Trajectory Planning for Grade Transitions: A Restricted Form Approach

W.A.M. Gilbert, J.F. Forbes[†], M.G. Lipsett[‡]
[†]Department of Chemical & Materials Engineering
University of Alberta
Edmonton, Alberta, Canada, T6G 2G6
[‡] Syncrude Canada Ltd., Edmonton Research Centre
Edmonton, Alberta, T6N 1H4

Abstract

Trajectory planning is of great importance as many chemical processes normally operate under transient conditions. Many of the available techniques for trajectory optimization can yield trajectories that are unrealistic and not implementable. Furthermore, many transient processes are operated manually, which further reduces the classes of trajectories that are implementable. In this paper, input trajectories are restricted to step/ramp functions, which are easily implemented by manual control or simple control schemes, through the application of input constraints and input switching times. Direct transcription of the dynamic optimization problem was used to allow the solution to be found using NLP techniques. The proposed method also serves as a useful control policy evaluation tool, where comparison of unrestricted and restricted trajectories leads to valuable insight into the gains associated with adopting a more sophisticated control scheme. The restricted inputs approach is applied to several benchmark problems.

1. Introduction

Optimization of transient operations (*e.g.* batch processing, grade transitions, etc.), is crucial to the success of many manufacturers. There is considerable literature on solving these dynamic optimization problems as open-loop control problems. A range of techniques have been used in attempts to solve these problems, including: variational methods, dynamic programming, discretization methods, and other transformation methods. The resulting optimal transition policies are often difficult to implement; and so most operating plants are forced to treat transients in a sub-optimal fashion. A key difficulty with the existing approaches to solving the grade transition problem is that optimal policies are not restricted to take on forms that are readily implementable within the plant. Any general approach to the problem of determining optimal grade transitions must consider the restrictions placed on the allowable form of the input trajectories by the functionality of the plant automation system or operating procedures.

Optimization of industrial operations is characterized by a few to a few hundred degrees of freedom, constraints on the inputs and outputs, and restrictions on the forms that the optimal transition policy can take. Most often, the trajectories to be optimized are the setpoints for the process controllers. These setpoint trajectories must be implemented via Distributed Control Systems (DCS), or manually by an operator. Distributed control systems easily allow the input of simple steps and ramp functions, and are commonly found in almost all industrial processes. Restricting the inputs to steps and ramp functions gives input policies that are easily implemented on existing control systems.

In this work, we build on the ideas of Dabros *et al.*[1] where in optimal input trajectories were restricted to take on the form of a small set of steps. The previous work of Dabros *et al.* treated the dynamic optimization problem by discretizing in time and treating the resulting problem as a quasi-steady state, algebraic optimization problem using conventional nonlinear programming solvers. Additionally, work in this area includes that of Wang *et al.* [14] where the solution of polymerization grade transition problems was done with the inputs restricted to a set of approximated step inputs and solved using control vector parametrization.

There are a range of techniques for solving dynamic optimization problems. Variational methods do not scale well with problem size, due to the evaluation of first and second order optimality conditions, and cannot handle constraints, except in special cases. Dynamic programming [10] also does not scale well with problem size and has a limited range of applicabilty to problems, due to the method's inability to handle inequality constraints. Control vector parametrization, a variant of dynamic programming, requires that the parameter sensitivities of the inputs be calculated analytically, which can be difficult for large and/or highly nonlinear problems. Transformation methods convert infinite optimization problems to algebraic ones through the exploitation of problem structure, however these are limited to a comparatively small class of dynamic optimization problems. Discretization methods can be used on any dynamic optimization problem, and can be solved via a variety of techniques, including iterative dynamic programming(IDP) [9], orthogonal collocation [8] and direct transcription [2]. IDP combines both discretization and dynamic programming methods, and has been successfully used for many small problems. With increasing problem size, IDP problems become large, with the magnitudes for computation costs and times becoming unrealistic. The methods of orthogonal collocation and direct transcription reduce the dynamic optimization problem to that of solving a NLP problem. Using NLP algorithms, it is possible to solve large problems quickly and efficiently.

Orthogonal collocation uses Lagrange polynomials to approximate the trajectories of the states and input functions, where the roots of the polynomial occur at the discretization points. Note that the number of points needed equals the order of the approximating polynomial. The inflexibility of the number of mesh points, makes using this method for the formulation presented below unwieldily. In this work direct transcription and single step or simple multi-step numerical methods are used to remove this complexity.

This paper: 1) presents the structure of the optimal grade transition problem; 2) provides a clear discussion of the solution issues and method and 3) illustrates the proposed approach with three detailed case studies, that have been widely used as benchmark problems in the dynamic optimization literature.

2. Restricted Input Optimal Trajectory Problem

The systems of interest for this paper are those optimal trajectory generation problems with models described by sets of both differential and algebraic equations (DAE) and take the following form

$$\begin{aligned}
&\min_{\mathbf{u}(t)} \quad \Phi\left[\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t\right] \\
&s.t. \\
&\qquad \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \\
&\qquad \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) = 0 \\
&\qquad \mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), t) \le 0 \\
&\qquad \mathbf{x}(t)^L \le \mathbf{x}(t) \le \mathbf{x}(t)^U \\
&\qquad \mathbf{z}(t)^L \le \mathbf{z}(t) \le \mathbf{z}(t)^U \\
&\qquad \mathbf{u}(t)^L \le \mathbf{u}(t) \le \mathbf{u}(t)^U
\end{aligned} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{z}$ are vectors of states with dimension $\mathbb{R}^{n_x}$, $\mathbb{R}^{n_z}$, and $\mathbf{u}$ is a vector of inputs with dimension $\mathbb{R}^{n_u}$, respectively and L and U represent their respective lower and upper bounds. The objective function Φ is a scalar function, which is to be minimized. The vectors $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ each have dimension $\mathbb{R}^{n_f}$, $\mathbb{R}^{n_g}$ and $\mathbb{R}^{n_h}$, respectively. The set of equations $\mathbf{f}$ represents the set of differential equations which describe the system dynamics. The set $\mathbf{g}$ contains any algebraic relationships including constitutive relationships. Any inequality or path constraints are in $\mathbf{h}$.

Restriction of $\mathbf{u}(t)$ to the set $\mathbb{U}$ of n_a analytical functions, where only one function from $\mathbb{U}$ is allowed to occur during any of the m intervals of time for each of the n_u elements of $\mathbf{u}$, requires the use of equilibrium constraints. Simply stated, only one input function is allowed for any input trajectory during a given time interval. To solve the generalized problem, equation (1) must be recast using equilibrium constraints and solved using the methods of mathematical programs with equilibrium constraints (MPEC). The methods of solving MPECs for process engineering has been considered by Raghunathan and Biegler [11] and these methods can be used with existing trajectory optimization techniques to solve the generalized problem.

The MPEC formulation allows the inputs to take on any trajectory; however, special solvers are necessary, making this form of problem not readily solvable. As mentioned previously, most DCS can easily implement step and ramp input trajectories. Limitation of the input trajectories to these simple input functions results in a formulation that can be easily solved using existing NLP solvers. The formalization follows that provided by Betts [2], where changes in all of $\mathbf{x}$, $\mathbf{z}$, $\mathbf{u}$, $\mathbf{f}$, $\mathbf{h}$ and $\mathbf{h}$ are allowed at discrete points in time, termed input switching times (IST). The IST are decided using the prescribed switching intervals, defined as the input interval Δu_i. Each input $u_i(t)$ in $\mathbb{R}^{n_p}$ has associated with it a set of IST $\mathbb{T}_i$, where more than one input can contain the same IST. Figure 1 shows a simple grid of IST for a double input system, with input in-

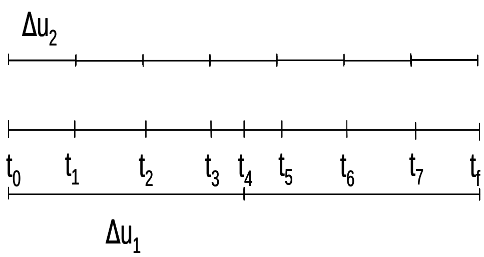


Figure 1. Time Grid Development-2 Inputs

tervals of Δu_1 and Δu_2. Here the sets of IST are

$$\mathbb{T}_1 = \{t_0, t_4, t_f\}$$
$$\mathbb{T}_2 = \{t_0, t_1, t_2, t_3, t_5, t_6, t_7, t_f\}$$

where $\mathbb{T}_1$ and $\mathbb{T}_2$ are the sets of input switching times for $u_1(t)$ and $u_2(t)$, respectively. At any given IST, the active set of inputs, (*i.e.* those with the current IST within their set of IST), are allowed to implement another step or ramp function. Any input not in the active remains unchanged in its present interval. States are assumed to be continuous across all IST, with continuity enforced using equality constraints. Discretizing Problem (1) using the IST grid, and applying the necessary constraints, the following continuous version of the restricted input optimal trajectory (RIOT) problem results

$$\begin{aligned}
&\min_{\mathbf{a}_o^k, \mathbf{a}_1} \quad \Phi\left[\mathbf{x}^k(t), \mathbf{z}^k(t), \mathbf{u}^k(t), t^k\right] \\
&s.t. \\
&\quad \mathbf{f}(\dot{\mathbf{x}}^k(t), \mathbf{x}^k(t), \mathbf{z}^k(t), \mathbf{u}^k(t), t^k) = 0 \\
&\quad \mathbf{g}(\mathbf{x}^k(t), \mathbf{z}^k(t), \mathbf{u}^k(t), t^k) = 0 \\
&\quad \mathbf{h}(\mathbf{x}^k(t), \mathbf{z}^k(t), \mathbf{u}^k(t), t^k) \leq 0 \\
&\quad u_j^k = a_{o_j}^k + a_{1_j}^k t \quad t_k \in \mathbb{T}_j \\
&\quad u_j^k = u_j^{k-1} \quad t_k \notin \mathbb{T}_j \\
&\quad \mathbf{x}_F^{k-1} = \mathbf{x}_I^k \quad k = 1, .., h \\
&\quad \mathbf{z}_F^{k-1} = \mathbf{z}_I^k \quad k = 1, .., h \\
&\quad \mathbf{x}(t)^L \leq \mathbf{x}^k(t) \leq \mathbf{x}(t)^U \\
&\quad \mathbf{z}(t)^L \leq \mathbf{z}^k(t) \leq \mathbf{z}(t)^U \\
&\quad \mathbf{u}(t)^L \leq \mathbf{u}^k(t) \leq \mathbf{u}(t)^U \\
&\quad j = 1..p
\end{aligned} \tag{2}$$

where $\mathbf{a}_o$ and $\mathbf{a}_1$ are vectors of time independent parameters for the step and ramp formalization, which become the decision variables of interest in this problem. Here the objective function, differential, equality and inequality constraints remain unchanged. Following these equations the input restriction and continuity equations are included. Other possible constraints which may be included are smoothness of the input trajectory, as well as rate limits on inputs. The above form of the RIOT problem allows both step and ramp functions; however, reformulating the problem to include either function is trivial, where step functions alone can be achieved simply by seting $\mathbf{a}_1 = 0$. Ramp functions only require $\mathbf{a}_0 = 0$, with the addition of continuity constraints between IST.

As stated, problem (2) cannot be solved using dynamic programming techniques [10], and discretization must be used to reduce the problem to a NLP. Discretizing time into q mesh points in each input interval, the RIOT problem reduces to the following form

$$\begin{aligned}
&\min_{\mathbf{a}_o^k, \mathbf{a}_1} \quad \Phi\left[\mathbf{x}_i^k, \mathbf{z}_i^k, \mathbf{u}_i^k, t_i^k\right] \\
&s.t. \\
&\quad \mathbf{f}(\dot{\mathbf{x}}_i^k, \mathbf{x}_i^k, \mathbf{z}_i^k, \mathbf{u}_i^k, t_i^k) = 0 \\
&\quad \mathbf{g}(\mathbf{x}_i^k, \mathbf{z}_i^k, \mathbf{u}_i^k, t_i^k) = 0 \\
&\quad \mathbf{h}(\mathbf{x}_i^k, \mathbf{z}_i^k, \mathbf{u}_i^k, t_i^k) \leq 0 \\
&\quad u_j^k = a_{o_j}^k + a_{1_j}^k t, \quad t_k \in \mathbb{T}_j, \quad \forall i \\
&\quad u_j^k = u_j^{k-1}, \quad t_k \ni \mathbb{T}_j, \quad \forall i \\
&\quad \mathbf{x}^L \leq \mathbf{x}_i^k \leq \mathbf{x}^U \\
&\quad \mathbf{z}^L \leq \mathbf{z}_i^k \leq \mathbf{z}^U \\
&\quad \mathbf{u}^L \leq \mathbf{u}_i^k \leq \mathbf{u}^U \\
&\quad \Delta t^L \leq \Delta t^k \leq \Delta t^U \\
&\quad \mathbf{x}^{k-1} = \mathbf{x}^k \\
&\quad \mathbf{z}^{k-1} = \mathbf{z}^k \\
&\quad i = 1, .., q \quad j = 1..p \quad k = 1, .., h
\end{aligned} \tag{3}$$

where $q \geqslant 2$, a requirement that must be met to satisfy the presence of IST bounds and to ensure input restriction. In this form the RIOT problem still may contain functions of derivatives and integrals in the state variables. The method of direct transcription [2] is best suited for solving RIOT problems, where differential/integrals terms are directly transcribed using commonly used numerical methods. NLP techniques can then be used to solve Problem (3).

3. Restricted Input Optimal Trajectory Method

The RIOT method can be stated easily using the following steps:

1. Formulate problem as a desired dynamic opimization problem of the form given in (1).

2. Determine the IST grid using the predefined values of the input intervals for all inputs.

3. Restrict inputs using equality constraints, treating those inputs which are in the active set, and inactive set appropriately.
4. Introduce continuity constraints at the IST for all states, the problem is now in the form of Problem (2).
5. Discretize the continuous problem into a grid of mesh points, and use direct transcription to remove any derivative and integral terms.
6. Solve using NLP techniques.
7. If error tolerances are not met, refine mesh and/or increase order of numerical method, then go back to step 6.

Mesh refinement can be accomplished by using some sort of refinement rules (*e.g.* rules used for the upcoming studies are taken from Betts and Huffman [3]).

4. Case Studies

The method presented above will be illustrated using three commonly studied problems: 1) the single integrator system, examined by Goh and Teo [4], Guay *et al.* [12] and Luus [5]; 2) the consecutive reaction problem; and 3) the parallel reaction problem both examined by Ray [13].

The problems were solved using Boeing's® Sparse Optimal Control Software (SOCS), using sequential quadratic programming or interior point solvers, and exploiting problem sparsity to efficiently solve optimal control problems. All problems were solved to meet a the minimum objective function and ODE tolerance allowable in SOCS, and were initially solved using the trapezoidal rule. Mesh refinement and numerical methods were then allowed to change freely as mesh refinement continued. Optimizations were done using a dual AMD 2400+ with 1 Gb RAM.

4.1. Single Integrator Problem

An analytical solution for the single integrator has been solved via linear least squares optimal control theory [15]. The single integrator problem can be stated as

$$\begin{aligned} &\min_{u(t)} \quad x_2(t_f), \quad t_f = 1 \\ &\text{s.t.} \\ &\quad \dot{x_1}(t) = u(t) \\ &\quad \dot{x_2}(t) = x_1^2(t) + u^2(t) \\ &\quad x_1(0) = 1 \\ &\quad x_2(0) = 0 \\ &\quad x_1(1) = 1 \end{aligned} \tag{4}$$

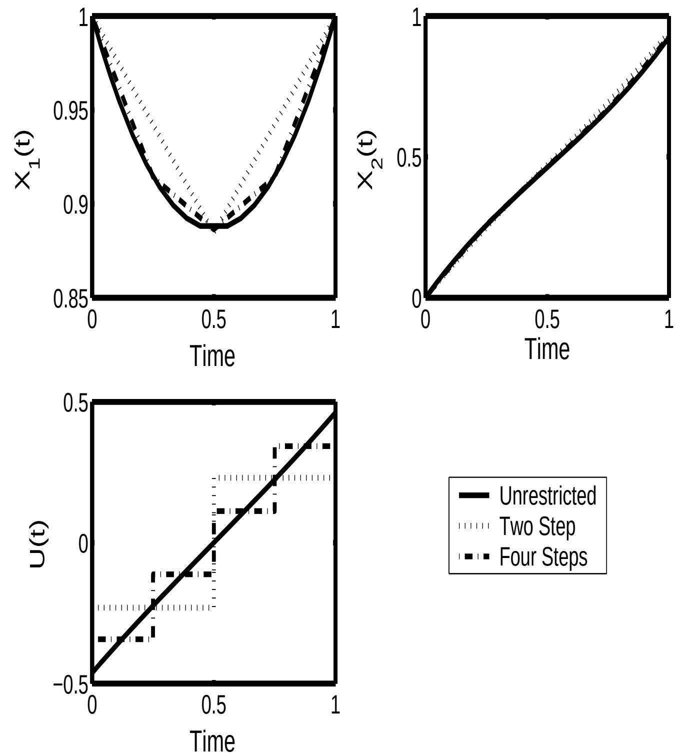


Figure 2. Optimal Solution of Integrator Problem

Restricting the inputs of the integrator system to be step functions, and using direct transcription, the RIOT optimization problem can be reformulated to give

$$\begin{aligned} &\min_{\beta} \quad x_2(t_f), \quad t_f = 1 \\ &\text{s.t.} \\ &\quad \dot{x}_1^k(t) = u^k(t) \\ &\quad \dot{x}_2^k(t) = (x_1^k)^2(t) + (u^k)^2(t) \\ &\quad u^k(t) = \beta_k \\ &\quad x_{1_I}^{k+1} = x_{1_F}^k \\ &\quad x_{2_I}^{k+1} = x_{2_F}^k \\ &\quad x_1(0) = 1 \\ &\quad x_2(0) = 0 \\ &\quad x_1(1) = 1 \\ &\quad k = 1, .., n \end{aligned} \tag{5}$$

Figure 2 shows the optimal trajectories of both the restricted and unrestricted problems. For the unrestricted problem, the optimal value of the objective function, $x_2^*(t_f)$, was found to agree to within 6 significant figures of the analytical solution to the problem. All RIOT problem solutions were found to be less optimal. For the case where only one step function is allowed the degrees of freedom become zero and the solution reduces to $u(t) = 0$.

4.2. Consecutive Reaction Problem

The consecutive reaction problem, first examined by Ray [13], McAuley and Dadebo [6], and Guay et al. [12] follows the following reaction

$$\mathrm{A} \underset{k_1}{\rightarrow} \mathrm{B} \underset{k_2}{\rightarrow} \mathrm{C}$$

It is desired to maximize the recovery of the middle product by manipulation of the reactor temperature. The problem can be stated mathematically as

$$\begin{aligned}
&\max_{u(t)} \quad x_2(t_f), \quad t_f = 1 \\
&\text{s.t.} \\
&\qquad \dot{C_A}(t) = -u(t)C_A^2(t) \\
&\qquad \dot{C_B}(t) = -0.03875u^2(t)C_B(t) + u(t)C_B^2(t) \\
&\qquad C_A(0) = 1 \\
&\qquad C_B(0) = 0 \\
&\qquad 0.9092 \leq u(t) \leq 7.4831
\end{aligned} \tag{6}$$

where the input $u(t)$ is defined as $u \equiv 4000e^{-2500/T}$. Restricting the input $u(t)$ to take on n steps, the problem then becomes the following

$$\begin{aligned}
&\max_{\beta} \quad x_2(t_f), \quad t_f = 1 \\
&\text{s.t.} \\
&\qquad \dot{C_a}^k(t) = -u(t)^k(C_a^k)^2(t) \\
&\qquad \dot{C_b}^k(t) = -0.03875(u^k)^2(t)C_b^k(t) + u^k(t)(C_a^k)^2(t) \\
&\qquad u^k(t) = \beta_k \\
&\qquad C_{A_I}^{k+1} = C_{A_F}^k \\
&\qquad C_{B_I}^{k+1} = C_{B_F}^k \\
&\qquad C_A(0) = 1 \\
&\qquad C_B(0) = 0 \\
&\qquad 0.9092 \leq u^k(t) \leq 7.4831 \\
&\qquad k = 1, .., n
\end{aligned} \tag{7}$$

where I and F refer to the beginning and end of the current input interval. The optimal trajectories for both the restricted and unrestricted problems are shown in Figure 3. The optimal conversion, $C_b^*(t_f)$ for the unrestricted trajectory problem was found to be 0.610803, in good agreement with those published in the literature. All RIOT formalizations had optimal solutions less than that of the restricted problem.

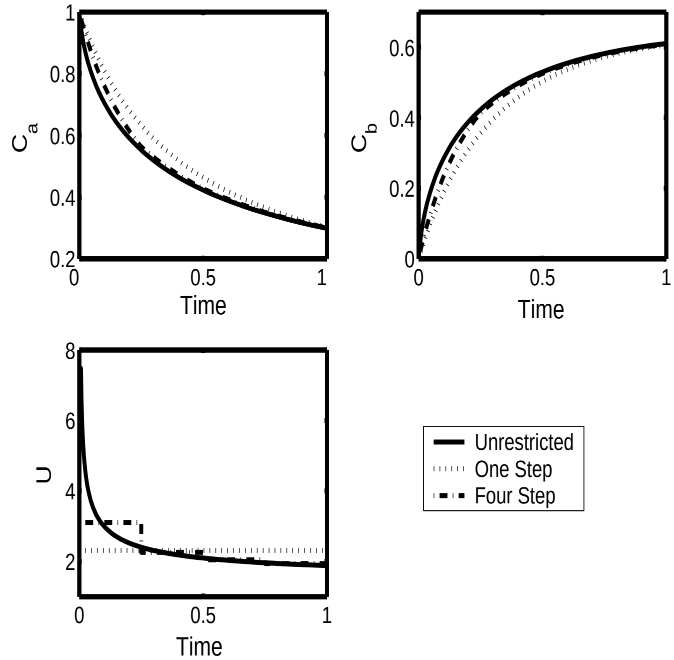


Figure 3. Optimal Trajectory of Consecutive Reaction

4.3. Parallel Reaction Problem

The parallel reaction problem presented here has been examined thoroughly in the literature by Beigler [16], Ray [13] and others [17] [18] [6]. Here two reactions are taking place within a tubular reactor, which are stated as

$$\mathrm{A} \underset{k_1}{\rightarrow} \mathrm{B}$$
$$\mathrm{A} \underset{k_2}{\rightarrow} \mathrm{C}$$

Defining the dimensionless concentrations of species A and B in the reactor as $x_1 \equiv C_A/C_{A_o}$ and C_B/C_{A_o} where C_{A_o} is defined as the initial concentration of A, the following maximization problem results

$$\begin{aligned}
&\max_{u(t)} \quad x_2(t_f), \quad t_f = 1 \\
&\text{s.t.} \\
&\qquad \dot{x_1}(t) = -u(t)x_1(t) - \frac{1}{2}u^2(t)x_1(t) \\
&\qquad \dot{x_2}(t) = u(t)x_1(t) \\
&\qquad x_1(0) = 1 \\
&\qquad x_2(0) = 0 \\
&\qquad 0 \leq u(t) \leq 5
\end{aligned} \tag{8}$$

where $u(t) \equiv k_1L/v$, L is the reactor length, and v is the spacial velocity. Dividing the problem into n_p in-

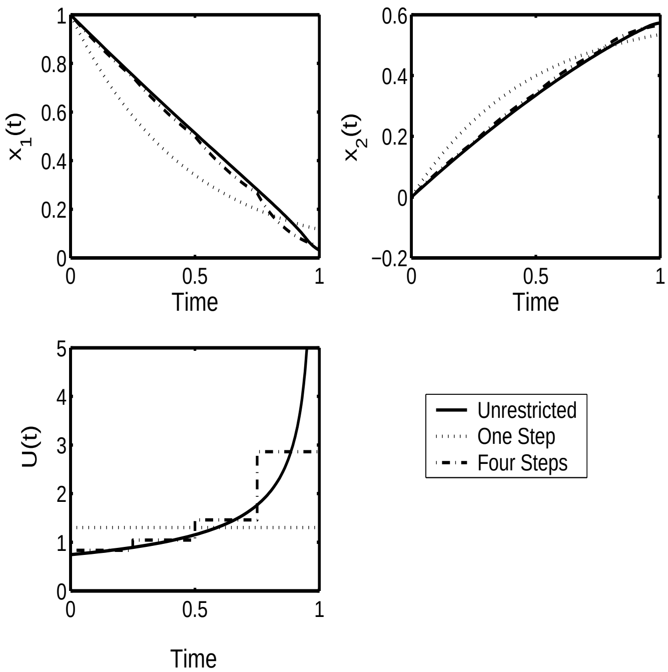


Figure 4. Optimal Trajectory of Parallel Reaction

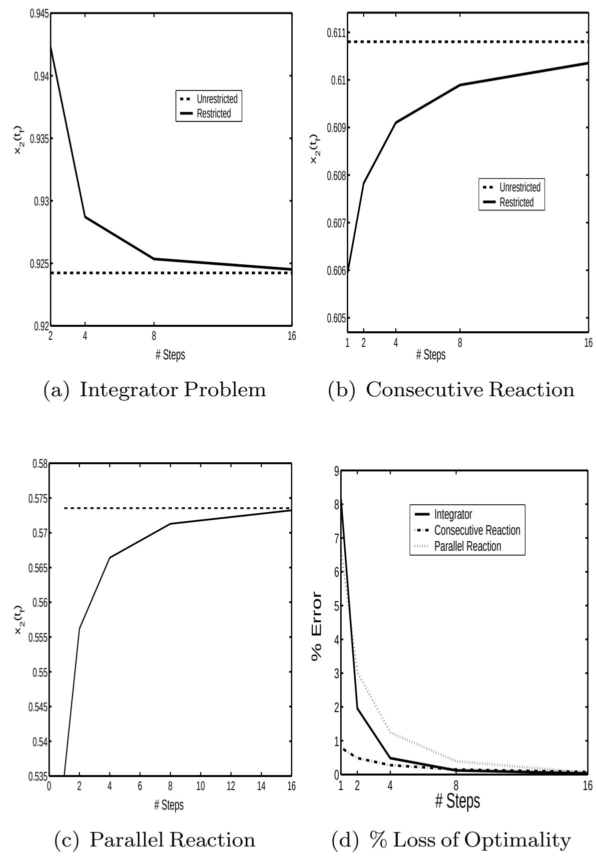
put intervals, and recasting in continuous RIOT form

$$
\begin{aligned}
&\max_{\beta} \quad x_2(t_f), \quad t_f = 1 \\
&\text{s.t.} \\
&\qquad \dot{x_1}^k(t) = -u(t)^k x_1^k(t) - \frac{1}{2}(u^k(t))^2 x_1^k(t) \\
&\qquad \dot{x_2}^k(t) = u(t)^k x_1^k(t) \\
&\qquad u^k(t) = \beta_k \\
&\qquad x_{1_I}^{k+1} = x_{1_F}^k \\
&\qquad x_{2_I}^{k+1} = x_{2_F}^k \\
&\qquad x_1(0) = 1 \\
&\qquad x_2(0) = 0 \\
&\qquad 0.9092 \le u^k(t) \le 7.4831 \\
&\qquad k = 1, .., n_p
\end{aligned}
\tag{9}
$$

Both problems (8) and (9) were then discretized and direct transcription was used to convert to a NLP. The optimal value of $x_2^*(t_f) = 0.573545$, corresponding well with those presented in literature. All restricted solutions were found to be less optimal than their unrestricted counterparts. The optimal trajectories for the unrestricted, 1–step and 4–step restricted cases are shown in Figure 4.

5. Results and Discussion

As stated above, all RIOT problem solutions to the integrator, consecutive, and parallel reaction problems were found to be less optimal then their unrestricted counterparts. Comparison of the objectives for both the unrestricted and restricted inputs are compared in Figure 5 for all three case studies. It appears that, as the

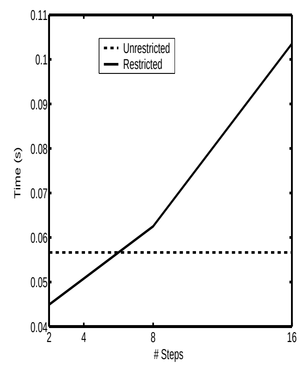


(a) Integrator Problem (b) Consecutive Reaction (c) Parallel Reaction (d) % Loss of Optimality

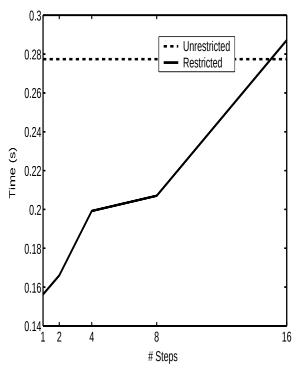
Figure 5. Objective Functions and Error of Case Studies

number of steps are increased, the RIOT formulation approaches the optimal value of the unrestricted case asymptotically. The RIOT problems with two allowable steps show approximately 1–3% difference when compared to the unrestricted solutions, with the loss becoming negligible for both the 8 and 16 step cases. For optimal cases, where the input was held constant for all time, large deviations from the unrestricted optimum were found to occur for the integrator and parallel reaction problems.

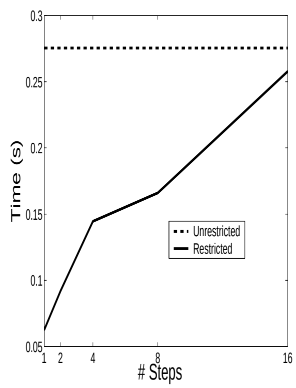
Restriction of the inputs was found to affect the computational efficiency of the problem. Figure 6 shows the effects of input restriction on the computation time. All problems are found to have a monotonically increasing relationship between the computational time and the number of steps. Interestingly for highly restricted systems (low number of steps), the computational costs were below that of the unrestricted problem. This suggests that it may be possible to reduce optimization costs by reformulation of a problem in RIOT form. For highly restrictive problems the degrees of freedom are lowered and the number of additional constraints added to the system are minimal, which is expected to give rise to shorter solution times. As the number of allowable steps is relaxed, the problem gains more degrees of freedom and an increasing number of equality constraints, which leads to large solution times. For the parallel reaction problem, even the 16–step restricted input computational costs were below that of the unrestricted case. The number of mesh refinement iterations necessary was found to increase with increasing number of steps, and for both problems the number of mesh refinements became larger than that necessary for the unrestricted case. Decreasing the input restriction also led to an increase in the overall number of mesh points.



(a) Integrator



(b) Consecutive Reaction



(c) Parallel Reaction

Figure 6. Solution Times of the Case Studies

Input saturation was found to occur on both the unrestricted cases for the consecutive and parallel reactions; however, restriction of the inputs for these two cases was found to remove input saturation. The necessary range for the inputs was found to shrink for all three problems studied. This suggests that the RIOT formalization has a natural back off on the magnitude of the inputs.

The RIOT problem also provides a useful tool in benefits analysis for determining whether implementation of advanced control policies would result in large gains to a process. Comparison of the unrestricted and restricted optima, when combined with cost analysis, may provide useful information into whether substantial gains exist for more sophisticated controllers structures.

6. Conclusions

This paper proposed a restricted input approach to solving dynamic optimization problems, using direct transcription. Dynamic optimization techniques can produce input trajectories that are too complicated to implement using existing control policies. The method presented above addresses this problem by restricting the input trajectories to step and ramp functions, easily implemented using Distributed Control Systems. The proposed method builds on existing dynamic optimization theory, using direct transcription to the problem solving using NLP techniques.

The method was found to offer computational advantages, exploiting low degrees of freedom for problems with low numbers of steps and ramp functions. All RIOT problems were found to be less optimal then their unrestricted counterparts, providing a useful control system evaluation tool, where large differences may provide incentives for adopting advanced control policies. Furthermore, restriction of the inputs reduced the aggressiveness of the control action, where unrestricted trajectories with input saturation were found to operate within operating bounds.

References

[1] Dabros, M., P. Stuart , J.F. Forbes, M. Perrier, M. Fairbank, Improving the Broke Recirculation Strategy in a Newsprint Mill, *Pulp & Paper Canada*, to appear (2003).

[2] Betts, J.T., *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM (2001).

[3] Betts, J.T., W.P. Huffman, Mesh refinement in direct transcription methods for optimal control *Optim. Control Appl. Methods*, **19**, (1998), 1–21.

[4] Goh, C.J., L.K. Teo, Control Parametrization: A Unified Approach to Optimal Control Problems with General Constraints *Automatica* , **24**, (1988),3–18.

[5] Luus, R., Application of iterative dynamic programming to state constrained optimal control problems *Hung. J. Ind. Eng.*, (1991), 245–254.

[6] Dadebo, S. A., K. B. McAuley, Dynamic Optimization of Constrained Chemical Engineering Using Dynamic Programming *Comput. Chem. Eng.*, **19**(5), (1995), 513–525.

[7] Cuthrell, J.E., L.T. Biegler, Simultaneous Optimization and Solution Methods for Batch Reactor Control Profiles *Comp. Chem. Eng.*, **13**(1/2), (1989), 49–62.

[8] Cuthrell, J.E., L.T. Biegler, On the Optimization of Differential-Algebraic Process Systems *AIChE*, **33**(8), (1987), 1257–1270.

[9] Luus, R. *Iterative Dynamic Programming*, Chapman and Hall/CRC Monographs and Surveys in Pure and Applied Mathematics, (2000).

[10] Bellman, R.E., *Dynamic Programming*, Princeton University Press, (1958).

[11] Raghunathan, A.U., L.T. Biegler, Mathematical Programs with Equilibrium Constraints (MPECs) in Process Engineering *Comp. & Chem. Eng.*, **27**, (2003), 1381–1392.

[12] Guay, M., S. Kansal, J.F. Forbes, Trajectory Optimization for Flat Dynamical Systems *Ind. Eng. Chem. Res.*, **40**, (2001), 2089–2102.

[13] Ray, W.H. *Advanced Process Control* McGraw Hill, New York, 1981.

[14] Wang, Y., H. Seki, S. Ohyama, K. Akamatsu, M. Ogawa, M. Ohshima, Optimal Grade Transition Control for Polymerization Reactors *Comp. Chem. Eng.*, **24**, (2000), 1555–1561.

[15] Brokett, R.W. *Finite Dimensional Linear Systems* Wiley, New York, 1969.

[16] Biegler, L.T., Solution of Dynamic Optimization Problems by successive quadratic programmin and orthogonal collocation. *Comp. Chem. Eng.*, **8**(3/4), (1984), 243–248.

[17] Logsdon, J.S., L.T. Biegler, Accurate Solution of differential algebraic equations. *Ind. Chem. Eng. Res.*, **28**, (1989), 1628–1639.

[18] Logsdon, J.S., L.T. Biegler, Decomposition Strategies for Large Scale Dynamic Optimization Problems. *Chem. Eng. Sci.*, **47**, (1992), 851–864.