



Cedarville University
DigitalCommons@Cedarville

Engineering and Computer Science Faculty
Publications

School of Engineering and Computer Science

2014

Implementation Lessons and Pitfalls for Real-time Optimal Control with Stochastic Systems

S. M. Ross

R. G. Cobb

W. P. Baker

Frederick G. Harmon

Cedarville University, fharmon@cedarville.edu

Follow this and additional works at: [http://digitalcommons.cedarville.edu/
engineering_and_computer_science_publications](http://digitalcommons.cedarville.edu/engineering_and_computer_science_publications)

 Part of the [Engineering Commons](#)

Recommended Citation

Ross, S. M.; Cobb, R. G.; Baker, W. P.; and Harmon, Frederick G., "Implementation Lessons and Pitfalls for Real-time Optimal Control with Stochastic Systems" (2014). *Engineering and Computer Science Faculty Publications*. 195.
http://digitalcommons.cedarville.edu/engineering_and_computer_science_publications/195

This Article is brought to you for free and open access by DigitalCommons@Cedarville, a service of the Centennial Library. It has been accepted for inclusion in Engineering and Computer Science Faculty Publications by an authorized administrator of DigitalCommons@Cedarville. For more information, please contact digitalcommons@cedarville.edu.



Implementation lessons and pitfalls for real-time optimal control with stochastic systems

Steven M. Ross¹, Richard G. Cobb^{1,*}, William P. Baker² and Frederick G. Harmon¹

¹*Department of Aeronautics and Astronautics, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45433, USA*

²*Department of Applied Mathematics, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45433, USA*

SUMMARY

Modern computational power and efficient direct collocation techniques are decreasing the solution time required for the optimal control problem, making real-time optimal control (RTOC) feasible for modern systems. Current trends in the literature indicate that many authors are applying RTOC with a recursive open-loop structure, relying on a high recursion rate for implicit state feedback to counter disturbances and other unmodeled effects without explicit closed-loop control. The limitations of using rapid, instantaneous optimal solutions are demonstrated analytically and through application to a surface-to-air missile avoidance control system. Two methods are proposed for control structure implementation when using RTOC to take advantage of error integration through either classical feedback or disturbance estimation. Published 2014. This article is a U.S. Government work and is in the public domain in the USA.

Received 29 September 2010; Revised 26 July 2012; Accepted 16 January 2014

KEY WORDS: real-time optimal control; stochastic optimal control; optimal feedback control

NOMENCLATURE

\mathbf{D}	= derivative matrix
e_p	= orthogonal position error (unit length)
$E[\cdot]$	= expected value operator
f	= state dynamics
\mathcal{H}	= Hamiltonian
J	= cost functional (unit time)
k_p, k_i	= proportional, integral feedback gains (unit length/radian)
\mathcal{L}	= running cost (Lagrangian)
n_{SAM}	= total known surface-to-air missile locations
N	= total number of collocation nodes
Q	= noise strength
t	= time (unit time)
u	= control
\mathbf{U}	= polynomial approximation of control vector
V_{ac}	= total aircraft velocity (unit length/unit time)
w	= wind disturbance (unit length/unit time)
\hat{w}	= estimated wind speed (unit length/unit time)

*Correspondence to: Richard G. Cobb, Department of Aeronautics and Astronautics, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45433, USA.

†E-mail: richard.cobb@afit.edu

- w_k = Legendre–Gauss weight
 x, y = x -axis, y -axis position coordinates (unit length)
 \mathbf{x}_{ac} = aircraft position state vector
 \mathbf{X} = polynomial approximation of state vector
 Δt = time step (unit time)
 $\delta(\cdot)$ = delta function operator
 η = stochastic noise input
 Γ = Mayer cost functional
 $\theta, \theta_e, \theta_{fb}, \theta_t$ = aircraft heading (rad)—commanded, instantaneous error, feedback, and true
 λ_n = Lagrange multiplier
 P_N = Legendre polynomial of degree N
 ξ = integration variable
 ρ_i = effective range of i^{th} surface-to-air missile
 τ = time, transformed to $[-1, 1]$

Subscripts

- ac = aircraft
 f = final
 $gust$ = portion of wind attributable to gusts
 k = epoch
 $meas$ = measured value
 $pred_wind$ = predominant portion of wind
 ref = reference trajectory or control
 x, y = component in the x -axis or y -axis direction
 0 = initial

1. INTRODUCTION

Great strides have been made recently in the field of real-time optimal control (RTOC). The combination of recent advances in processing power with efficient solution techniques has opened doors for the use of several methods of RTOC that were previously infeasible because of longer computational times. As RTOC structures begin to see integration into physical systems, it is critical that the method of implementation be examined in the light of classical control theory principles in order to avoid potential design pitfalls. The purpose of this paper is to examine, for the general case, one of these potential oversights—the loss of error integration when RTOC is applied with a fast open-loop recursive structure.

1.1. Solution methods

As a generalization of calculus of variations, optimal control solutions are sought after for the obvious benefits of providing the *best* method from a set of feasible alternatives—the fastest route, the least fuel burned, the maximum time aloft, etc. Stated in the general Bolza form, the optimal control problem is to determine the state-control function pair, $\{\mathbf{x}(t), \mathbf{u}(t)\}$, and final time, t_f , which minimize the cost functional:

$$J = \Gamma(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (1)$$

subject to the dynamic constraints

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (2)$$

the path constraints

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t_0, t_f) \geq \mathbf{0} \quad (3)$$

and the boundary conditions

$$\boldsymbol{\gamma}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) \geq \mathbf{0} \quad (4)$$

with equality constraints imposed via a second constraint on the additive inverse.

Unfortunately, optimal control problems are notoriously difficult to solve analytically, and for nonlinear systems, it is rare to achieve a closed-loop optimal feedback law. Traditional application of optimal control has involved applying an optimal control time history in an open-loop manner with an additional closed-loop feedback law around the error between the actual and expected optimal state paths. The optimal open-loop control is numerically solved for off-line prior to its need, through any of several direct or indirect methods of optimization [1].

Indirect methods involve determining extremals with the Hamiltonian and first-order optimality conditions [2, 3], and have traditionally been a relatively slow process, unsuitable for RTOC applications. In addition, indirect methods are often not usable for problems with unknown boundary conditions because of extreme sensitivity [4], and they require a (generally nonintuitive) guess of the initial states and costates—often with a smaller radii of convergence around the true answer than direct methods, which can be more forgiving of the initial state guess [5]. If the problem is constrained, the user must break the indirect method into constrained and unconstrained arcs, which may not be known *a priori* [6].

For RTOC, direct methods have been used to take advantage of their relative computational efficiency. Although dormant for quite some time, direct methods were initially proposed by Euler [7], and involve transcribing the infinite-dimensional, continuous optimal control problem into a finite-dimensional, nonlinear programming problem (NLP), which in modern days is solved numerically [8–10].

The underlying technique of transcription, or collocation, is performed by approximating the state vector with a discrete number of variables with any of a number of methods (e.g., polynomial interpolation and coefficients of a Fourier series). The continuous *dynamic* constraints for the system are then evaluated at select collocation points, often called nodes, producing a discrete (although sometimes very large) set of *static* equations consisting of one equation for every state, at every node [11]. The result is a new static optimization problem, with the same cost function (in a discretized form), and a set of static constraints that ensure the dynamics of the original optimal control problem are enforced. This NLP can then be solved with a host of applications designed for that purpose, most of which employ sequential quadratic programming as the primary solution method. Additional computation time can be saved when matrix sparseness is accounted for by using a reduced-Hessian semi-definite quadratic solver, as is performed in the industry standard, Sparse Nonlinear Optimizer (SNOPT) [12]. There is a large variety in direct methods with respect to which approximation techniques to use, and which points to use for collocation nodes. Problem specific characteristics often determine which methods are most suitable, and there are several software packages available that ease the transcription process. For the solutions in this paper, efficient direct techniques known as pseudospectral methods are employed, as implemented in programs such as DIDO or General Pseudospectral Optimal Control Software (GPOPS) [13–15].

Regardless of the method selected to solve the problem, once the optimal control time history is known, the traditional application of nonlinear optimal control is to first apply that control to the system in a feed-forward, open-loop manner. Once the control is applied, the difference between the actual and predicted optimal states is tracked, and perturbations that arise from unmodeled effects such as disturbances, measurement errors, or simplifications of the system dynamics are minimized or eliminated with the addition of an error feedback control law. This feedback can be provided by conventional feedback laws for the plant linearized about the reference trajectory, by time-varying gains produced by a neighboring optimal controller [16], or a myriad of other techniques.

2. RECURSIVE OPEN-LOOP CONTROL WITH TRAJECTORY ERROR FEEDBACK

To move toward a solution that is truly optimal in real time, an additional DOF can be added by re-solving the optimal control problem in a recursive fashion. Upon achieving an optimal solution,

the optimal control, \mathbf{u}_{ref} , is applied. As the state begins to change, a new solution epoch, k , is computed, using the expected state from the previous state solution, $\mathbf{x}_{ref_{k-1}}$, at the time the solution is expected to be available, t_k , for the initial conditions [17]:

$$\mathbf{x}_k(t_0) = \mathbf{x}_{ref_{k-1}}(t_k) \tag{5}$$

This method typically includes ‘bootstrapping’ the portion of the solution that will be remaining beyond the new initial conditions as the seed for the next guess. Two basic control methods are then typically followed—the first being a closed-loop solution of some sort that tracks the reference solution, such as

$$\mathbf{u}(t) = \mathbf{u}_{ref} + \mathbf{K}\{\mathbf{x}_{ref}(t) - \mathbf{x}(t)\} \tag{6}$$

and the second option includes applying purely the open-loop control, quickly updating the reference solution as path errors develop:

$$\mathbf{u}_k(t) = \mathbf{u}_{k_{ref}}(t) \tag{7}$$

The idea of a recursive open-loop solution is compelling—once perturbed from the initial optimal path, why waste control effort working back toward the original reference trajectory? Figure 1 illustrates the situation. Once the state is perturbed from the expected optimal path, \mathbf{x}_{ref} , correcting back to that trajectory is likely not optimal from the disturbed position, and a new path originating from the current state should be introduced. Re-solving the optimal control problem as often as possible, and maintaining that reference path through a faster, inner control loop, results in a two DOF design, such as the one shown in Figure 2, which can be found in similar forms in [17] and [18].

Recently, several authors have taken the speed advantages of efficient optimization techniques and increased processing power to move the control concept one step further—eliminating the inner loop altogether and controlling in a purely recursive open-loop manner. Conceptually, if you have control at any point that you have defined as optimal, why would you add anything to it? This is further supported by the fact that the outer loop does not merely provide a scalar control value that is held until the next update (as in a zero-order-hold system), but instead provides a control time history as a function, $\mathbf{u}_{ref}(t)$, which allows much more freedom with computation time, and allows something of a fail-safe plan that will continue to completion even in the face of a loss of sensor data or the ability to calculate new trajectories [19].

It is tempting to draw the conclusion that if the recursive open-loop optimal control can just be solved fast enough, there is no need for feedback, or that the recursive open-loop control can be

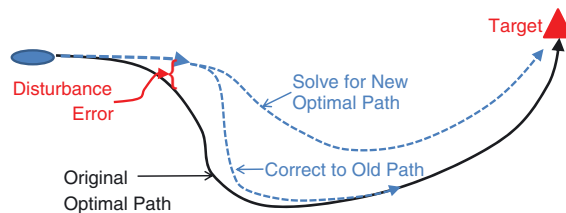


Figure 1. Decision to follow initial optimal trajectory or to re-solve the optimal path from the current condition.

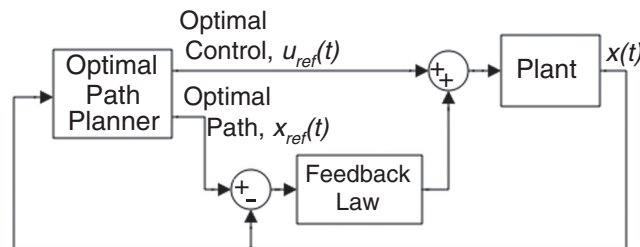


Figure 2. Two DOF control scheme.

equated to feedback control. Many scholars who have provided great advancement in the field of RTOC have supported this concept in recent literature. For instance, in control guidance for a car, an author stated:

‘The feedback law is not analytically explicit; rather, closed-loop control is obtained by a rapid re-computation of the open-loop time-optimal control at each update instant.’ [20]

Another researcher used a similar technique for satellite guidance, again suggesting that rapid open-loop control would provide the disturbance rejection of closed-loop feedback:

‘A conceptually simple approach to controlling such nonlinear systems is by solving the problems online. If such problems can be solved online, there is no need for an off-line design of closed-form feedback laws as, by definition, the control system would have acquired this intelligence....Rather than tracking a pre-computed solution, the control scheme proposed in this paper re-solves the optimal control problem and updates the control command as soon as a new solution is obtained. This results in a sampled-data feedback law which provides optimality in the presence of various types of disturbances.’ [21]

Others equated rapid open-loop control to closed-loop control for re-entry vehicles:

‘The key for successful implementation of these feedback principles rely on a sufficiently fast generation of open-loop controls. Thus, if open-loop controls can be generated as demanded by [a given speed requirement for his problem], closed loop is achieved quite simply.’ [22]

In an outstanding foundational work on RTOC, it was claimed simply:

‘Suppose optimal open-loop controls could be computed in real time. This implies optimal feedback control.’ [19]

Finally, another proposed the same premise:

‘It has been known since the birth of optimal control that if open-loop controls can be generated in real-time, they are basically equivalent to feedback controls.’ [18]

The purpose of quoting several authors who have each made great contributions to the field of RTOC is to show the pervasiveness of the concept in the optimal control community that recursive open-loop optimal control, performed quickly enough, is equivalent to optimal closed-loop feedback control. The resulting control structure eliminates the need for the inner loop of Figure 2. While there certainly is a level of feedback that is implicitly achieved, purporting that that recursive open-loop control implies ‘optimal feedback control’ suggests that there is no further control desired or required—in fact, the addition of any other control would, by definition, no longer be optimal. Inner-loop control is therefore seen as detrimental, and removed.

3. LACK OF ERROR INTEGRATION IN INSTANTANEOUS OPTIMAL SOLUTIONS

The basic recursive open-loop structure has shown success in simulations for the earlier problems, but real-world considerations may indicate that perhaps this should not be the baseline design for RTOC. It is the intent of this paper to show that lessons from classical control theory suggest significant limitations of a purely recursive open-loop structure that must be considered by the designer in building a real-time optimal controller. The single DOF design, removing the inner feedback loop, recursively provides an instantaneous optimal solution for the control and state at several steps in the process (the faster, the better, in theory). However, if the designer of an RTOC system makes the assumption that rapid open-loop solutions yield the same performance as traditional feedback control, the resulting design will fail to leverage the information that can be gleaned from comparing the historical efforts to outcomes.

Although it seems paradoxical to claim that we can do better than optimal control, the answer can only be as good as the assumptions. The optimality of any solution is based only on the assumed system dynamics, and lacks consideration of the path and control history. The whole question of RTOC implies that there are disturbances or unmodeled effects to be rejected. If the true dynamics

were all known perfectly, the optimal solution would only need to be found once, rather than in real time. Recursively solving the problem gives freedom to respond to stochastic or unanticipated effects. Especially for cases where these disturbances end up not falling into the classic categories of Gaussian, white, and zero-mean, integration of the error between the expected and actual states and control history can supply either additional compensation, or a more accurate model of the true system dynamics through estimation of the disturbance. If the likely errors for the system are indeed nonzero mean, or at least time correlated (and thus likely nonzero mean over some time interval), these effects should be accounted for in the selection of the control. This requires one of many methods of feedback control that are not achieved with a purely recursive open-loop design. Two nonlinear optimal control problems are posed to demonstrate this principle. The first is an overly simplified course guidance problem to allow analysis and a counterexample via an analytic solution. The second case study expands the problem to a realistic threat-avoidance scenario that is numerically solved with RTOC methods.

4. CASE STUDY A: AIRCRAFT COURSE PLANNING

Consider an aircraft simply modeled as a point mass system with rectilinear position components:

$$\mathbf{x}_{ac}(t) = \begin{bmatrix} x_{ac}(t) \\ y_{ac}(t) \end{bmatrix} \quad (8)$$

The aircraft is flying at a constant altitude, with a constant velocity, V_{ac} . The pilot has been cleared direct to a waypoint, or fix, (x_{ac_f}, y_{ac_f}) , and is using the autopilot to provide course guidance. The system dynamics are simply

$$\dot{\mathbf{x}}_{ac}(t) = \begin{bmatrix} V_{ac} \cos \theta(t) \\ V_{ac} \sin \theta(t) \end{bmatrix} \quad (9)$$

where aircraft heading, $\theta(t)$, is the control variable. Turn dynamics are ignored for simplicity.

The optimal control problem is a two-point BVP, with a minimum time performance index presented in Mayer formulation:

$$J_{ac} = t_f \quad (10)$$

Assigning $\lambda_1(t)$ and $\lambda_2(t)$ as Lagrange multipliers, the Hamiltonian is defined as follows:

$$\mathcal{H}(t) = \lambda_1(t)V_{ac} \cos \theta(t) + \lambda_2(t)V_{ac} \sin \theta(t) \quad (11)$$

The first-order necessary conditions provide the costate equations:

$$\begin{aligned} \dot{\lambda}_1(t) &= 0 \\ \dot{\lambda}_2(t) &= 0 \end{aligned} \quad (12)$$

allowing the minimization of the Hamiltonian by the unconstrained control:

$$\frac{d\mathcal{H}}{d\theta}(t) = 0 = -\lambda_1(t)V_{ac} \sin \theta(t) + \lambda_2(t)V_{ac} \cos \theta(t) \quad (13)$$

$$\Rightarrow \frac{\lambda_2(t)}{\lambda_1(t)} = \tan \theta(t) \quad (14)$$

The control is therefore constant, implying for this case that the state dynamics are constant, which allows a solution for the optimal control through simple integration of both states from the initial conditions (x_{ac_0}, y_{ac_0}) at $t_0 = 0$:

$$\begin{aligned} \mathbf{x}_{ac_f} &= \mathbf{x}_{ac_0} + \int_{t_0}^{t_f} \dot{\mathbf{x}}_{ac}(t) dt \\ &= \mathbf{x}_{ac_0} + t_f \begin{bmatrix} V_{ac} \cos \theta \\ V_{ac} \sin \theta \end{bmatrix} \end{aligned} \quad (15)$$

The unknown final time, t_f , is removed by solving both equations for t_f and equating them, leaving the optimal control:

$$\theta(t) = \tan^{-1} \left(\frac{y_{ac_f} - y_{ac_0}}{x_{ac_f} - x_{ac_0}} \right) \quad (16)$$

Note that for a recursive open-loop control, the initial values in Eq. (16) are simply the current position for each iteration, and the optimal control solved for by any method will simply be a function of the relative position ratio. Absent disturbances, the optimal path, and the actual path will unsurprisingly be direct to the target as shown in Figure 3.

4.1. Nonzero-mean or time-correlated disturbances

As they are unknown beforehand, the addition of the typical zero-mean, white, Gaussian, stochastic elements in the forms of model deficiencies or disturbances does not change the predicted solution for the optimal control. The effects of disturbances can be countered, somewhat, by re-solving for a new optimal path at various time steps, as was illustrated in Figure 1. However, the production of a new, instantaneous solution does not provide anticipation of future disturbance effects or any correction for past errors. For unmodeled effects that are more time correlated, or those that can be characterized by an unknown, nonzero mean, effective control requires some level of feedback, such as integration of the error between the expected and actual state paths for each step, or estimation of the unknown parameter(s) causing the disturbance.

To illustrate this, a constant bias, w , is added to the system in one axis. This bias represents some of the effects of a wind component parallel to that unit direction. Smaller stochastic effects of the wind are not modeled for this case study in order to more clearly show the predominant impact and to provide the opportunity for an analytical solution. The effects of a time-correlated noise source can be seen by simply replacing the experiment with a correlated function, $w(t)$. Even a time-correlated function that is zero-mean overall can be cut into segments of time where the mean is biased in one direction or the other, so the general effects of the noise contribution will be the same as demonstrated here, on smaller time scales.

The dynamics of Eq. (9) become

$$\dot{\mathbf{x}}_{ac}(t) = \begin{bmatrix} V_{ac} \cos \theta(t) \\ V_{ac} \sin \theta(t) + w \end{bmatrix} \quad (17)$$

and the Hamiltonian is updated to be

$$\mathcal{H}(t) = \lambda_1(t)V_{ac} \cos \theta(t) + \lambda_2(t)[V_{ac} \sin \theta(t) + w] \quad (18)$$

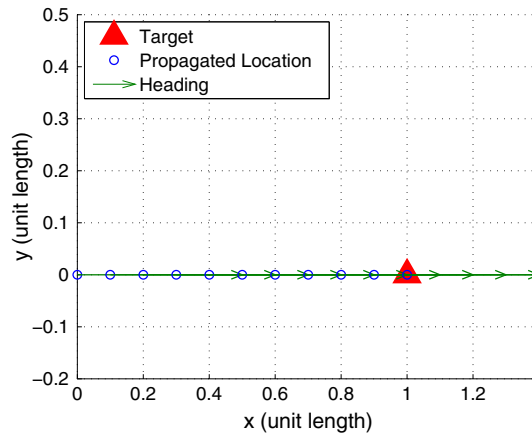


Figure 3. Recursive optimal control solution with no disturbances, $\Delta t = 0.1$ units.

The costate equations do not change, and the Lagrange multipliers are still found to be constant. Minimization of the Hamiltonian shows that the optimal control is constant as well, allowing integration of the states and removal of the unknown final time, leaving the relationship for the true optimal control, θ_t :

$$\frac{y_{ac_f} - y_{ac_0}}{x_{ac_f} - x_{ac_0}} = \frac{V_{ac} \sin \theta_t + w}{V_{ac} \cos \theta_t} \tag{19}$$

The true optimal path is shown in Figure 4, with an arbitrary constant wind bias of -4 (unit length)/(unit time).

If, however, the optimal steering is calculated without knowledge of the bias for each step of the digital controller, there obviously is error between the calculated optimal steering and the true optimal steering. With the appropriate trigonometric identities, the instantaneous steering error from any point may be found by re-solving the optimal problem from the new initial location and defining

$$\theta_e \equiv \theta - \theta_t = \sin^{-1} \left[w / \left(V_{ac} \sqrt{1 + \frac{y_{ac_f} - y_{ac_0}}{x_{ac_f} - x_{ac_0}}} \right) \right] \tag{20}$$

This steering error results in a ‘homing’ trajectory instead of a direct flight path, as shown in Figure 5.

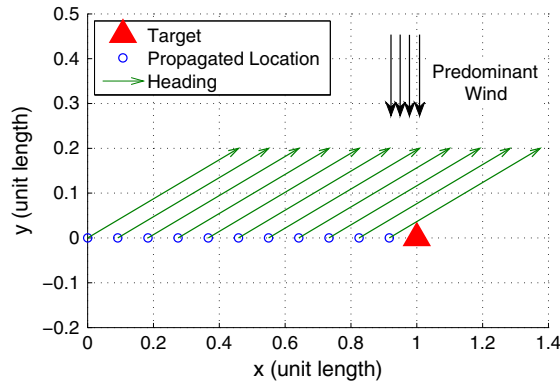


Figure 4. True optimal solution, with nonzero-mean disturbance, $\Delta t = 0.1$ units.

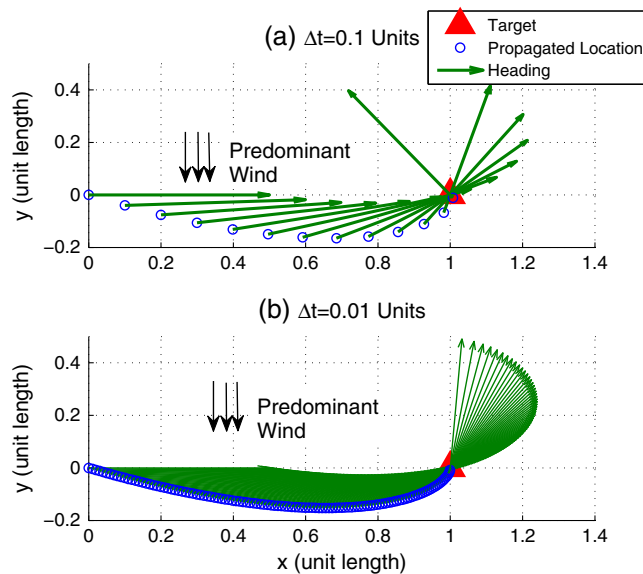


Figure 5. Recursive optimal solution with nonzero-mean disturbance (homing).

Equation (20) analytically proves the key point—that this steering error will always exist (excepting a displacement in the direction of a pure head or tail wind). Note that Eq. (20) is not dependent on sample time, or the speed of the recursive solution update, but only on the geometry of the problem at the time of the update and the intensity of the wind. A recursive open-loop solution will always produce a flawed steering solution, without the use of some sort of feedback to allow accounting for the wind bias. Attempts to increase the recursion rate may decrease the total path error but never overcome the bias. Figure 5(b) shows a recursion rate of 0.01 time units.

These results highlight the main lesson of this paper—the pitfall of assuming that a high recursion rate on an open-loop optimal solution is equivalent to optimal feedback control. In the face of nonzero-mean disturbance, the resultant path in Figure 5 is clearly short of what would be considered ‘optimal’. A simple feedback scheme demonstrates that the control solved for through rapid recursive open-loop planning requires additional input. Figures 6 and 7 show the effects of adding the PI control in the form

$$\theta_{fb}(t) = \theta(t) + k_p e_p(t) + k_i \int_{t_0}^t e_p(\xi) d\xi \tag{21}$$

where $e_p(t)$ represents the orthogonal component between the current position and the intended direct path. The gains were arbitrarily selected as $k_p = -20$ (unit length)/radian and $k_i = -60$ (unit length)/radian.

The addition of feedback to the recursive optimal solution causes the resultant path and control to more clearly follow the true optimal solution, regardless of the recursive update rate. In terms

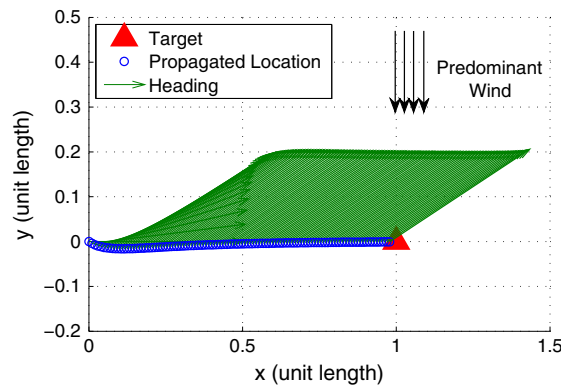


Figure 6. Optimal recursion with the addition of PI feedback, $\Delta t = 0.01$ units.

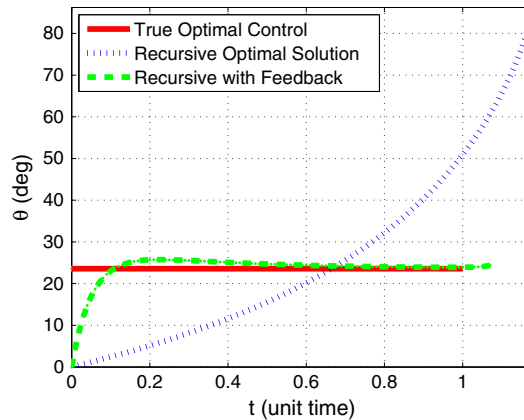


Figure 7. Control requirements with and without feedback.

of total time-to-target (the objective), the analytical solution for this particular example took 1.092 time units to complete the route, compared with 1.094 units for the recursive open-loop system with feedback, and 1.19 units with recursive open-loop updates only.

Beyond just the timing differences and the associated increase in fuel requirements, the arced path of the route found without an inner control loop has real-world navigation implications. On a regular basis under both instrument and visual flight rules, aircrafts are assigned to ‘proceed direct’ to a certain fix or are filed to proceed along a route corridor by means of navigation aids (e.g., TACAN (tactical air navigation system) and VOR (very high frequency omnidirectional radio range)), which provide a bearing angle to a fix. In either case, separation from other aircraft, clearance of terrain, and line-of-sight for reception of the navigation aid signals is only protected for a narrow corridor width. The clearance to proceed direct implies correcting against the winds to fly a direct ground path, not merely homing to the target as you would with a recursive open-loop controller, which would result in the large lateral excursions illustrated in Figure 5. Could the RTOC approach be changed to minimize error from a direct path? Certainly, but again, this implies implementing some sort of error feedback. The intent of this demonstration was to provide a counterexample to the concept that speeding up the open-loop recursion rate was equivalent to achieving optimal feedback control.

Therefore, in the design of control schemes to implement RTOC with a fast open-loop structure, consideration of the expected character of anticipated disturbances becomes critical. For systems that can anticipate time-correlated (at least relative to the system dynamics), or nonzero-mean, disturbances, some sort of integral control is required to achieve near-optimum performance.

4.2. Error integration through the addition of noise estimates into the system dynamics

For this simple case study, adding feedback was straightforward, and an inner PI error loop around the planned and actual state paths was included. For more complex, highly nonlinear systems, this technique may not be feasible. This is especially the case for systems with large deviations from the planned path as a result of a high ratio of disturbances to control authority, or systems with severe nonlinearities that would require, for example, an inordinate amount of gain scheduling. In such cases, one can avoid ‘manually’ choosing a feedback law by adding the additional information into the system dynamics prior to solving the optimal control problem for each step. For this application, this would involve first using the path error to form an estimate for the wind bias, $\hat{w}(t)$, and then updating the dynamics equation for each recursive solution to include the current estimate for the wind. For more complicated scenarios, the effects of the disturbances on the dynamics could be estimated through both proportional and integrated error elements. For this simple case study, however, the optimal control estimate, $\hat{\theta}$, can be solved analytically:

$$\hat{\theta}(t_k) = \tan^{-1} \left(\frac{y_{ac_f} - y_{ac}(t_k)}{x_{ac_f} - x_{ac}(t_k)} \right) - \sin^{-1} \left[\hat{w}(t_k) / \left(V_{ac} \sqrt{1 + \frac{y_{ac_f} - y_{ac}(t_k)}{x_{ac_f} - x_{ac}(t_k)}} \right) \right] \quad (22)$$

Again, this is an instantaneous solution at any time, t_k , used by substituting the current state into the original problem as new initial conditions. The effects are shown in Figure 8. Note that no attempt is made to return to any previous reference solution, but instead the system follows the optimal path that was calculated from each current position. Because the only disturbance that was added was constant and linear and no measurement noise was considered, the estimate is correct after only one time step. Beyond that, the calculated solution matches the true optimal solution from that point, because all of the information about the disturbance is completely known.

This final control structure represents the best of all worlds, combining the positive aspects of classical control with the emerging benefits of RTOC. The control to be applied is completely generated by the numerical optimization scheme, but implicitly contains the integrated error feedback, which is used to update the system dynamics and change the optimal control problem for each iteration, overcoming the inability of the purely recursive open-loop structure to handle time-correlated or nonzero-mean unmodeled effects.

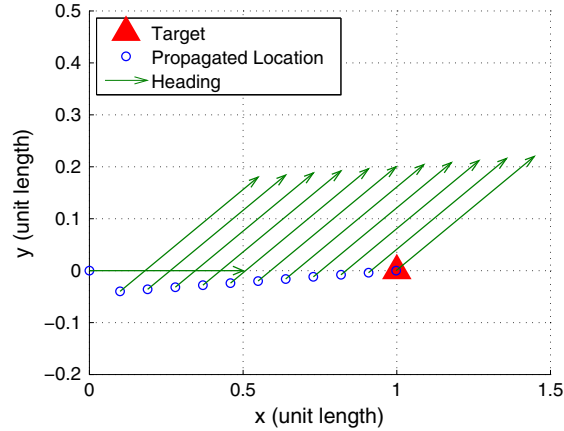


Figure 8. Recursive optimal control using feedback to update dynamics.

5. CASE STUDY B: REAL-TIME AIRCRAFT ATTACK PLANNING

A more robust and realistic example quickly shows the potential impacts of a failure to consider error integration in recursive RTOC. One of the most obvious applications for optimal path planning is for threat avoidance. Stealth considerations of radar cross section, threat radar detection capability, and effective surface-to-air missile (SAM) engagement ranges must be considered in attack planning. For maximum effectiveness, the plan should be accomplished in real time. Pop-up threats, by definition unanticipated, cannot be avoided using mission planning that was accomplished prior to takeoff. In addition, without the ability to change the plan enroute, a pilot cannot immediately exploit weaknesses such as a defense system that has been removed or reduced in operational capability in some manner by another strike package. All of this is possible with RTOC.

Consider a strike planned on a soft target, defended with a perimeter of SAM threats along the planned route. For specific applications, the performance index would be designed to consider the specific capabilities of each threat and the advantages of the attacking aircraft, but as a generic illustration, a 15 nm (nautical mile) or 30 nm ring is assigned to each SAM location, and the run is constrained to a constant altitude with a constant 350 knots true air speed. The SAM ring represents a weapon employment zone, outside of which the aircraft can safely operate in the absence of aware air-to-air assets.

Admittedly avoiding minimum exposure and stealth issues (which could be incorporated with the appropriate modeling), the basis for the performance index for this attack scenario remains a Mayer cost function of final time, as it was for the case study in Eq. (10). The other likely option would be a minimum fuel consumption index, which is equivalent for a constant altitude and an airspeed (if throttle increases during turns are considered negligible). The no-wind dynamics remain unchanged from Eq. (9), and the control is still a commanded heading, which would be the input to a standard heading-hold autopilot with a feedback-based bank angle control law to drive the physical actuators. RTOC provides the flexibility of avoiding additional pop-up threats simply by adding new path constraints, ensuring flight outside of the SAM threat rings:

$$[x(t) - x_i]^2 + [y(t) - y_i]^2 \geq \rho_i^2 \quad i = 1 \dots n_{SAM} \quad (23)$$

where n_{SAM} is the number of currently known SAMs.

Control is accomplished by recursively solving the optimal control problem, with no explicit feedback (as before, some implicit feedback is available through the re-initializing of the optimal control problem at the current measured position). This mirrors the structure of RTOC becoming popular in the literature, eliminating the inner feedback loop around the optimal path.

The optimal control solution is found using a direct technique of the class of pseudospectral methods, defined well in [1, 6], with an excellent framework for three methods found in [23]. In

particular, the results of this paper were found with the Gaussian pseudospectral method, briefly described in the next section and more completely in [14] and [15].

5.1. Gaussian pseudospectral method

For the Gaussian pseudospectral method, the time period of the problem is mapped from $t \in [t_0, t_f]$ to $\tau \in [-1, 1]$ via the affine transformation:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \quad (24)$$

Collocation is performed on this interval at N Legendre–Gauss (LG) points, which are the roots of the Legendre polynomial:

$$P_N(\tau) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} [(\tau^2 - 1)^N] \quad (25)$$

Collocating at the LG points allows accurate approximation of the initial guess with a polynomial. Lagrange interpolating polynomials, $\mathbf{U}(\tau)$ and $\mathbf{X}(\tau)$, are found for the control, $\mathbf{u}(\tau)$, at the collocation points, and for the state, $\mathbf{x}(\tau)$, using both the collocation points and the initial point, $\tau = -1$. Other pseudospectral techniques, such as Legendre–Gauss–Radau and Legendre–Gauss–Lobatto, are defined using different combinations of collocation and interpolating points.

Linearity of the summation operator and the isolation property of the Lagrange interpolating polynomials allows the elements of a derivative matrix, \mathbf{D} , to be precalculated through the Legendre polynomials with

$$\mathbf{D}_{ki} = \begin{cases} \frac{(1+\tau_k) \frac{dP_N}{d\tau}(\tau_k) + P_N(\tau_k)}{(\tau_k - \tau_i) \left[(1+\tau_i) \frac{dP_N}{d\tau}(\tau_i) + P_N(\tau_i) \right]} & i \neq k \\ \frac{(1+\tau_i) \frac{d^2P_N}{d\tau^2}(\tau_i) + 2 \frac{dP_N}{d\tau}(\tau_i)}{2 \left[(1+\tau_i) \frac{dP_N}{d\tau}(\tau_i) + P_N(\tau_i) \right]} & i = k \end{cases} \quad (26)$$

where

$$\frac{dP_N}{d\tau} = \frac{-(n+1)P_N(\tau)}{1-\tau^2} \quad (27)$$

such that when the dynamic constraints of Eq. (17) are defined as $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$, they can be transcribed at each collocation point to the static constraint:

$$\sum_{i=0}^N \mathbf{D}_{ki} \mathbf{X}(\tau_i) - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0} \quad (k = 1, \dots, N) \quad (28)$$

Finally, an additional variable is defined at the ending boundary point, $\mathbf{X}(\tau = 1)$, and an additional constraint is defined by integrating the state derivative path from the initial condition via Gaussian quadrature:

$$\begin{aligned} \mathbf{0} &= \mathbf{X}(\tau = 1) - \mathbf{X}(\tau = -1) - \frac{t_f - t_0}{2} \sum_{k=1}^N w_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \\ &= \mathbf{X}(\tau = 1) - \mathbf{X}(\tau = -1) - \sum_{i=0}^N \sum_{k=1}^N w_k \mathbf{D}_{ki} \mathbf{X}(\tau_i) \end{aligned} \quad (29)$$

where the LG weights, $w(\tau_k)$, are used that correspond to the LG collocation points (see [14] and [15] for more detail on the method).

Although somewhat complicated, the optimization method is remarkably efficient. The collocation points, their corresponding Gauss weights, and the \mathbf{D} matrix are only a function of the number of nodes desired for the solution, N , and can, therefore, all be calculated off-line *a priori*. Even the actual work of finding the interpolating polynomial to approximate the guess is not required,

because the algorithm only needs evaluations of the polynomial at each collocation node, and these will always have the exact value of the guess with Lagrange interpolation.

The transcription is then complete, and an NLP is then formed with the original cost function (Eq. (10)), the original boundary conditions, the path constraints of Eq. (23) transformed to the τ -domain, the final constraint of Eq. (29), and the now static constraints of Eq. (28) to enforce the original dynamics. Evaluation of the constraints is extremely fast, with the derivative of every state at every node being known with only one multiplication of the \mathbf{D} matrix and the state vector. More complicated cost indices with Lagrange or Bolza forms can also be quickly evaluated using the quadrature and the known Gauss weights.

The NLP can be solved with a host of software programs designed for that purpose. For this paper, the shareware program GPOPS [14] was used as the shell that performs the transcription and hands the resultant NLP to the program SNOPT [12], which performs the optimization. The optimization was performed with a standard desktop 2.49 GHz processor with Microsoft Windows® XP running a MATLAB® (MathWorks, Natick, MA, USA) environment. Actual computation time took an average of 0.18 s for each solution (using 30 nodes, which is likely overconservative for this problem). Increases in speed could be expected if the software were tailored for this specific application and the algorithm translated into a faster programming language such as C++™. However, simulation was artificially accomplished with zero computation time for the recursive optimal solution (i.e., the same as you could accomplish with an analytic process, in simulation). Although this is unrealistic, it puts the recursive solution in the best possible light, showing the limitations of what could be accomplished even as the optimal control problem approaches being solved in real time. In this manner, the optimal solution is available exactly at any desired recursive solver rate, which can be increased to the point where the bandwidth restrictions are due to the sample time of the controller and the speed of the physical actuators. Any limitations remaining, therefore, are deficiencies in the technique, and not complications from computational delay between the request and the receipt of the optimal solution.

6. RESULTS

Figure 9 shows the initial optimal path, planned by the subject aircraft as it starts the attack run, avoiding the known SAM rings and proceeding to the target. In a deterministic system, with the absence of disturbances such as wind or any further information, this route would be flown perfectly, and according to Bellman’s principle of optimality, even as the optimal problem is recalculated along the route of flight, the resultant course will never change [3].

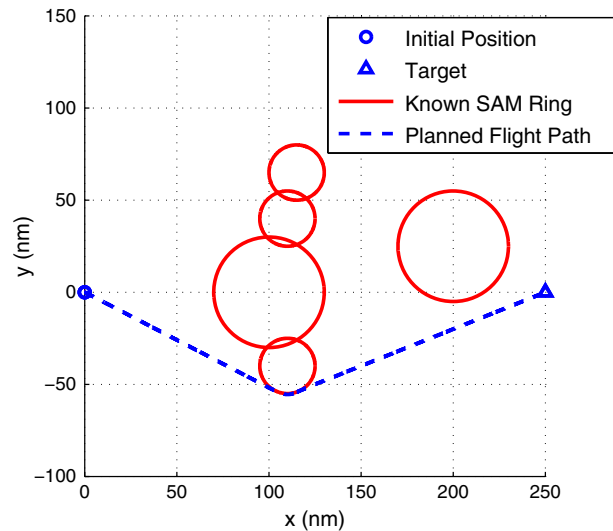


Figure 9. Recursive optimal planner initial flight path around surface-to-air missile (SAM) threats to target.

Introduction of new information is incorporated and adjusted by adding the appropriate constraints. Figure 10 shows the position of the aircraft along the optimal path as the aircraft's systems become aware of a new emitter. Direct methods are relatively insensitive to initial guesses, and no additional effort is required for minor changes made on-the-fly to avoid new pop-up threats other than adding a new path constraint before the next recursion of the optimal solver.

6.1. Search methods to ensure a global minimum

Optimization via gradient methods of any type guarantees only local minimums. When convex threat constraints such as these SAM rings are enforced, the remaining feasible solution space becomes a nonconvex set of convex channels. For example, with a single constraint, two convex channels are created. A guess on either side of the center of the constraint will produce its own unique local minimum solution. Determination of which solution is the global minimum can only be found by exhaustively investigating the performance of an initial guess in each of the convex channels.

Although this technique may be feasible of scenarios such as this one, where the planner is determining paths in fractions of a second that will cover hundreds of miles, it is by no means elegant or efficient. For more time-critical scenarios, or for scenarios that are less well-understood, more intelligent techniques must be employed to narrow the search space. These techniques will be scenario dependent and are most effectively accomplished when mandatory attributes of the optimal path can be identified and exploited. For instance, in this case study, with a constant speed and a minimum time objective, once a clear, direct path from any point can be found to the target, there is no need to search for other paths from that point (or, by extension, from any channels that contain that point).

Several guess-generating algorithms can be designed to determine the possible channels for investigation, such as the branch-and-bound technique found in [24]. This algorithm defines routes by initially attempting a direct path and then marking two new paths on either side of the first violated constraint (or set of constraints if continuously connected). Each of those branches then attempts to complete the course with a direct path, and the process continues. It is a simple matter to determine the minimum possible time for each channel (always either riding the edge of constraint or moving directly from and to tangent points). The set of potential paths is sorted by minimum possible time and provided to the optimization algorithm, which checks channel for a feasible optimum path (including maximum turn rates and other dynamics). For the methods used in this paper, providing a single waypoint on the desired side of a constraint to be avoided is sufficient for a guess in each channel.

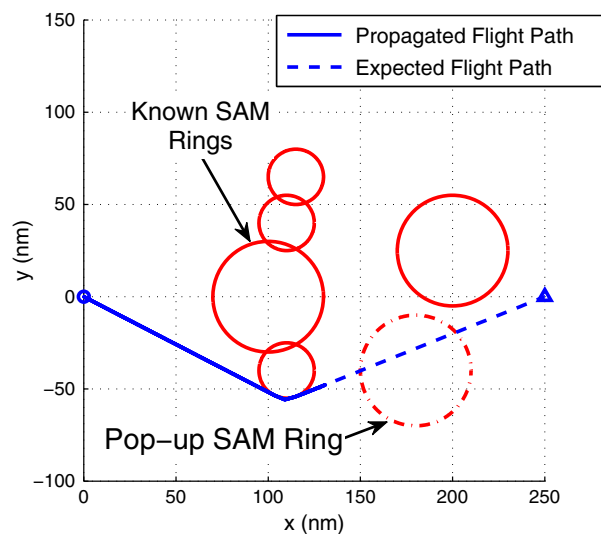


Figure 10. Recursive optimal flight path at time = 25 min, as a pop-up threat emerges—no wind. SAM, surface-to-air missile.

For a purely minimum time scenario, once the first feasible path is found, the remaining set of routes to be investigated can be narrowed—and often completely discarded (there is no need to investigate routes with larger minimum possible times). For more complicated scenarios, such as stealth aircraft that not only have a minimum allowable distance from each SAM site, but also a running Lagrange penalty for *proximity* to any site, more complicated discriminators must be used to ensure a global minimum without an exhaustive search.

Figure 11 shows the result of two new emitters being sensed by the aircraft. The guess-generation algorithm provides two routes to investigate, and after running the optimization routine on the shorter, the longer route is discarded because the minimum possible time is greater than the time of the feasible solution, resulting in the completed flight path of Figure 12.

These results support the efficacy of using an optimal control solution in defining flight paths, which include changing parameters or constraints. As stochastic inputs in the form of disturbances and measurement noises are applied to the system, the controller will still achieve the primary goal, as was demonstrated in Case study A. However, the path taken may be far less than the best that could be accomplished in the same circumstances, and may still result in mission failure. For the

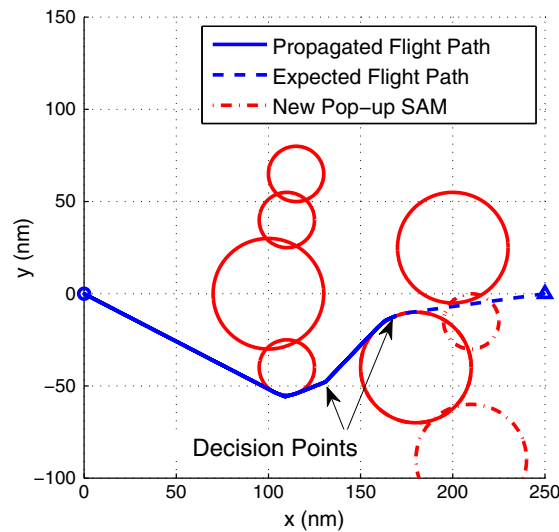


Figure 11. Recursive optimal flight path at time = 30 min, with new pop-up threats. Major adjustment required—no wind. SAM, surface-to-air missile.

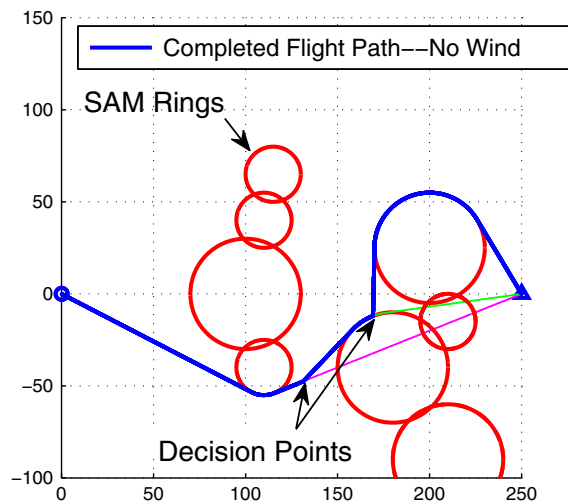


Figure 12. Completed flight path using recursive optimal planning—no wind. SAM, surface-to-air missile.

design of an RTOC system, additional control may likely be desired to overcome the lack of path-error integration in the recursive-only structure, especially in the presence of possible nonzero-mean or time-correlated disturbances or measurement errors.

6.2. Stochastic inputs

A first-order Gauss–Markov process is used to simulate a realistic profile for the intensity of potential wind gusts:

$$\dot{w}_{gust}(t) = -\frac{1}{T}w_{gust}(t) + \eta(t) \quad (30)$$

where T is a time constant for the system, and η is zero-mean, white, Gaussian noise with

$$\begin{aligned} E[\eta(t)] &= 0 \\ E[\eta(t)\eta(t + \tau)] &= Q_{gust}\delta(\tau) \end{aligned} \quad (31)$$

using the standard definition for the delta function. Similar Gauss–Markov processes were used to determine a lower frequency variation in wind intensity, w_{pred_wind} , and for determining variance in the wind direction. Measuring wind velocity in knots, and direction in degrees, the time constants for the two wind components were 200 and 40 h, with respective input strengths of 0.25 and 0.2 knots², and wind direction was determined with a time constant of 50 h and unit intensity noise. The resulting wind intensity and direction were added to a predominant wind and predominant direction, respectively, resulting in the disturbance input shown in Figure 13. This is representative of a weather forecast for winds 220° variable 230° at 30 gusting 35 knots (or an average summer day at altitude).

As the final time was unknown, more wind was generated than was actually used. The wind input as a disturbance causes the same steering difficulty shown in Figure 5 for case study A. No matter how fast a recursive optimal solution is calculated, without feedback to either compensate externally (with an inner control loop), or to estimate wind component intensity (for the outer planning loop), there is a steering error between the optimal control that is unaware of the disturbance and the true optimal control. Once near the SAM rings, the errors in steering become more critical, and a constraint is violated, as shown in the inset of Figure 14. The magnitude of the constraint violation is a function of the size of the disturbance, the recursive solution update timing, and any applied turn rate limit. The more time between updates, the longer the steering error is applied and the greater the encroachment. If there were no limits on the turn rate of the aircraft, the recursive system could always meet the constraint as the update interval approaches zero, because in the limit there will always be room for ‘one more’ heading update that will be closer to the tangent of the constraint arc due when the optimal control is re-solved for (this assumes the vehicle is riding the ‘outside’

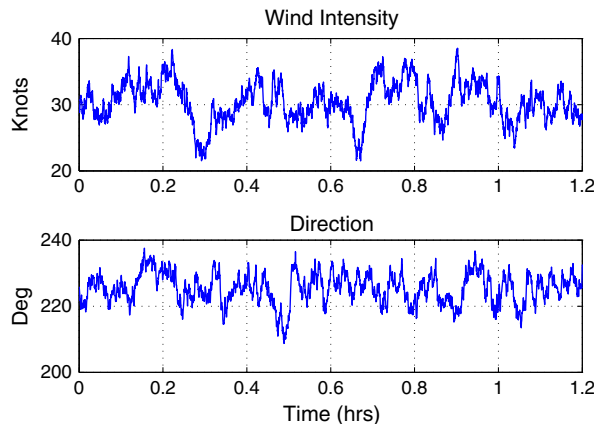


Figure 13. Wind input to the system.

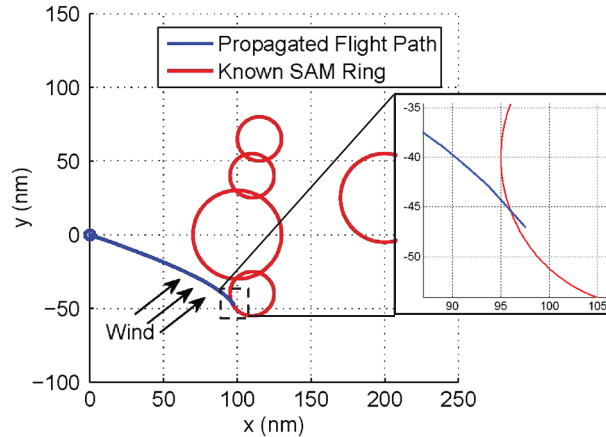


Figure 14. Steering failure with recursive real-time optimal control structure in the presence of wind. SAM, surface-to-air missile.

of a constraint that is curved away and does not necessarily hold for attempts to ride the inside of a curve).

To be fair in the example of flying through the SAM rings, there is no ‘magic line’ in the sky where ‘1 inch to the left’ equals safe and ‘1 inch to the right’ equals dead, but there certainly are systems with hard limits (physical terrain, structures, etc.), and optimal solutions often ride as close as allowable to those limits. If the ability of the system to change course is limited (i.e., a slow maximum turn rate), then late steering corrections approaching a constraint can cause large violations.

Even disregarding the SAM ring violations for this example, the path itself is clearly not optimal. Recall from the simple case study that there will *always* be steering error in the case of a time-correlated or nonzero-mean disturbance. This can be seen in the bending of the optimal path of Figure 14, just as was the case for the homing solution of Figure 5. For case study A, it was shown in Eq. (20) that the steering error was not a function of the update timing, only a function of the geometry and the magnitude of the disturbance. This is why faster updates did not remove the problem, as illustrated in Figure 5(b). Increasing the update rate does decrease the amount of time that you follow the erroneous heading, but there will only be small changes in the erroneous control command for the next step until there is significant deviation from the optimal path.

Classical techniques such as integration of path error give us the ability to bias the commanded steering or alter the optimal control problem in order to best anticipate the wind effects and better estimate the true optimal control. Either of these methods represent an improvement in the structure of the RTOC system.

6.3. The remedy—feedback for estimation and control

As was shown in case study A, classical techniques, when usable, offer a simple remedy. One method of correcting for stochastic inputs to the system is to apply a proportional, closed-loop bias to the control solution generated from the error signal between the expected and actual motion for each time step. Integrating this error signal allows compensation for steady-state errors that exist because of nonzero-mean disturbances. Alternatively, if the disturbance can be quantified, feedback can be used to form an estimate of the disturbance, which can be applied to the dynamics used in the optimal control solution. This feedback technique leaves solving for the optimal control comfortably in the hands of the optimal path planner, and is therefore the method of choice.

6.4. Wind estimation

For implementation, estimates of the wind direction and velocity are required, which can be replaced with estimates of the magnitudes of the wind components in the x and y directions. In the absence

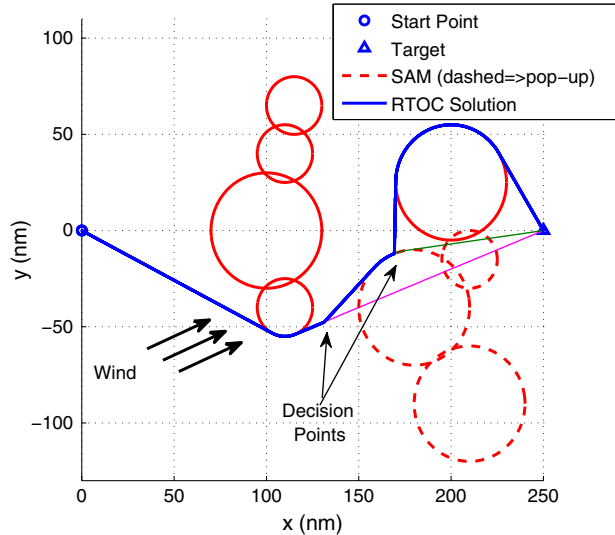


Figure 15. Complete flight path, wind compensated through estimation from position feedback. SAM, surface-to-air missile; RTOC, real-time optimal control.

of an available wind sensor, the wind component measurement can be produced from the difference between the expected and actual position in each axis divided by the time step (or an averaged position over several time steps depending on the accuracy of the position measurement). For simplicity, a wind measurement is assumed available for every time step from some source.

With available wind measurements, a number of estimation techniques can be used for wind estimation, the most optimal choice being a Kalman filter [25, 26], but any simple corrective estimation algorithm is sufficient. Some smoothing or estimation is recommended over raw data, as the optimal path planner will assume constant wind for the remainder of the attack route, and the large variance in gusts will cause the control solution to oscillate from step to step. For this paper, the initial route is planned without wind. The second time step assumes a wind measured during the first step, and from there a smoothing algorithm is used for estimation, recursively updating the wind estimate to include a portion of the residual error between the latest estimate and the last measurement:

$$\hat{w}_x(t_{k+1}) = \hat{w}_x(t_k) + k_{wind} [w_{x_{meas}}(t_k) - \hat{w}_x(t_k)] \quad (32)$$

An identical formulation is used for the y -axis component. For simplicity, one tenth of the residual error is applied at each time step, but the Kalman filter equations could easily be implemented for a more optimal choice of the proportional parameter k_{wind} .

With an available wind estimate generated from the closed-loop feedback of the vehicle state, the assumed system dynamics are updated by adding the appropriate components into each channel, and the recursion is allowed to proceed. Using decision points similar to those from Figure 12 where the aircraft is made aware of pop-up SAM threats, the completed flight path can be seen in Figure 15, with a path almost indistinguishable from the no-wind optimal path of Figure 12. The aircraft in this simulation avoided the obstacles while correcting for the random disturbances, illustrating the ability of RTOC with disturbance estimation in lieu of inner-loop feedback to achieve mission objectives.

7. CONCLUSION

This paper addressed the suggestion, prevalent in recent literature, that use of an optimal path planner with a recurrent open-loop control structure was equivalent to optimal feedback control if the recurrence was accomplished at a high enough rate. The current trend in the design of RTOC algorithms has been to eliminate the use of an explicit inner feedback loop entirely. Although implicit feedback is available, it is by no means optimal, and suffers greatly in the presence of stochastic inputs, particularly those which are nonzero-mean or time-correlated.

Two case studies were presented to demonstrate this lesson. The first study analytically proved that the instantaneous control signal error was not a function of recursion rate, and both case studies graphically demonstrated the potential implications of designing a controller for RTOC with the structure currently found in the literature. The second case study also presented a full, realistic application of RTOC for a course planning algorithm of an attack aircraft through a dynamic set of simulated SAM threats that changed during the flight. The Gaussian pseudospectral method was used for optimization in a direct manner through transcription of the optimal control problem to an NLP, and solved in a recursive fashion as the aircraft proceeded along the route. Stochastic, biased wind inputs were added as disturbances. In both case studies, the application of RTOC with a fast open-loop recursive structure was found to be not only nonoptimal but also infeasible for the specific applications once stochastic disturbances were added.

Two methods were proposed as options for the RTOC design. The first included the use of the traditional inner feedback loop with classical control theory applied to generate a control bias that is added to the instantaneous control from the path planner. The second method, applied in both scenarios, was to use explicit feedback for disturbance estimation in order to counter the effects of disturbances through the optimal path planning process. This allows feedback to be used in cases where there is no ability to generate a suitable closed-loop feedback law.

Individual control problems will require a designer's eye to find the best particular control structure. The capacity to respond to changing threat conditions and other dynamic requirements and constraints makes the use of RTOC very appealing. For realistic disturbance rejection, however, pure recursive open-loop control will likely be insufficient, regardless of the ability to increase the recursion rate. In some manner, future designs for RTOC system structures must incorporate explicit feedback to integrate state path error in order to counter the effects of disturbances, particularly for nonzero-mean or time-correlated stochastic inputs.

REFERENCES

1. Benson D, Huntington GT, Thorvaldsen TP, Rao AV. Direct trajectory optimization and costate estimation via an orthogonal collocation method. *Journal of Guidance, Control, and Dynamics* 2006; **29**(6):1435–1440.
2. Bryson AEJ, Ho YC. *Applied Optimal Control*. Hemisphere Publishing Corporation: Washington D.C., 1975.
3. Kirk DE. *Optimal Control Theory: An Introduction*. Dover: Mineola, NY, 2004.
4. Rao AV, Tang S, Hallman WP. Numerical optimization study of multiple-pass aeroassisted orbital transfer. *Optimal Control Applications and Methods* 2002; **23**:215–238.
5. Betts JT. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics* 1998; **21**(2):193–207.
6. Benson D. A Gauss pseudospectral transcription for optimal control. *Ph.D. Dissertation*, Massachusetts Institute of Technology, 2005.
7. Euler L. *Institutiones Calculi Integralis*. Academy of Sciences: St. Petersburg, Russia, 1768.
8. Betts JT. *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Design and Control. SIAM: Philadelphia, PA, 2001.
9. von Stryk O. Numerical solution of optimal control problems by direct collocation. *Optimal Control Theory and Numerical Methods* 1993; **111**:129–143.
10. Hargraves CR, Paris SW. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics* 1987; **10**(4):338–342.
11. Paris SW. Enhanced procedures for direct trajectory optimization using nonlinear programming and implicit integration. *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Vol. 2. Keystone, CO, USA, 21–24 August 2006; AIAA-2006-6309.
12. Gill PE, Murray W, Saunders MA. SNOPT: an SPQ algorithm for large-scale constrained optimization. *SIAM Journal on Optimization* 2002; **12**(4):979–1006.
13. Ross IM. A beginner's guide to DIDO, version 7.3. *Technical Report TR-711*, Elissar, LLC: Monterey CA 93942, 2007.
14. Rao AV, Benson D, Darby CL, Francolin C, Patterson M, Sanders I, Huntington GT. User's manual for GPOPS version 2.3: a MATLAB® software for solving multiple-phase optimal control problems using the Gauss pseudospectral method. *Technical Report*, TOMLAB™: Gainesville, FL 32607, 2009.
15. Rao AV, Benson DA, Darby CL, Patterson MA, Francolin C, Huntington GT. Algorithm 902: GPOPS, a MATLAB® software for solving multiple-phase optimal control problems using the Gauss pseudospectral method. *ACM Transactions on Mathematical Software* 2010; **37**(2):1–39.
16. Yan H, Fahroo F, Ross I. Real-time computation of neighboring optimal control laws. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, 5–8 August 2002; 902–908.

17. Milam MB, Mushambi K, Murray RM. A new computational approach to real-time trajectory generation for constrained mechanical systems. *39th IEEE Conference on Decision and Control*, Sydney, NSW, Australia, 12–15 December 2000; 845–851.
18. Strizzi J, Ross IM, Fahroo F. Towards real-time computation of optimal controls for nonlinear systems. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, 5–8 August 2002.
19. Ross IM, Sekhavat P, Fleming A, Gong Q, Kang W. Pseudospectral feedback control: foundations, examples and experimental results. *Proceedings of AIAA – Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, USA, 21–24 August 2006; AIAA-2006-6354.
20. Hurmi MA, Sekhavat P, Ross IM. Autonomous trajectory planning using real-time information updates. *26th AIAA Applied Aerodynamics Conference*, Honolulu, HI, 18–21 August 2008; AIAA-2008-6305.
21. Sekhavat P, Fleming A, Ross IM. Time-optimal nonlinear feedback control for the NPSAT1 spacecraft. *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, CA, 24–28 July 2005; 843–850.
22. Bollino K, Ross IM. A pseudospectral feedback method for real-time optimal guidance of reentry vehicles. *Proceedings of the 2007 American Control Conference*, New York City, NY, 11–13 July 2007; 3861–3867.
23. Garg D, Patterson MA, Hager WW, Rao AV, Benson DA, Huntington GT. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* 2010; **46**(11):1843–1851.
24. Eele A, Richards A. Comparison of branching strategies for path-planning with avoidance using nonlinear branch-and-bound. *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 18–21 August 2008. DOI: 10.2514/6.2008-6463, AIAA-2008-6463.
25. Maybeck PS. *Stochastic Models, Estimation, and Control*, Vol. 1. Navtech Book and Software Store: Arlington, VA, 1994.
26. Brown RG, Hwang PYC. *Introduction to Random Signals and Applied Kalman Filtering* (3rd ed.) John Wiley & Sons: New York, NY, 1997.