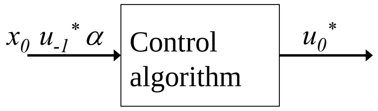
Storage and Retrieval of Optimal Control

J. D. Hedengren and T. F. Edgar
The University of Texas at Austin

Introduction

Another area of applicability for the ISAT algorithm lies in storage and retrieval of the optimal control. Real-time control presents many unique challenges. When large, non-linear first principles models are used to predict optimal control, the controller computational cycle time can be undesirably slow. Storage and retrieval of the optimal control is desirable since the same control problem is repeatedly solved with different initial conditions. In this case the initial conditions are the values of the current states of the model (x_0), the previous optimal control values (u_{-1}^*), and any adjustable parameters in the model or objective function (α). If all of the state values are not exactly measured, state estimation must be used to estimate the remaining states. For deterministic control algorithms, the same set of initial conditions will always produce the same set of outcomes. In this case the outcomes are the optimal control values (u_0^*). In block diagram form, the predictive variables or current states enter the system and leave as a set of outcomes or optimal control values. The block represents the control algorithm that determines the optimal control values. There is no limitation on the type of control algorithm that is used to calculate the optimal control values, only that the same set of initial conditions produces the same set of optimal control values.



The block diagram is equivalent to the following equation form.

$$u_0^* = f(x_0, u_{-1}^*, \alpha)$$

Sensitivity information may also be available from the control algorithm. The sensitivity matrix reveals the amount that the optimal control values change with a small perturbation in the initial conditions. In storage and retrieval of the optimal control it is desirable for the estimated values of u_0^* to be within some error tolerance (ε_{tol}) of the calculated u_0^*.

$$\left| u_0^* - u_0^*{}_{est} \right| \le \varepsilon_{tol}$$

Costs associated with the storage and retrieval of optimal inputs include configuration costs, CPU time costs, and storage costs. Overall, storage and retrieval of the optimal inputs may be desirable if the following conditions exist.

- Retrieval time is much faster than the control algorithm
- The control algorithm is deterministic
- Real time constraints make the control algorithm infeasible
- The CPU time to generate the database of optimal controls is small compared with retrieval savings
- Storage costs are small

Control algorithms based on large, non-linear first principles models meet these qualifications. In situ adaptive tabulation (ISAT) provides the fast retrieval time, compact storage of optimal controls, and database generation in regions of the state space accessed by the controller.

The ISAT Record

The basic unit of the ISAT database is the record. For optimal control storage and retrieval an ISAT record consists of the initial conditions, the optimal control values, a sensitivity matrix, and an ellipsoid of accuracy (EOA). The initial conditions and optimal control values are from a previous optimal control evaluation. If the control algorithm includes a sensitivity calculation, then the sensitivity matrix was created at the same instance. If it does not include a sensitivity calculation then the sensitivity is statistically estimated. Details about the statistical approximation of the sensitivity are given in a subsequent section. Finally, the EOA is a matrix used to approximate the amount of error in the estimated optimal controls. A distinguishing feature of ISAT over other storage and retrieval methods is that the EOA maintains automatic error control. Details about the initialization and modification of the EOA are also given in a subsequent section. The elements *accessed* and *stem* are additional elements for database administration. *Accessed* records the number of times a record is used for retrieval. *Stem* is a FORTRAN pointer to the binary tree.

Elements of an ISAT Record	Alias	Data Type
initial conditions	*start*	$\Re^m$
optimal control values	*finish*	$\Re^n$
sensitivity	*sensitivity*	$\Re^{nxm}$
ellipsoid of accuracy	*eoa*	$\Re^{mxm}$
accessed	*accessed*	integer
stem	*stem*	pointer

Searching the Records with Binary Trees

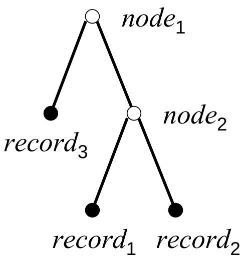
When accessing the database, the only piece of information that is known is a query vector of initial conditions. Ideally, the closest record would be obtained by minimizing a measure of closeness. In this case the measure of closeness is the 1-norm given by the difference between the query vector and the stored *start* vector.

$$\phi = |start_{query} - start_{stored}|$$

Searching the ISAT records sequentially would require $O(N)$ operations to completely search the database and find the closest record. A more efficient search structure is the binary tree. A balanced binary tree requires $O(log_2(N))$ operations for locating a record.

Elements of a Binary Tree Node	Data Type
v	$\Re^m$
a	$\Re$
lt	pointer
gt	pointer
leaf (when pointing to a record)	pointer

Each node of a binary tree can either be a leaf or branch. The leaves of the binary tree are individual records of the ISAT database. A branch, on the other hand, points to two other nodes. All branches divide until a leaf terminates the line.



In the above figure, *$node_1$* and *$node_2$* are the elements of the binary tree. Each element consists of a cutting plane, defined by *v* and *a*, that describe a division between two records. The *$node_2$* cutting plane is defined by the following two equations.

$$v = start_2 - start_1$$

$$a = v^T \left(\frac{start_2 + start_1}{2} \right)$$

The pointers *lt* and *gt* direct the search to the appropriate successive node. When $v(start_{query}) \leq a$ the pointer *lt* (less than) is chosen. Likewise, when $v(start_{query}) > a$ the pointer *gt* (greater than) is selected. In this case both of the pointers are to leaves, and the search terminates by selecting one of the records.

One of the drawbacks to binary tree searching is that the closest record is not always selected. To overcome this deficiency, multiple binary trees are used to increase the probability of finding the closest record. The records are equally divided among the binary trees to maintain a balance in search times. Once all of the binary trees are searched, a sequential search is performed to determine the closest record among the ones the binary trees selected. By adjusting the number of binary trees, an effective compromise is reached between the accuracy of the sequential search and the speed of the binary tree search.

Once a close record is located, ISAT performs one of three scenarios. These scenarios include retrieval, growth, and addition. Each of these is described in more detail below.

ISAT Retrieval

The automatic error control decides if retrieval is appropriate. The error control is accomplished with the ellipsoid of accuracy (EOA) with a center being the stored *start*. Another point, $start_{query}$ is within the EOA if $(start_{query} - start_{stored})^T\ eoa\ (start_{query} - start_{stored}) \leq \varepsilon_{tol}^2$. If the query point is within the EOA then *finish* is estimated with a linear approximation.

$$finish_{est} = finish_{stored} + \frac{\partial\, finish_{stored}}{\partial start_{stored}} \left(start_{query} - start_{stored} \right)$$

If $(start_{query} - start_{stored})^T\ eoa\ (start_{query} - start_{stored}) > \varepsilon_{tol}^2$ then the point $\mathrm{start}_{\mathrm{query}}$ is outside of the EOA and a retrieval cannot be performed. Even though the query point is not inside the EOA, the linear approximation may still be within the error tolerance for *finish*. The next step of the algorithm is to check the actual error.

ISAT Growth

In order to check the actual error, an original function calculation must be performed ($finish = f(start)$). If $|finish - finish_{est}| > \varepsilon_{tol}$ the EOA should not be expanded. Instead a new record should be added to the ISAT database. The growth step should be skipped and the algorithm jumps ahead to the ISAT addition phase.

If $|finish - finish_{est}| \leq \varepsilon_{tol}$ the EOA can be expanded to include $start_{query}$. This new region is a minimum volume ellipsoid that includes the new point, $start_{query}$, and the original EOA. The growth algorithm involves six steps. Each of the steps is described first in mathematical terms and subsequently with a two dimensional graphical example.

Step 1: Definition of the EOA and growth point

The EOA is defined by the following equation.

$$(start - start_{stored})^T\ eoa\ (start - start_{stored}) = \varepsilon_{tol}^2$$

To make the mathematics more compact and readable, the center point of the coordinates is aligned with the center of the ellipse.

$x = start - start_{stored}$

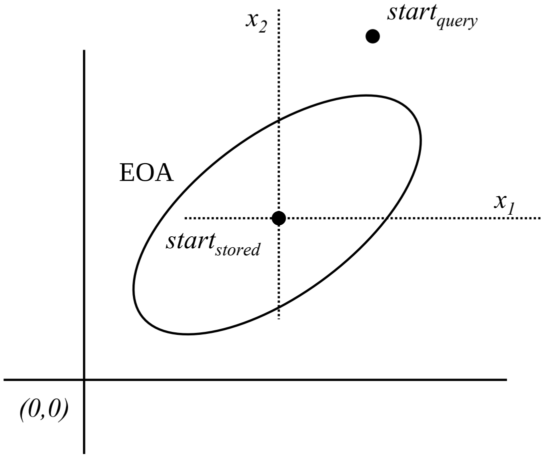
$x_{query} = start_{query} - start_{stored}$

$M_x = eoa$

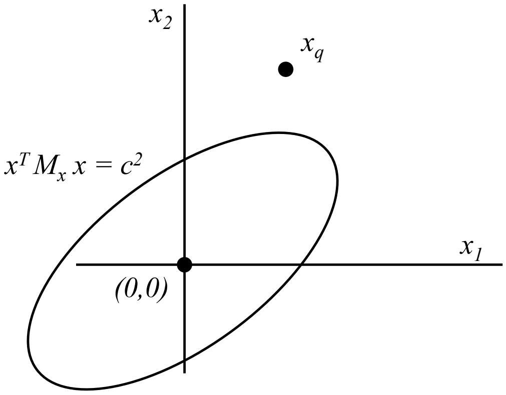
$c^2 = \varepsilon_{tol}{}^2$

With these substitutions, the EOA equation becomes $x^T M_x x = c^2$ and the growth point becomes x_{query}.

A two dimension example includes a symmetric ellipse with a center point at $start_{stored}$ and a growth point at $start_{query}$.



The axes are shifted to align the center point of the ellipse with the origin. The growth point $start_{query}$ becomes x_q after the translation and the EOA is defined in terms of x.



Step 2: Transform the coordinates to map the EOA to a unit hypersphere

In this step a matrix T_{yx} is computed to map the original x-coordinates onto a new y-coordinate system that transforms the EOA into a unit hypersphere. A unit hypersphere is simply a higher-dimensional generalization of the three-dimensional sphere with radius of one. The matrix T_{yx} maps all points in x into the y coordinates with the relation $y = T_{yx} x$. Likewise, the inverse of T_{yx} (or T_{xy}) maps y into the x coordinates with $x = T_{yx}^{-1} y = T_{xy} y$. The first subscript letter of T refers to the transformed coordinate system while the second subscript letter refers to the original coordinates.

A Schur decomposition gives $M_x = Q_x \Sigma_x Q_x^T$ with Q_x being a unitary matrix ($Q_x^T = Q_x^{-1}$). The square root of the diagonal matrix Σ_x is computed by taking the square root of the individual elements along the diagonal. The transformation matrix becomes $T_{yx} = c^{-1} \Sigma_x^{1/2} Q_x^T$.

It will now be shown that the coordinate transform does, in fact, transform the EOA to a unit hypersphere in the new coordinate system. First, the inverse of T_{yx} is found to be

$$T_{yx}^{-1} = T_{xy} = c\, Q_x \Sigma_x^{-1/2}$$

Making the substitution $T_{xy} y = x$ in the EOA equation $x^T M_x x = c^2$ gives

$$(c\, Q_x \Sigma_x^{-1/2} y)^T M_x (c\, Q_x \Sigma_x^{-1/2} y) = c^2$$

Rearranging and substituting $M_x = Q_x \Sigma_x Q_x^T$ gives

$$c^2 (y^T \Sigma_x^{-1/2} Q_x^T) Q_x \Sigma_x Q_x^T (Q_x \Sigma_x^{-1/2} y) = c^2$$

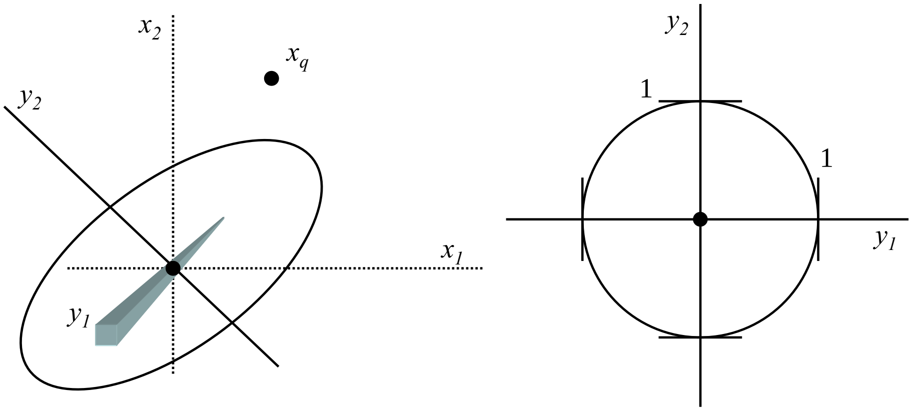
The c^2 term cancels and $Q_x^T Q_x = I$ since Q_x is a unitary matrix. This leaves

$$y^T \Sigma_x^{-1/2} \Sigma_x \Sigma_x^{-1/2} y = 1$$

Finally, since $\Sigma_x^{-1/2} \Sigma_x \Sigma_x^{-1/2} = I$ the EOA in transformed space becomes a unit hypersphere.

$$y^T I \;\, y = 1$$

Returning to the two dimensional example, the y-axes are shown relative to the x-axes. In the y-axes reference frame, the ellipse becomes a unit circle centered at the origin.



<u>*Step 3: Map the growth point to the transformed coordinates*</u>

The same transformation matrix T_{yx} is used to transform the growth point to the new coordinates.

$$y_q = T_{yx} x_q$$

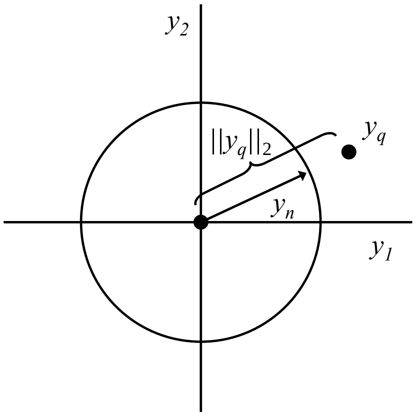
The magnitude and normalized direction of the vector y_q are important for subsequent calculations. The magnitude is the 2-norm of the vector.

$$mag(y_q) = \|y_q\|_2$$

The normalized direction is simply the vector divided by the magnitude.

$$y_n = \frac{y_q}{\|y_q\|_2}$$

Graphically, the growth point is translated to the new *y* reference frame. The magnitude is the distance between y_q and the origin. The normalized vector y_n has a unit length and points in the direction of y_q.



Step 4: Align one of the y-axes with the direction of the growth point

One of the y-axes must be aligned with the direction of the growth point. This is accomplished by computing an orthonormal basis to y_n. An orthonormal basis is produced by first subtracting the outer product of y_n from the identity matrix of appropriate dimension.

$$R = I - y_n y_n^T$$

A Schur decomposition of *R* gives

$$R = Q_R\ \Sigma_R\ Q_R^T$$

The diagonal matrix Σ_R is equal to the identity matrix except that one of the diagonal elements is zero. This diagonal element corresponds to the axis that is aligned with y_q in the *z* coordinate system. The transformation matrix is the transpose of the unitary matrix from the Schur decomposition.

$$T_{zy} = Q_R^T$$

A new coordinate system is defined by $z = T_{zy}\, y$. Transforming the y coordinates to the *z* coordinates the EOA becomes

$$(T_{yz}\, z)^T I\ \ (T_{yz}\, z) = 1$$

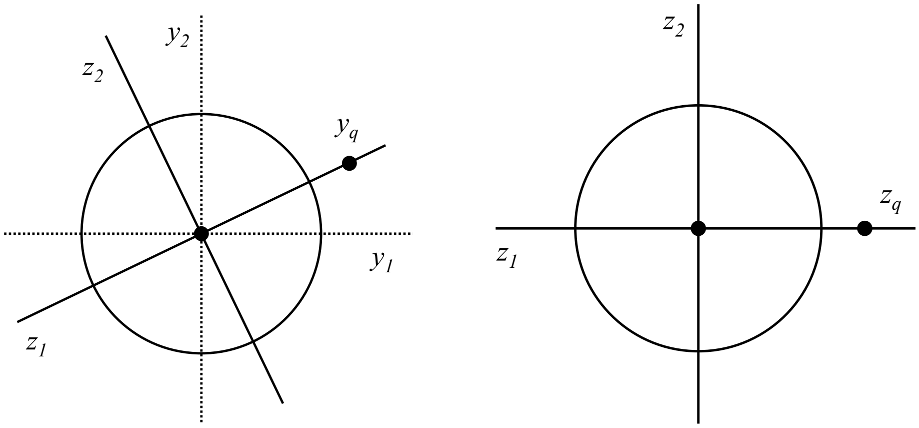
Rearranging gives

$$z^T\, T_{yz}^T\, T_{yz}\, z = 1$$

Since T_{yz} has the special property of a unitary matrix that $T_{yz}^T\, T_{yz} = I$, the EOA is also a unit hypersphere in the z coordinates.

$z^T I\ z = 1$

Graphically, the axes are rotated so that one axis aligns with the growth point. This rotation is important so that the ellipse can be expanded along the aligned axis.

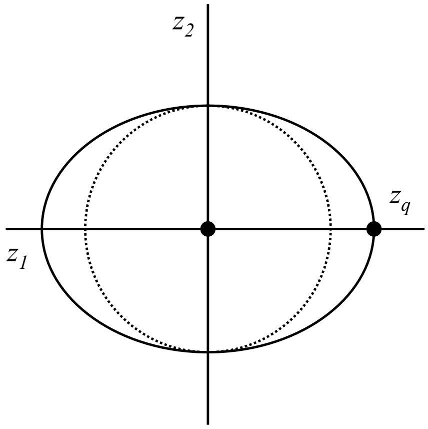


Step 5: Grow the hypersphere into an ellipsoid that reaches the growth point

The half length of the axis, aligned with y_q, is expanded by modifying the appropriate element of the identity matrix. This is the same element that corresponds to the zero diagonal element of Σ_R. In this case, the first diagonal element is shown as the appropriate selection. The half length of the ith axis of an ellipsoid $z^T M z = c^2$ is $(c^2/\sigma_i)^{1/2}$ where σ_i is the ith eigenvalue of M. In order to stretch the hypersphere into an ellipsoid that includes z_q, the half length is lengthened to the magnitude of z_q. Since the magnitude of z_q is equal to that of y_q, the matrix element is set to $\|y_q\|_2^{-2}$.

$$M_z = \begin{bmatrix} \|y_q\|_2^{-2} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

The grown EOA is $z^T M_z z = 1$. Graphically, the circle is expanded into an ellipse that reaches the growth point. This is a minimum area expansion of a symmetric ellipse.



Step 6: Transform the expanded ellipsoid back to the original coordinate system

The grown EOA is transformed back to the original coordinate system with inverse transformation matrices. The z coordinates are a function of the x coordinates according to

$$z = T_{zy}\, y = T_{zy}\, T_{yx}\, x$$

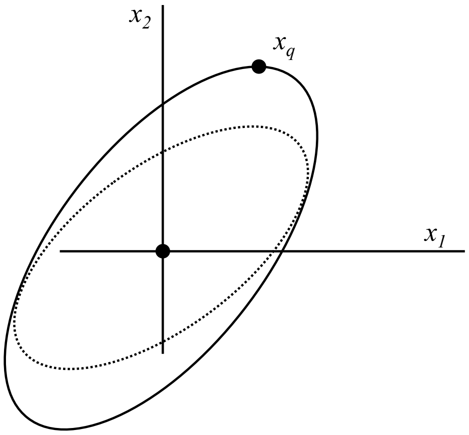
Substituting for z in the EOA equation and multiplying both sides of the equation by c^2 reverts back to the x coordinates and recovers the form of the original EOA. This is the minimum volume ellipsoid that includes the original ellipsoid and the growth point.

$$x^T M_x^{expanded}\, x = c^2$$

with

$$M_x^{expanded} = c^2\, (T_{yx}^{\;T}\, T_{zy}^{\;T}\, M_z\, T_{zy}\, T_{yx})$$

Graphically, when the ellipse is transformed back to the original coordinates, x_q is on the ellipse perimeter. In addition, the ellipse is a symmetric minimum area expansion that includes the growth point and the original ellipse.



ISAT Addition

When $|finish - finish_{est}| > \varepsilon_{tol}$ the EOA should not be expanded. Instead a new record should be added to the ISAT database. The core elements of an ISAT record are *start*, *finish*, *sensitivity*, and *eoa*. Each of these elements is discussed in the subsequent sections.

ISAT record element: start

The vector *start* is the query point that is not eligible for retrieval or growth. This addition point becomes the center of the symmetric EOA.

$$start = start_{query}$$

ISAT record element: finish

The vector *finish* comes from a function evaluation ($finish = f(start_{query})$). There is no speed up with ISAT growth or ISAT addition since a function evaluation is required. ISAT growth and addition are part of the database building phase. The real advantage of ISAT occurs when retrievals greatly outnumber growths and additions. The vector *finish* is stored to provide an estimate for retrievals. In addition, *finish* is used to test a *start* vector to determine if it should be a growth point or addition point.

ISAT record element: sensitivity

Sensitivity information may also be optionally available from the function evaluation. The sensitivity matrix reveals the amount that *finish* changes with a small perturbation in *start*.

$$sensitivity = \frac{\partial\, finish}{\partial start}$$

When the sensitivity is not available from the function evaluation a statistical approximation can be made. At least *m*, where *m* is the dimension of the vector *start*, function evaluations are required to calculate an accurate sensitivity. The function evaluations can be obtained by sorting through a database of previous results or by generating new results. When sorting through a database of previous results, care should be taken to select records that are close to $start_{query}$ otherwise the sensitivity may not be locally accurate. When new results option is selected, *m* linearly independent vectors of *start* should be generated around $start_{query}$. Performing a function evaluation *m* times for all of the *start* vectors can be a cpu time intensive step.

Once the *m* function evaluations are completed, the sensitivity can be estimated through multivariate linear regression. Each of the *start* and corresponding *finish* vectors are first subtracted from $start_{query}$ and $finish_{query}$.

$$\Delta\, start = start_{query} - start$$

$$\Delta\, finish = f(start_{query}) - finish \;\; = finish_{query} - finish$$

The linear regression model includes a residual vector, ε_{resid}, as an indication of how much each record deviates from the linear model. A large residual indicates that a perturbation of *start* does not fit in with the linear model. This could indicate that the perturbation of *start* should be reduced to generate locally linear solutions to *finish*.

$$\Delta\, finish = sensitivity\ \Delta\, start + \varepsilon_{resid}$$

The vectors are assembled into matrices *X* and *Y*.

$$X = \begin{bmatrix} \Delta start_1 \\ \vdots \\ \Delta start_m \end{bmatrix}^T$$

$$Y = \begin{bmatrix} \Delta\, finish_1 \\ \vdots \\ \Delta\, finish_m \end{bmatrix}^T$$

An estimate of the sensitivity is calculated by standard matrix multiplications.

$$sensitivity = Y\, X^T\, (X\, X^T)^{-1}$$

ISAT record element: eoa

An initial estimate of the EOA should be conservative for good error control. A conservative estimate is derived by calculating an approximate EOA for a zero-order function estimation. ISAT normally uses a first-order approximation for estimating *finish*. The higher order terms have been truncated from this approximation.

$$finish_{est} = finish_{stored} + \frac{\partial\, finish_{stored}}{\partial start_{stored}}\left(start_{query} - start_{stored}\right)$$

By assuming a zero-order function estimation, the first-order term becomes an approximation to the truncation error.

$$finish_{est} = finish_{stored}$$

$$\varepsilon_{trunc} \approx \frac{\partial\, finish_{stored}}{\partial start_{stored}}\left(start_{query} - start_{stored}\right) = Ax$$

Recall that the equation for the EOA is given by $x^T M_x x = \varepsilon^2$. Here, A is the sensitivity and $x = start_{query} - start_{stored}$. Substituting $\varepsilon_{\mathrm{trunc}}$ for the error term in this equation gives an approximation for the zero-order EOA.

$$x^T M_x x = \varepsilon^2 = \varepsilon_{trunc}{}^T \varepsilon_{trunc} = (A\,x)^T (A\,x) = x^T (A^T A)\, x$$

$$M_x^{zero\text{-}order} = A^T A$$

Sometimes the zero-order approximation produces an EOA with a large principle axis because of a low sensitivity in a particular direction. To remedy this problem, the singular values of A are adjusted to be at least $\varepsilon_{tol}{}^2 / 2$. To accomplish this, a singular value decomposition of A is performed to give $U \Sigma V^T$. Any diagonal elements below $\varepsilon_{tol}{}^2 / 2$ are raised to that value. The corrected matrix is reconstructed from the new diagonal matrix of singular values.

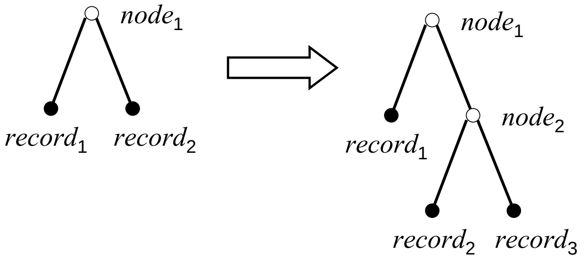
$$\tilde{A} = U\,\tilde{\Sigma}\,V^T$$

The corrected zero-order approximation of the EOA ensures that large principle axes are eliminated.

$$\tilde{M}_x^{\,zero-order} = \tilde{A}^T \tilde{A}$$

<u>ISAT record placement in the binary tree</u>

Once all of the ISAT record elements are computed, the record is added to the binary tree. The growth of the binary tree involves the creation of a new node. In this case, the record added to the tree is *record₃*. Supposing that *record₃* is closer to *record₂*, the tree is grown on the right branch with the creation of *node₂*.



The new *node₂* is defined by the following equations.

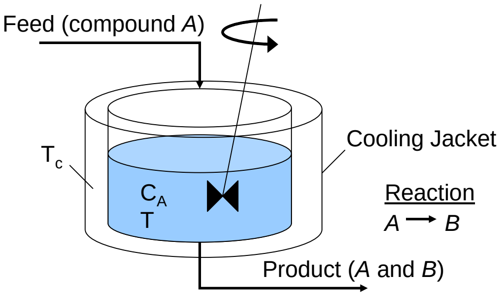
$$v_2 = start_3 - start_2$$

$$a_2 = v_2{}^T\left(\frac{start_3 + start_2}{2}\right)$$

As a final step, the appropriate pointers are defined by *lt* pointing to *record*$_2$ and *gt* pointing to *record*$_3$.

Example Problem

A simple academic problem is considered to show the applicability of ISAT to storage and retrieval of optimal control. A perfectly mixed, adiabatic CSTR has an exothermic reaction of compound *A* transformed into compound *B*.

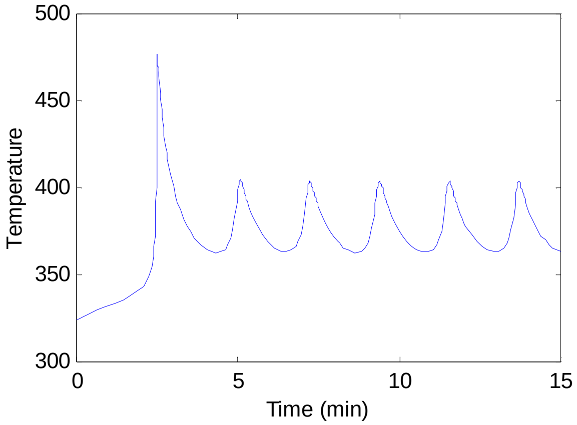


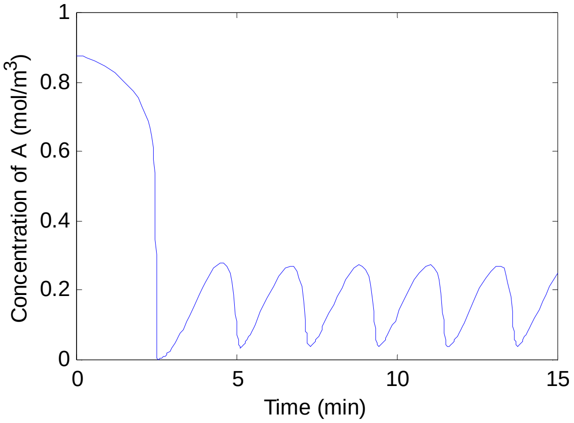
Temperature control of the reactor is a challenge due to the highly exothermic reaction (ΔH_{rxn} = 50,000 J/mol). The temperature of the fluid in the jacket surrounding the CSTR is manipulated to control the temperature of the reactor fluid. The dynamics of the reactor are described by a set of ODEs generated from a mole balance on *A* and an energy balance on the reactor.

$$\frac{dC_A}{dt} = \frac{q}{V}\left(C_f - C_A\right) - k_0 C_A \exp\left(-\frac{E}{RT}\right)$$
$$\frac{dT}{dt} = \frac{q}{V}\left(T_f - T\right) + \frac{\Delta H_{rxn}}{\rho C_p}\left(k_0 C_A \exp\left(-\frac{E}{RT}\right)\right) - \frac{UA}{V\rho C_p}\left(T_c - T\right)$$

Variable	Description	Nominal Value
T_c	Temperature of the cooling jacket fluid	300 K
C_A	Concentration of *A* in the reactor	0.877 mol/m^3
T	Temperature of the reactor fluid	325 K
q	Feed volumetric flow rate	100 m^3/min
V	Reactor volume	100 m^3
ρ	Density of *A* and *B*	1000 kg/m^3
C_p	Heat capacity of *A* and *B*	0.239 J/kg-K
ΔH_{rxn}	Heat of reaction	5e4 J/mol
E/R	Activation energy divided by the universal gas constant	8750 K
k_0	Frequency factor	7.2e10 1/min
UA	Overall heat transfer coefficient multiplied by the contact area	5e4 J/min-K
C_f	Feed concentration of *A*	1 mol/m^3
T_f	Feed temperature	350 K

At a constant cooling temperature of 305 K, the reactor temperature spikes continuously as the reactor goes through cycles of concentration buildup followed by moments of intense reaction.





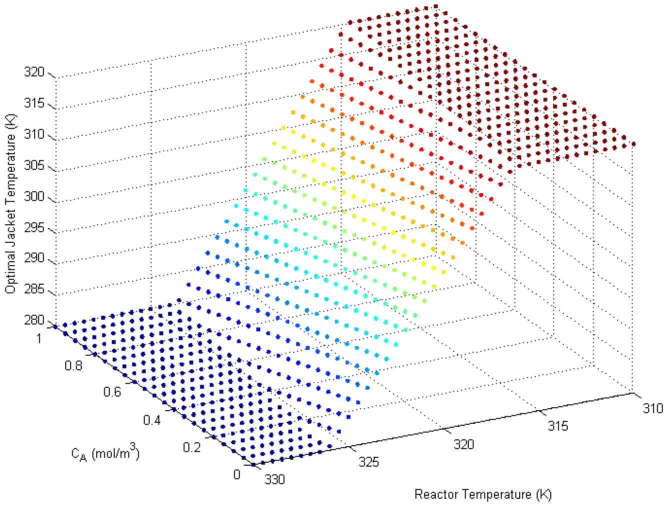
The unsteady response of the reactor with a constant cooling jacket temperature suggests that unsteady control may be necessary when pushing the reactor to the stability limit. A sequential direct single shooting approach to dynamic optimization is used as the control algorithm. The N-step finite-horizon NMPC problem formulation is given by the following.

$$\min_u \Phi(x,u) \quad s.t. \quad x_0 \; given, \; x_{k+1} = F(x_k, u_k), \; Du_k \le d$$

$$\Phi(x,u) \overset{def}{=} \sum_{k=1}^{N} \left[(x_k - x_{set\,po\,\mathrm{int}})^T Q (x_k - x_{set\,po\,\mathrm{int}}) + (u_k - u_{k-1})^T R (u_k - u_{k-1}) \right]$$

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad R = [0] \quad D = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad d = \begin{bmatrix} 320 \\ -280 \end{bmatrix}$$

In this problem formulation, Φ (the cost function) is quadratic in *x* (states) and *u* (manipulated variables) and therefore strictly convex. The source of nonlinearity comes from the model function $F(x_k,u_k)$ that is solved by integrating the ODE model. With a constant reactor temperature set point, the first optimal control step u_0^* is a unique function of the current concentration and temperature of the reactor. The optimal cooling jacket temperature (u_0^*) to drive the reactor temperature to 320 K was calculated for reactor concentrations between 0 and 1 mol/m^3 and reactor temperatures between 310 and 330 K.



Even though the model is highly nonlinear, the optimal control surface is surprisingly linear with respect to the state initial conditions. With clipping of the ISAT predicted value to meet the control constraints, only one record is required to store all of the optimal control solutions.

Elements of the ISAT Record	Alias	Values
initial conditions	*start*	$\begin{bmatrix} C_A \\ T \end{bmatrix} = \begin{bmatrix} 0.5 \\ 320 \end{bmatrix}$
optimal control value	*finish*	$\left[T_c^*\right] = [296]$
sensitivity	*sensitivity*	$\begin{bmatrix} \frac{\partial T_c^*}{\partial C_A} & \frac{\partial T_c^*}{\partial T} \end{bmatrix} = \begin{bmatrix} -6.2 & -4.1 \end{bmatrix}$
ellipsoid of accuracy	*eoa*	$eoa = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Conclusions

Slow computational cycle time can degrade the performance of controllers based on large, non-linear first principles models. Storage and retrieval of the optimal control is desirable since the same control problem is repeatedly solved with different initial conditions. ISAT is a storage and retrieval method to store the optimal control values as a function of the initial conditions. These initial conditions include any adjustable parameter that influences the optimal control values. ISAT stores the optimal control values in linear ellipsoid regions with a strategy to control the amount of permissible error. Once the permissible error is exceeded, a new linear region is added to the ISAT database. In this manner, a complete mapping of the optimal control surface can be obtained for future retrieval. An example problem involving the temperature control of a CSTR demonstrates the capability of ISAT to compactly store the optimal control value as a function of the initial conditions. In this case only one ISAT record was required to store all of the optimal control solutions.