equation system obtained from the weighted residual method of collocation is implemented in (3b) as

$$f\left(\dot{\tilde{x}}^{(k)}(d_i^{(k)}), \tilde{x}^{(k)}(d_i^{(k)}), \tilde{u}^{(k)}(d_i^{(k)}), d_i^{(k)}\right) = 0, \quad (8)$$

for all $i \in \mathbb{I}_{N^{(k)}}$, $k \in \mathbb{I}_K$. Such a formulation can be made computationally efficient, because the weighted residual integration (7) is not needed, and the derivative information can be easily computed. This aspect contributes to the popularity of the DC approach in developing general-purpose DOP solvers — the benefit of using a more sophisticated setting, e.g. the Galerkin method, can only be exploited if the nature of the dynamical system is known beforehand.

*2) Dynamic relationships as penalty terms or as inequality constraints:* The drawback of only forcing the residual to be zero at the collocation points is that large errors could still arise between the collocation points [8]. In the implementation of the DOP solution, the trajectories are forward integrated, leading to the accumulation of errors. This accumulation results in constraint violations and deterioration in solution optimality.

The IRM addresses the challenge by working with the integrated form of the residuals, namely

$$\hat{R}^{(k)}(\tilde{x}^{(k)}, \tilde{u}^{(k)}, t_0, t_f) := \int_{\mathbb{T}_k} \|\alpha \circ r^{(k)}(t)\|_2^2 \, \mathrm{d}t. \quad (9)$$

with $\circ$ representing the Hadamard product (component-wise multiplication), and $\alpha \in \mathbb{R}^{n_f}$ being additional weighting parameters, e.g. to account for differences in the numerical range of variables and constraints. In practice, this integration could be approximated by a quadrature rule similar to the discrete approximation of the Lagrange cost in (3a), as

$$\mathcal{R}\left(\chi, \upsilon, t_0, t_f, w, q\right) := \sum_{k=1}^{K} \sum_{i=1}^{Q^{(k)}} w_i^{(k)} \left\| \alpha \circ r(q_i^{(k)}) \right\|_2^2. \quad (10)$$

where $w := [w_1^{(1)}, \ldots, w_{Q^{(K)}}^{(K)}]^\top$ and $q := [q_1^{(1)}, \ldots, q_{Q^{(K)}}^{(K)}]^\top$.

For commonly used state and input trajectory parameterisations, approximation errors are inevitable [8]. Hence, an NLP with (10) forced to zero with an arbitrary design of the quadrature rule may not always be feasible except in a few special cases; one such special case is to select a quadrature rule such that the quadrature abscissae $q_i^{(k)}$ match the collocation points $d_i^{(k)}$, which recovers the DC scheme [8].

In other words, with a quadrature rule of sufficiently high order to capture the error behavior inside the interval, (10) cannot be reduced to zero in general. The QPM minimizes the integrated residuals via an additional penalty term, being

$$\min_{\chi, \upsilon, t_0, t_f} J_h\left(\chi, \upsilon, t_0, t_f\right) + \frac{1}{2\rho} \mathcal{R}\left(\chi, \upsilon, t_0, t_f, w, q\right) \quad (11)$$

subject to (3c) and (3d). $J_h$ is the original objective as in (3a) and $\rho > 0$ is the regularization weight of the penalty term. In contrast, the direct alternating integrated residuals (DAIR) approach solves the problem by alternating between the minimization of the integrated residuals (10) and the minimization of the original objective subject to inequality constraints on the MIRS error for each dynamic equation. Other benefits of working with the residual error in its integrated form, beyond the aspects discussed in this paper, are covered in previous work [5], [8].

## III. Integrated Residual Regularized Direct Collocation

The development of IRRDC aims to bring the benefit of integrated residual approaches in handling challenging problems to the DC framework, while maintaining DC's ease of implementation and computational efficiency as much as possible.

The NLP formulation arising from IRRDC uses the objective formulation from QPM and the constraint formulation from DC, i.e.:

$$\min_{\chi, \upsilon, t_0, t_f} J_h\left(\chi, \upsilon, t_0, t_f\right) + \frac{1}{2\rho} \mathcal{R}\left(\chi, \upsilon, t_0, t_f, w, q\right) \quad (12)$$

subject to (8) and (3c) for all $k \in \mathbb{I}_K$ and $i \in \mathbb{I}_{N^{(k)}}$, and (3d) for some $k_i \in \mathbb{I}_K$ and $k_j \in \mathbb{I}_K$.

For DOPs with consistent overdetermined constraints [6], [8], use of (8) may result in an infeasible NLP. Relaxing (8) to

$$-\epsilon \leq f\left(\dot{\tilde{x}}^{(k)}(d_i^{(k)}), \tilde{x}^{(k)}(d_i^{(k)}), \tilde{u}^{(k)}(d_i^{(k)}), d_i^{(k)}\right) \leq \epsilon, \quad (13)$$

is generally not recommended for DC, as it is very difficult to find a vector of constants $\epsilon$ that is large enough to make the NLP feasible but small enough to ensure a sufficiently accurate solution. In contrast, IRRDC has a second mechanism where the IRE is penalized, thus alleviating the risk that $\epsilon$ is chosen too large. In the extreme case, IRRDC is equivalent to QPM.

The choice of $\rho$ in IRRDC can be seen as a mechanism for balancing solution accuracy and optimality for a given discretization mesh design. A large value of $\rho$ drives the solution of (III) towards the DC solution, prioritizing the reduction of the original objective. However, as later shown in Figure 2 with the example problem, a low objective value reported by the NLP solver may be misleading. As $\rho$ decreases, the method can automatically converge to a more accurate solution when multiple solutions exist with minimal or negligible differences in nominal cost. Further reduction of $\rho$ leads to a solution that is heavily biased towards higher accuracy, albeit at the cost of a higher objective value.

Although IRRDC may seem like a simple blend of DC and IRM, it stands out as a method that explicitly handles residual error through both point-wise and integral measures, whereas previous methods focus on only one of the two. The method addresses several practical implementation challenges and offers additional benefits.

### A. Benefits of IRRDC against DC

By forcing the residual to be zero only at collocation points, DC relies on posterior error analysis and MR to address the challenge of errors arising between collocation points. This works well except for DOPs with singular arcs or high-index DAEs, where excessive fluctuations can occur in the solution. These fluctuations makes MR inefficient and ineffective in reducing the errors.

To suppress the fluctuations in practice, it is common to introduce regularization terms in the objective to penalize control actions [10] (with $\int_{t_0}^{t_f} u^\top(t) R u(t)\,\mathrm{d}t$ where $R$ is a positive-definite weighting matrix, referred to later as control regularization, or CR) or to penalize the control rate. For DOPs with a large number of input variables and complex structures, managing the regularization to limit its impact on solution optimality can be challenging.

In the development of IRM, it was found that minimizing the IRE automatically suppresses the singular arc fluctuations in the solution [8], [11]. For solving DOPs numerically on a discretization mesh, trading off solution optimality with accuracy, as in IRRDC, is a better and more justifiable approach compared to the trade-off between optimality and control magnitudes, as seen in DC with CR.

### B. Benefits of IRRDC against other IRM Schemes

The addition of point-wise residual constraints (8) or (13) in IRRDC helps reduce the need for carefully tuning of configuration and scaling parameters. For DOPs with state variables spanning vastly different numerical ranges, appropriate constraint scaling in the computation of (10) can be critical. Section IV-A illustrates such an example, highlighting how improper trade-offs in the accuracy of different dynamic equations can lead to erroneous solutions. In IRRDC, the risk of certain dynamic equations being overshadowed is significantly reduced, since each dynamic relationship must independently satisfy the respective point-wise constraint. While DAIR faces a similar challenge, it is less sensitive to constraint scaling. However, DAIR requires users to specify the desired MIRS error for each dynamic equation, which makes it more complicated to configure than IRRDC.

In QPM, balancing the progress in minimizing the original objective with maintaining dynamic feasibility — primarily through careful selection of the penalty parameter $\rho$ — is both critical and challenging. Therefore, the scheme prefers an NLP solver that handles both terms in the objective (11) separately. Additionally, it is unlikely that the integrated residual would vanish to zero at the converged solution, making the direct use of NLP solvers that handle equality constraints through quadratic penalty terms also unsuitable. These factors make QPM prefer tailored solvers [12]. In contrast, IRRDC enforces dynamic constraints primarily through (8) or (13), while the penalty term allows further improvements to solution accuracy where degrees of freedom remain. As a result, IRRDC works well with standard off-the-shelf NLP solvers, without the need to distinguish between the two terms in the objective (12). Furthermore, due to the use of both mechanisms, IRRDC can often employ a lower sampling of quadrature points compared to QPM and DAIR, leading to reduced computational cost. If desired, the penalty term can even be introduced only in the later stages of the iteration process, once the solution is nearly converged. This allows IRRDC to take advantage of many inexpensive DC iterations early in the optimization.

## IV. Example Problems

Here, we present two example problems to demonstrate the main advantages of the IRRDC. Both DOPs are transcribed using `ICLOCS2` [13], and numerically solved to a tolerance of $10^{-9}$ with NLP solver `IPOPT` [14] (version 3.12.9).

### A. Singular Control Example: Goddard Rocket

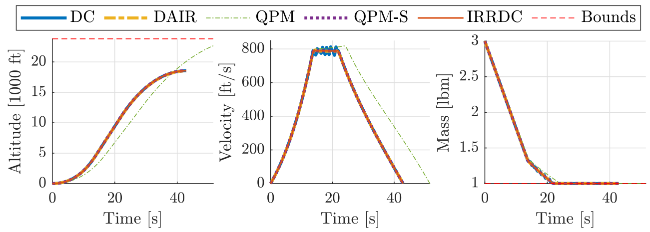
The Goddard rocket problem [15] is a frequently used example for the analysis of optimal control problems with singular arcs. Here, we implement the Goddard rocket problem as described in [2, Ex. 4.9] with a solution structure of *bang-singular-bang*. We use Hermite-Simpson discretisation with 100 equispaced mesh nodes, unless stated otherwise.

*1) Suppression of singular arc fluctuations:* Using DC, it is known for the solution to be oscillatory on the singular arc if no special treatment is implemented. To remove the singular arc oscillations, a multiphase formulation is typically used with additional constraints known as singular arc conditions imposed specifically for the second phase, which corresponds to the one with singular control.

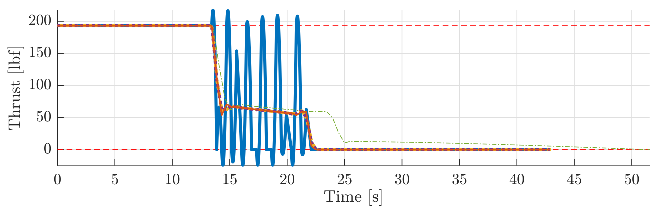
In [8], the ability of the IRM of DAIR to alleviate the oscillations on the singular arc has been demonstrated on a fixed equidistant discretization mesh as a single-phase problem. The IRRDC method yields similar improvements to the results: the large fluctuations on the singular arc have been suppressed (Figure 1).

*2) Comparison with DC and other IRM schemes:* Table I demonstrates that the IRRDC solution achieves significantly higher accuracy across all metrics compared to DC. In fact, for DC to produce a solution with maximum AL and RL errors lower than those of the IRRDC solution using 100 equispaced nodes, DC requires a much denser mesh of 1310 equispaced nodes (denoted as DC-D). Even then, the dense mesh solution still exhibits pronounced fluctuations along the singular arc and large CV errors.

As shown in Table II, the computation time for IRRDC is higher than DC when using the same mesh size, primarily due to the denser sampling of quadrature points and more complicated derivative computations using spares finite difference. However, when comparing solutions of similar accuracy, IRRDC is more than 10 times faster than the DC-D approach. This highlights the efficiency advantage of IRRDC and underscores the importance of comparing numerical methods based on equivalent accuracy levels rather than identical mesh sizes.




(a) State trajectories




(b) Input trajectories

Fig. 1: Numerical solution to the Goddard rocket problem.

TABLE I: Comparison of error measures (maximum error)

| | AL Error | RL Error | CV Error | MIRNS Error |
|---|---|---|---|---|
| DC | 3.6e-01 | 1.1e-04 | 24.1 | 1.4e-03 |
| DC-D[a] | 2.2e-03 | 8.9e-07 | 24.1 | 8.7e-06 |
| DAIR[b] | 1.1e-02 | 7.2e-06 | 2.4e-01 | 1.8e-06 |
| QPM-S[c, d] | 1.3e-02 | 8.8e-06 | 2.2e-01 | 3.2e-06 |
| IRRDC[c] | 2.4e-03 | 3.1e-06 | 4.4e-05 | 2.8e-07 |

[a] This is a DC solution with a dense mesh of 1310 equispaced nodes.
[b] Requested MIRS error being $10^{-2}$, $10^{-4}$ and $10^{-8}$ for the three dynamic constraints respectively.
[c] A fixed $\rho = 10^{-6}$ used.
[d] Scaling of [1 4.76 11900] used for the three dynamic constraints.

When compared to QPM, the addition of point-wise residual constraints in IRRDC alleviate the need for constraint scaling in (10). Even without constraint scaling, the IRRDC solution closely matches the reference solution from the literature [2], whereas the QPM solution differs as seen in Figure 1. The erroneous solution is obtained because the altitude state equation with large amplitude overshadows the mass equation. Hence the dynamic relationship for mass contains larger errors, resulting in a lower mass reduction. This error propagates through the dynamic relationship with the QPM solution showing the rocket reaching a much higher, but physically unachievable altitude.

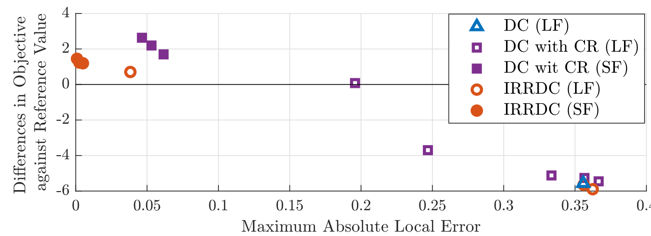
For QPM to yield a more accurate solution, constraint scaling is needed, with the solution shown as QPM-S. DAIR manages such trade-offs through the specification of the requested MIRS error for each dynamic constraint separately, making its configuration also more complicated than IRRDC. In terms of computational performance, although the comparison in Table II may not be very representative (see footnotes of the table), the reduction in the number of quadrature points needed within each interval already suggests that IRRDC is computationally lighter than QPM and DAIR.

*3) Comparison with other commonly used regularization mechanisms:* Figure 2a compares the NLP solution of the IRRDC transcription against DC with CR. The comparison with

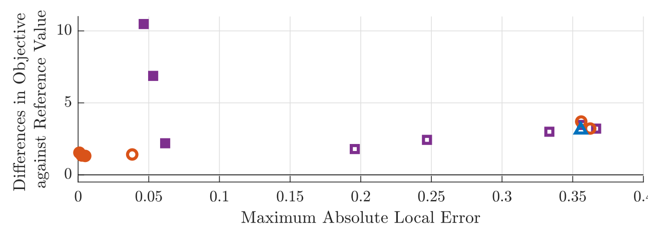
TABLE II: Comparison of computational performance

| | Quadrature Points[a] | NLP Iterations | Derivatives Comp. Time[b] | Solver Time |
|---|---|---|---|---|
| DC | 2 | 119 | 1.54 s | 0.58 s |
| DC-D | 2 | 357 | 19.96 s | 18.51 s |
| DAIR | 8 | 49 + 92[c] | 63.61 s[e] | 0.78 s |
| QPM-S[d] | 8 | 311 | 149.46 s[e] | 1.03 s |
| IRRDC | 4 | 42 | 3.08 s | 0.24 s |

[a] Inside each mesh interval.
[b] Derivatives are computed using sparse finite difference.
[c] DAIR solves 2 NLPs in sequence.
[d] QPM prefers a tailored solver, so using IPOPT is not recommended. For a more representative comparison of computational performance with DC, please refer to [6].
[e] Obtained using an older non-optimised code-base (efficient computations learned from IRRDC development not yet back implemented to other IRM schemes in ICLOCS).




(a) Solver reported objectives from the NLP solutions




(b) Actual objectives from the simulation of the solutions

Fig. 2: IRRDC compared to DC with CR for the suppression of singular arc fluctuations when using different regularization weights. Reference objective is -18550.9 from [2, Ex. 4.9]. LF indicates large fluctuations can be visually observed in the solution trajectories. SF indicates the fluctuations has been suppressed — some minor ones may still be seen where control switches, but not inside the singular arc.

input rate regularization yields similar conclusions hence not included for better clarity. It can be seen in the figure that many of the solutions with singular arc fluctuations report lower objective values in comparison to that of a high accuracy solution [2, Ex. 4.9]. This highlights an important point of caution when comparing numerical DOP solutions: the lower objective values obtained by solutions with higher errors may be misleading.

For a proper comparison, we obtained the actual objective by implementing the numerical DOP solutions through high-accuracy simulations (with instead and present the results in Figure 2b. Both regularization schemes demonstrate the following trade-off behaviors:

- With a small penalty on the control action or on the integrated residuals, large fluctuations may occur in

the numerical solution, reducing its accuracy. Such solutions will result in significantly higher cost when implemented, compared to the values reported by the NLP solver, making them undesirable.
- As the penalty increases, the fluctuations will be suppressed leading to an increase in solution accuracy. Such solutions are the desirable ones with low objective values that are realistically achievable.
- With a further increase in penalty, the bias between the original objective and the regularization cost shifts. The solutions will become increasingly suboptimal with a higher nominal cost (1a).

Secondly, we compare the solution optimality in the multi-objective sense, focusing on solutions where singular arc fluctuations are suppressed (i.e. solutions in Figure 2b marked with SF). It is observed that the solution obtained by IRRDC is the preferred one, as it simultaneously achieves the lowest objective and the lowest error compared to the other regularization alternatives. This is because IRRDC penalizes the IRE, rather than the inputs. Hence, IRRDC will not discriminate against large inputs that may be required in the optimal solution, as long as the errors are low.

### B. *Algebraic Constraint Example: Ventilator Control*

In the second example, we focus on demonstrating the benefit of the IRRDC in handling problems with additional algebraic constraints, and on offering insights regarding the role of the regularization weights in IRRDC. The problem presented is the ventilator control problem introduced in [16].

The IRM's effectiveness in enhancing the precision of algebraic constraints with a very coarse discretization mesh (3 intervals, each with a middle point) was shown in [16] for illustration. Figure 3 reproduces these solutions together with the solution with IRRDC under different choices of $\rho$. Firstly, the figure provides a graphical illustration of the residual errors that arise between collocation points, a key consideration motivating the development of IRM as well as IRRDC. Secondly, it can be observed from the figure that as $\rho$ decreases, the IRRDC solution improves compared to the DC solution and tends toward the DAIR solutions.

## V. Conclusions

With the proposed integrated residual regularized direct collocation (IRRDC) method for the numerical solution of dynamic optimization problems, the residual errors of dynamic constraints are handled both through point-wise and integral measures. In comparison to the widely used direct collocation method, which only enforces point-wise dynamic constraints, IRRDC greatly improves the method's solution accuracy and reliability for challenging problems, such as those involving singular arcs. In comparison to integrated residual approaches, IRRDC provides an easy-to-configure framework that retains the simplicity and computational efficiency of DC. Future work could focus on compatibility of IRRDC with other aspects of the numerical framework, such as mesh refinement, to further assess its limitations and potential.

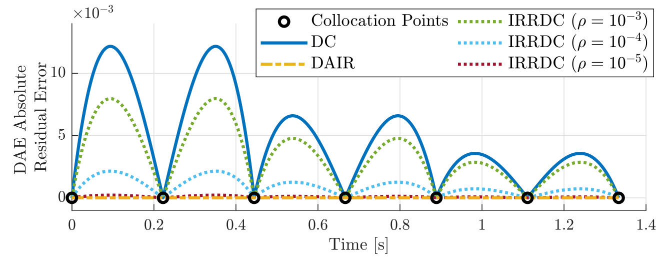



Fig. 3: Comparison of residual error for the algebraic constraint in the ventilator control problem

## References


[1] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," *Nonlinear model predictive control: towards new challenging applications*, pp. 391–417, 2009.

[2] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming: Second Edition*. Advances in Design and Control, Society for Industrial and Applied Mathematics, 2010.

[3] M. A. Patterson and A. V. Rao, "GPOPS-II: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 1, p. 1, 2014.

[4] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[5] M. P. Neuenhofen, *Quadratic integral penalty methods for numerical trajectory optimization*. PhD thesis, Imperial College London, 2022.

[6] M. P. Neuenhofen, E. C. Kerrigan, and Y. Nie, "Numerical comparison of collocation vs quadrature penalty methods," in *62nd IEEE Conference on Decision and Control (CDC)*, pp. 4285–4290, IEEE, 2023.

[7] M. P. Neuenhofen and E. C. Kerrigan, "A direct method for solving integral penalty transcriptions of optimal control problems," in *59th IEEE Conference on Decision and Control (CDC)*, pp. 4822–4823, 2020.

[8] Y. Nie and E. C. Kerrigan, "Solving dynamic optimization problems to a specified accuracy: An alternating approach using integrated residuals," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 548–555, 2023.

[9] B. A. Finlayson and L. E. Scriven, "The method of weighted residuals—a review," *Appl. Mech. Rev*, vol. 19, no. 9, pp. 735–748, 1966.

[10] E. R. Pager and A. V. Rao, "Method for solving bang-bang and singular optimal control problems using adaptive Radau collocation," *Computational Optimization and Applications*, vol. 81, no. 3, p. 857–887, 2022.

[11] M. P. Neuenhofen and E. C. Kerrigan, "Dynamic optimization with convergence guarantees," *arXiv preprint arXiv:1810.04059*, 2018.

[12] M. P. Neuenhofen and E. C. Kerrigan, "Solving problems with inconsistent constraints with a modified augmented lagrangian method," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2592–2598, 2023.

[13] Y. Nie, O. Faqir, and E. C. Kerrigan, "ICLOCS2: try this optimal control problem solver before you try the rest," in *2018 UKACC 12th International Conference on Control (CONTROL)*, pp. 336–336, IEEE, 2018.

[14] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[15] R. H. Goddard, "A method of reaching extreme altitudes," 1920.

[16] E. C. Kerrigan, Y. Nie, O. Faqir, C. H. Kennedy, S. A. Niederer, J. A. Solis-Lemus, P. Vincent, and S. E. Williams, "Direct transcription for dynamic optimization: A tutorial with a case study on dual-patient ventilation during the COVID-19 pandemic," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2597–2614, 2020.