# On the Uses of Linear-Quadratic Methods in Solving Nonlinear Dynamic Optimization Problems with Direct Transcription

IMECE2020-23885

Daniel R. Herber[1] Athul K. Sundarrajan[2]

[1] Assistant Professor
Colorado State University, Department of Systems Engineering
daniel.herber@colostate.edu

[2] Graduate Student, **Presenting Author**
Colorado State University, Department of Systems Engineering
Athul.Sundarrajan@colostate.edu

①

# Introduction

# → Introduction

- **Dynamic optimization (DO) and optimal control are the optimization of systems of equations with time-varying behavior** [Betts 2010]
- Because of time dependent elements, the solution must satisfy the infinite points in a given time horizon $t \in [t_o, t_f]$
- There are two family of methods that are used to solve DO problems: **direct numerical methods** and **indirect methods**
- **Indirect methods** use optimality conditions to find the optimal dynamics [Bryson and Ho 1975]
- **Direct methods** transcribe the infinite-dimensional problem to a finite-dimensional optimization problem [Rao 2010]
- Here we focus on a direct method solution strategy known as **Direct Transcription (DT)** [Betts 2010; Rao 2010]

# → Problem Description

$$\min_{x=[u,\xi,p]} \quad o = \int_{t_0}^{t_f} \ell\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{p}\right) dt + m\left(\boldsymbol{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f\right) \tag{1a}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{p}\right) = \boldsymbol{0} \tag{1b}$$

$$\boldsymbol{h}\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f\right) = \boldsymbol{0} \tag{1c}$$

$$\boldsymbol{g}\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{p}, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f\right) \leq \boldsymbol{0} \tag{1d}$$

$$\text{where:} \quad \boldsymbol{\xi}_0 = \boldsymbol{\xi}(t_0), \;\; \boldsymbol{\xi}_f = \boldsymbol{\xi}(t_f) \tag{1e}$$

- Consider the elements in a general **Nonlinear Dynamic Optimization (NLDO)** problem
  - The optimization variables are controls $\boldsymbol{u}$, states $\boldsymbol{\xi}$, and parameters $\boldsymbol{p}$
  - The states $\boldsymbol{\xi}(t)$ and open-loop controls $\boldsymbol{u}(t)$ are time varying quantities
  - $o(\cdot)$ is the objective term composed of the Lagrange term $\ell(\cdot)$ and the Mayer term $m(\cdot)$
  - $\boldsymbol{f}(\cdot)$ is the explicit first order state derivative function
  - $\boldsymbol{h}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are the additional equality and inequality constraints

# → Direct Transcription

$$\min_{\mathbf{x}=[\text{vec}(\boldsymbol{U}),\text{vec}(\boldsymbol{\Xi}),\boldsymbol{p}]} \quad v^{DT}\left(\boldsymbol{t},\boldsymbol{U},\boldsymbol{\Xi},\boldsymbol{p},\boldsymbol{\Xi}_1,\boldsymbol{\Xi}_{n_t}\right) \tag{2a}$$

$$\text{subject to:} \quad \mathbf{h}^{DT} = \begin{bmatrix} \mathbf{h}_{\zeta}^{DT}\left(\boldsymbol{t},\boldsymbol{U},\boldsymbol{\Xi},\boldsymbol{p}\right) \\ \mathbf{h}_{\boldsymbol{h}}^{DT}\left(\boldsymbol{t},\boldsymbol{U},\boldsymbol{\Xi},\boldsymbol{p},\boldsymbol{\Xi}_1,\boldsymbol{\Xi}_{n_t}\right) \end{bmatrix} = \mathbf{0} \tag{2b}$$

$$\mathbf{g}^{DT}\left(\boldsymbol{t},\boldsymbol{U},\boldsymbol{\Xi},\boldsymbol{p},\boldsymbol{\Xi}_1,\boldsymbol{\Xi}_{n_t}\right) \leq \mathbf{0} \tag{2c}$$

- DT methods transcribe the infinite-dimensional DO problem to a finite dimensional problem that can be solved with standard NLP solvers [Betts 2010]
  - The **objective function** in the continuous problem is approximated using numerical **quadrature** methods
  - The **dynamic** state equation is transcribed through different **collocation** methods
  - Path constraints are enforced at each time point
  - Initial and final value constraints can be expressed as they were

# → Linear-Quadratic Dynamic Optimization

$$\min_{\boldsymbol{x}=[\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p}]} \quad o^Q\left(t,\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p},\boldsymbol{\xi}_0,\boldsymbol{\xi}_f,\bar{\boldsymbol{x}}\right) \tag{3a}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}^L\left(t,\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p},\bar{\boldsymbol{x}}\right) = \boldsymbol{0} \tag{3b}$$

$$\boldsymbol{h}^L\left(t,\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p},\boldsymbol{\xi}_0,\boldsymbol{\xi}_f,\bar{\boldsymbol{x}}\right) = \boldsymbol{0} \tag{3c}$$

$$\boldsymbol{g}^L\left(t,\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p},\boldsymbol{\xi}_0,\boldsymbol{\xi}_f,\bar{\boldsymbol{x}}\right) \leq \boldsymbol{0} \tag{3d}$$

$$\text{where:} \quad \boldsymbol{\xi}_0 = \boldsymbol{\xi}(t_0),\ \boldsymbol{\xi}_f = \boldsymbol{\xi}(t_f) \tag{3e}$$

- **Linear-Quadratic Dynamic Optimization (LQDO)** is a special case of the **(NLDO)** with a quadratic objective function subject to linear constraints [Herber 2017]
- This implies linear dynamic models
- Through DT, the given **LQDO** problem is converted into a large sparse **quadratic programming (QP)**
- Many efficient tools have been developed to solve the QPs

# → Advantages of LQDO Problems

$$\min_{\mathbf{x}} \quad \mathbf{f}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{H}\mathbf{x} \tag{4a}$$

$$\text{subject to:} \quad \mathbf{A_h}\mathbf{x} - \mathbf{b_h} = \mathbf{0} \tag{4b}$$

$$\mathbf{A_g}\mathbf{x} - \mathbf{b_g} \leq \mathbf{0} \tag{4c}$$

- Structure of a given optimization problem can been used to increase the efficiency and robustness of the techniques used to solve the problem
- If the objective function of the QP problem is convex and the feasible set is nonempty, then there exists a **global solution** to the problem

# Research Questions

- Many DO problems have both LQ and nonlinear elements yet there are not many studies that look at efficient implementation of LQ elements
- There are efficient tools for LQDO problems
- We know that the problem structure contributes to their efficiency

---

1. So how can we incorporate these elements to solve NLDO problems?
2. If a NLDO has both LQ and NL elements, can we take advantage of this structure?
3. Even if there are no LQ elements, is there a way to still use LQ elements and solve the original NLDO problem?

②

# Solving NLDO using LQDO Elements

# → Direct Incorporation of LQ Elements

- For a given NLDO problem, we can treat the LQ elements differently than the general nonlinear elements
- This can be done by manually specifying which problem elements are LQ or automated methods

$$\min_{\boldsymbol{x}=[\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{p}]} \quad \int_{t_0}^{t_f} \left[\ell^{OL}(\cdot) + \ell^{NL}(\cdot)\right] dt + m^{OL}(\cdot) + m^{NL}(\cdot) \tag{5a}$$

subject to:

$$\dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}(\cdot) = \begin{bmatrix} \dot{\boldsymbol{\xi}}^{OL}(t) - \boldsymbol{f}^{OL}(\cdot) \\ \dot{\boldsymbol{\xi}}^{NL}(t) - \boldsymbol{f}^{NL}(\cdot) \end{bmatrix} = \boldsymbol{0} \tag{5b}$$

$$\boldsymbol{h}(\cdot) = \begin{bmatrix} \boldsymbol{h}^{OL}(\cdot) \\ \boldsymbol{h}^{NL}(\cdot) \end{bmatrix} = \boldsymbol{0} \tag{5c}$$

$$\boldsymbol{g}(\cdot) = \begin{bmatrix} \boldsymbol{g}^{OL}(\cdot) \\ \boldsymbol{g}^{NL}(\cdot) \end{bmatrix} \leq \boldsymbol{0} \tag{5d}$$

# → Direct Incorporation of LQ Elements (continued)

There are several potential advantages for using this strategy:

- NLP solvers use first and second derivative information
- For LQ elements, the hessian is zero, and the jacobian is constant
- Identifying and pre-computing these values for LQ elements will help reduce the problem complexity and cost

$$\mathcal{L}\left(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma}\right) = v(\mathbf{x}) + \boldsymbol{\lambda} \cdot \mathbf{h}(\mathbf{x}) + \boldsymbol{\sigma} \cdot \mathbf{g}(\mathbf{x}) \tag{6}$$

$$\boldsymbol{H}_{\mathbf{x}}^{\mathcal{L}}(\mathbf{x}) = \sum_j \boldsymbol{\lambda}_j^{OL} \cdot \cancelto{\mathbf{0}}{\left[\boldsymbol{H}_{\mathbf{x}}^{\mathbf{h}_j^{OL}}(\mathbf{x})\right]} + \sum_j \boldsymbol{\lambda}_j^{NL} \cdot \left[\boldsymbol{H}_{\mathbf{x}}^{\mathbf{h}_j^{NL}}(\mathbf{x})\right] + \cdots \tag{7}$$

# → Two-Level Methods

- Two-level optimization methods partition the problem variables into upper- and lower-level variables $\boldsymbol{x}^u$ and $\boldsymbol{x}^l$ [Bard 1988]

$$\min_{\boldsymbol{x}^u} \quad o\left(\boldsymbol{x}^u, \boldsymbol{x}^l\right) \tag{8a}$$

$$\text{subject to:} \quad \boldsymbol{x}^u \in \Omega \tag{8b}$$

$$\boldsymbol{x}^l \in L(\boldsymbol{x}^u) \tag{8c}$$

- For a candidate variable $\boldsymbol{x}^u$ provided by the upper-level problem, the lower-level problem involves finding the optimal value for $\boldsymbol{x}^l$
- Here we focus on NLDO problems where fixing the values of $\boldsymbol{p}$ yields a lower-level LQDO problem that solves for $\boldsymbol{\xi}(t)$ and $\boldsymbol{u}(\boldsymbol{t})$ [Herber and Allison 2019]

# → Quasilinearization

- Quasilinearization is a method that solves NLDO problems by expanding them about a reference trajectory $\bar{\boldsymbol{x}}$ using a Taylor series and sequentially solving QPs [Jaddu 2002]
  - Nonlinear constraints are expanded till the linear:

$$\boldsymbol{e}(\boldsymbol{x}) \approx \boldsymbol{e}^{L}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \boldsymbol{e}(\bar{\boldsymbol{x}}) + \boldsymbol{J}_{\boldsymbol{x}}^{\boldsymbol{e}}(\bar{\boldsymbol{x}})[\boldsymbol{x} - \bar{\boldsymbol{x}}] \tag{9}$$

  - Nonlinear objective terms are expanded till quadratic:

$$e(\boldsymbol{x}) \approx e^{Q}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = e(\bar{\boldsymbol{x}}) + [\boldsymbol{J}_{\boldsymbol{x}}^{e}(\bar{\boldsymbol{x}})][\boldsymbol{x} - \bar{\boldsymbol{x}}] + \frac{1}{2}[\boldsymbol{x} - \bar{\boldsymbol{x}}]\boldsymbol{H}_{\boldsymbol{x}}^{e}(\bar{\boldsymbol{x}})[\boldsymbol{x} - \bar{\boldsymbol{x}}] \tag{10}$$

- The reference LQDO problem is solved and the solution from the previous iteration used as the reference trajectory for the next iteration
- This process continues until the difference between successive iterations is small

③

# Case Studies

# → Container Crane

**All the case studies included in the paper are solved using a free and open-source MATLAB-based tool, *DTQP***
**https://github.com/danielrherber/dt-qp-project**

- In this problem, we want to optimally transfer the states of a container crane subject to linear path constraints
- All parts except for $\dot{\xi}_6$ are Linear Quadratic elements

$$\min_{\boldsymbol{u},\boldsymbol{\xi}} \quad \frac{1}{2} \int_0^{t_f} \left[ \xi_3^2 + \xi_6^2 + \rho \left[ u_1^2 + u_2^2 \right] \right] dt \tag{11a}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}} = \begin{bmatrix} \xi_4 \\ \xi_5 \\ \xi_6 \\ u_1 + c_4 \xi_3 \\ u_2 \\ - \left[ u_1 + c_5 \xi_3 + 2 \xi_5 \xi_6 \right] / \xi_2 \end{bmatrix} \tag{11b}$$

$$\boldsymbol{\xi}(0) = (0, 22, 0, 0, -1, 0) \tag{11c}$$

$$\boldsymbol{\xi}(t_f) = (10, 14, 0, 2.5, 0, 0) \tag{11d}$$

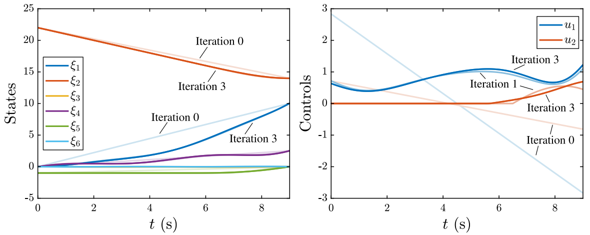
$$-c_6 \leq \xi_4 \leq c_6, \quad c_7 \leq \xi_5 \leq c_7 \tag{11e}$$

$$-c_1 \leq u_1 \leq c_1, \quad c_2 \leq u_2 \leq c_3 \tag{11f}$$

# → Container Crane Results

- The problem was tested using **quasilinearization** and the nonlinear optimization was solved using an **interior point (IP)** algorithm

## Quasilinearization Results for Container Crane



# → Container Crane Results (continued)

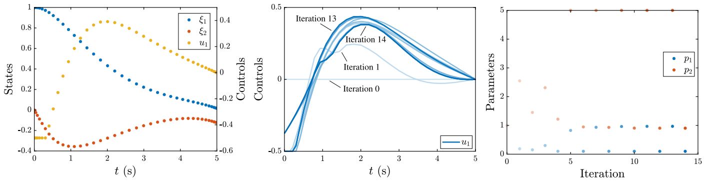
- **Quasilinearization was generally faster**
- When LQ problem elements were directly incorporated in IP method, the solution time was **15 – 165% faster**

| Method | $v$ | Iter. | $T_{sym}$ | $T_{int}$ | $T_{opt}$ | $T$ |
|---|---|---|---|---|---|---|
| QLIN-TR20 | 0.0380 | 3 | 0.57 | 0.007 | 0.03 | 0.03 |
| IP-SD-OLQ-TR20 | 0.0380 | 20 | 1.42 | 0.002 | 0.08 | 0.08 |
| IP-SD-NL-TR20 | 0.0380 | 20 | 2.17 | 0.003 | 0.16 | 0.16 |
| IP-FD-OLQ-TR20 | 0.0380 | 22 | 0.00 | 0.003 | 0.22 | 0.22 |
| IP-FD-NL-TR20 | 0.0380 | 22 | 0.00 | 0.001 | 0.24 | 0.24 |
| QLIN-TR200 | 0.0375 | 3 | 0.56 | 0.010 | 0.31 | 0.32 |
| IP-SD-OLQ-TR200 | 0.0375 | 29 | 1.41 | 0.002 | 0.41 | 0.41 |
| IP-SD-NL-TR200 | 0.0375 | 29 | 2.17 | 0.003 | 0.58 | 0.59 |
| IP-FD-OLQ-TR200 | 0.0375 | 30 | 0.00 | 0.003 | 0.88 | 0.89 |
| IP-FD-NL-TR200 | 0.0375 | 29 | 0.00 | 0.002 | 0.95 | 0.95 |
| QLIN-TR2000 | 0.0375 | 3 | 0.56 | 0.033 | 8.23 | 8.26 |
| IP-SD-OLQ-TR2000 | 0.0375 | 35 | 1.42 | 0.008 | 7.21 | 7.21 |
| IP-SD-NL-TR2000 | 0.0375 | 35 | 2.18 | 0.004 | 8.25 | 8.26 |
| IP-FD-OLQ-TR2000 | 0.0375 | 39 | 0.00 | 0.003 | 13.92 | 13.92 |
| IP-FD-NL-TR2000 | 0.0375 | 35 | 0.00 | 0.003 | 13.95 | 13.95 |
| QLIN-PS10 | 0.0376 | 4 | 0.55 | 0.009 | 0.03 | 0.04 |
| IP-SD-OLQ-PS10 | 0.0376 | 24 | 1.42 | 0.002 | 0.08 | 0.08 |
| IP-SD-NL-PS10 | 0.0376 | 28 | 2.17 | 0.003 | 0.21 | 0.21 |
| IP-FD-OLQ-PS10 | 0.0376 | 29 | 0.00 | 0.003 | 0.25 | 0.26 |
| IP-FD-NL-PS10 | 0.0376 | 31 | 0.00 | 0.002 | 0.31 | 0.31 |

# → Failure of Quasilinearization: Van der Pol

- Variations of the Van der Pol oscillator problem have been studied as a convergent problem using quasilinearization [Jaddu 2002]
- When some parameters are introduced [Azad and Alexander-Ramos 2019], the method **fails to converge** using quasilinearization
- The parameters candidates for successive iterations oscillates between two trajectories
- An accurate solution can be obtained using an IP method

**Results for a Variation of the Van der Pol Problem with Parameters**



# → Simple Suspension

- This is a simplified quarter-car vehicle suspension problem as discussed in [Herber and Allison 2019]

## Simple Suspension Problem Formulation

$$\min_{u,\boldsymbol{\xi},\boldsymbol{p}} \int_0^{t_f} \left[w_1 \xi_1^2 + w_2 \dot{\xi}_4^2 + w_3 u^2\right] dt \tag{12a}$$

subject to:

$$\dot{\boldsymbol{\xi}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_t}{m_{us}/4} & \frac{-[p_1+c_t]}{m_{us}/4} & \frac{p_2}{m_{us}/4} & \frac{p_1}{m_{us}/4} \\ 0 & -1 & 0 & 1 \\ 0 & \frac{p_1}{m_s/4} & \frac{-p_2}{m_s/4} & \frac{-p_1}{m_s/4} \end{bmatrix} \boldsymbol{\xi}(t) + \cdots \tag{12b}$$

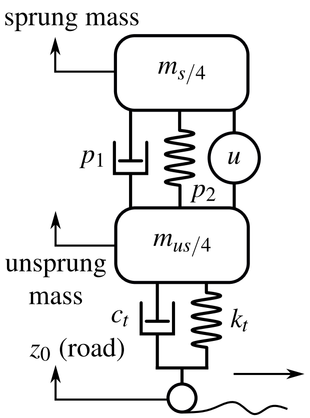
$$+ \begin{bmatrix} 0 \\ \frac{-1}{m_{us}/4} \\ 0 \\ \frac{1}{m_s/4} \end{bmatrix} u(t) + \begin{bmatrix} -1 \\ \frac{c_t}{m_{us}/4} \\ 0 \\ 0 \end{bmatrix} \dot{z}_0(t)$$

$$\boldsymbol{\xi}(0) = \mathbf{0} \tag{12c}$$

$$|\xi_3(t)| \le r_{\max} \tag{12d}$$

$$\boldsymbol{p}_{\min} \le \boldsymbol{p} \le \boldsymbol{p}_{\max} \tag{12e}$$

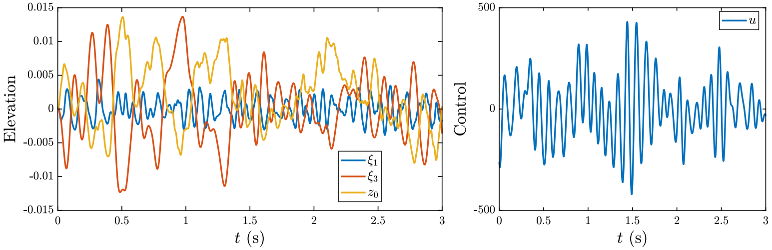
## Quarter-Car Suspension



# → Simple Suspension Results

- Tests were conducted using all the methods including the **two-level fixed-parameter (TLFP)** and **all-at-once (IP)** strategies
- Optimal value for the **damper constant is 101.12 Ns/m** and **21.558 kN/m for the spring constant**

**Optimal States and Controls in the Simple Suspension Problem**



# → Simple Suspension Results (continued)

| Method | $v$ | Iter. | $T_{sym}$ | $T_{int}$ | $T_{opt}$ | $T$ |
|---|---|---|---|---|---|---|
| TLFP-TR200 | 1.977 | 17 | — | — | — | 1.10 |
| IP-SD-OLQ-TR200 | 1.977 | 12 | 6.46 | 0.003 | 0.3 | 0.31 |
| IP-SD-NL-TR200 | 1.977 | 12 | 6.47 | 0.003 | 0.3 | 0.33 |
| IP-FD-OLQ-TR200 | 2.048 | 598 | 0.00 | 0.003 | 21.8 | 21.80 |
| IP-FD-NL-TR200 | 2.048 | 598 | 0.00 | 0.003 | 21.9 | 21.92 |
| TLFP-TR2000 | 1.996 | 22 | — | — | — | 9.18 |
| IP-SD-OLQ-TR2000 | 1.996 | 13 | 6.49 | 0.005 | 1.6 | 1.63 |
| IP-SD-NL-TR2000 | 1.996 | 13 | 6.51 | 0.004 | 1.7 | 1.74 |
| IP-FD-OLQ-TR2000 | 1.997 | 599 | 0.00 | 0.006 | 216.7 | 216.68 |
| IP-FD-NL-TR2000 | 1.997 | 599 | 0.00 | 0.004 | 216.4 | 216.43 |
| QLIN-TRX | — | did not converge | | | | |

- TLFP approach is nearly **20-24**$\times$ **faster** than the IP method when using finite-differences
- This example highlights the effectiveness of **TLFP** methods in the **absence of symbolic derivatives** or other efficient derivative methods
- The difference between OLQ and NL options is **relatively small**, as only two of the state derivative functions were OLQ elements, and the objective function was nonlinear
- This problem **did not converge** when using **quasilinearization**

# → References


- J. T. Allison, T. Guo, and Z. Han (2014). "Co-design of an active suspension using simultaneous dynamic optimization". *J. Mech. Des.* 136.8. doi: 10.1115/1.4027335. DOI: 10.1115/1.4027335
- S. Azad and M. J. Alexander-Ramos (2019). "Reliability-based MDSDO for co-design of stochastic dynamic systems". *International Mechanical Engineering Congress and Exposition*. doi: 10.1115/IMECE2019-10632. DOI: 10.1115/imece2019-10632
- J. F. Bard (1988). "Convex two-level optimization". *Math. Program.* 40-40.1-3. doi: 10.1007/BF01580720. DOI: 10.1007/bf01580720
- J. T. Betts (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. doi: 10.1137/1.9780898718577. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898718577
- A. E. Bryson Jr. and Y.-C. Ho (1975). *Applied Optimal Control*. Taylor & Francis
- D. R. Herber (2017). "Advances in combined architecture, plant, and control design". Ph.D. Dissertation. University of Illinois at Urbana-Champaign. URL: http://hdl.handle.net/2142/99394

# → References (continued)


- D. R. Herber and J. T. Allison (2019). "A problem class with combined architecture, plant, and control design applied to vehicle suspensions". *J. Mech. Des.* 141.10. doi: 10.1115/1.4043312. DOI: 10.1115/1.4043312
- H. Jaddu (2002). "Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials". *J. Franklin Inst.* 339.4-5. doi: 10.1016/S0016-0032(02)00028-5. DOI: 10.1016/s0016-0032(02)00028-5
- A. V. Rao (2010). "A survey of numerical methods for optimal control". *Advances in the Astronautical Sciences* 135.1
- *The DTQP Project* (n.d.). [Online]. url: https://github.com/danielrherber/dt-qp-project

# Questions?

On the Uses of Linear-Quadratic Methods in Solving Nonlinear Dynamic Optimization Problems with Direct Transcription

IMECE2020-23885

Daniel R. Herber
Colorado State University
daniel.herber@colostate.edu
www.engr.colostate.edu/~drherber

---

Athul K. Sundarrajan
Colorado State University
Athul.Sundarrajan@colostate.edu

---

https://github.com/danielrherber/dt-qp-project

∼/examples/IMECE2020-23885