Singular optimal control for stochastic linear quadratic singular system using ant colony programming

N. Kumaresan* and P. Balasubramaniam

Department of Mathematics, Gandhigram Rural University, Gandhigram – 624 302, Tamil Nadu, India

(Received 22 June 2007; revised version received 26 March 2009; accepted 03 May 2009)

In this article, singular optimal control for stochastic linear singular system with quadratic performance is obtained using ant colony programming (ACP). To obtain the optimal control, the solution of matrix Riccati differential equation is computed by solving differential algebraic equation using a novel and nontraditional ACP approach. The obtained solution in this method is equivalent or very close to the exact solution of the problem. Accuracy of the solution computed by the ACP approach to the problem is qualitatively better. The solution of this novel method is compared with the traditional Runge Kutta method. An illustrative numerical example is presented for the proposed method.

Keywords: ant colony programming; differential algebraic equation; matrix Riccati differential equation; Runge Kutta method; singular optimal control; stochastic linear singular system

AMS (MOS) Subject Classifications: 49 K 45; 68 N 19; 93 E 20

1. Introduction

Ant colony programming (ACP) is a metaheuristic approach that is inspired by the behaviour of real ant colonies to find a good enough solution to the given problem in a reasonable amount of computation time. It allows the programmer to avoid the tedious task of creating a program to solve a well-defined problem [9]. ACP is a stochastic search technique that is carried out on a space graph where the nodes represent functions, variables and constants. Functions are usually defined mathematically in terms of arithmetic operators, operands and Boolean functions. The set of functions defining a given problem is called a function set $F$ and the collection of variables and constants to be used are known as the terminal set $T$.

Ants are able to find their way efficiently from their nest to food sources. While searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trial on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will...
Figure 1. Shortest path of ants

guide other ants to the food source. If an ant has a choice of trails to follow, the preferred route is the trail with the highest deposit of pheromone [25]. This behaviour helps the ants to find the optimal route without any need for direct communication or central control. Therefore, artificial ants used in the ACP have some features taken from the behaviour of real ants, for example,

(a) artificial ants move in a random fashion;
(b) choice of a route of an artificial ant depends on the amount of pheromone;
(c) artificial ants co-operate in order to achieve the best result.

The ant colony algorithm can be described at a very simplified level as given in Figure 1. Two ants A1 and A2 are travelling along route P and come to a junction. A1 takes path A and A2 takes path B. As they are travelling along the route, the ants are depositing a pheromone trail. Both ants continue along their chosen paths, collect the food and return to the nest. A1 will reach the nest first because it has travelled the shortest route. A third ant A3 now leaves the nest, travels along path P and reaches the junction. At this point, A2 has not yet returned through the junction and is still travelling along path B, so there is twice the amount of pheromone deposited along path A at the junction as along path B. Therefore, A3 will opt for path A. Thus increasing the pheromone level on path A. In fact, experiments by biologists have shown that ants probabilistically prefer paths with high pheromone concentration. Dorigo et al. [17,18] used the ant colony algorithm for solving the travelling salesman problem. Roux and Fonlupt [23] made the first attempt to apply the ant colony algorithm for solving symbolic regression and multiplexer problem. Recently, researchers have been dealing with the relation of ant colony algorithm to other methods for learning, approximations and optimization. They have applied in the field of optimal control and reinforcement learning [7]. In this article, the ant colony algorithm is used in ACP to compute singular optimal control for stochastic linear quadratic singular system.

Stochastic linear quadratic regulator (LQR) problems have been studied by many researchers [1,6,10,16,26]. Chen et al. [13] have shown that the stochastic LQR problem is well posed if there are solutions to the Riccati equation and then an optimal feedback control can be obtained. For LQR problems, it is natural to study an associated Riccati equation. However, the existence and uniqueness of the solution of the Riccati equation in general seem to be very difficult problems due to the presence of the complicated nonlinear term. Zhu and Li [27] used the iterative method for solving stochastic Riccati equations for stochastic LQR problems. There are several numerical methods to solve the conventional Riccati equation and as a result of the nonlinear process, essential errors may accumulate. In order to minimize the error, recently the conventional Riccati equation has been analysed using neural network approach and genetic programming approach [2–5,19–21,24]. A variety of numerical algorithms [14] have been developed for solving the algebraic Riccati equation.

Singular systems contain a mixture of algebraic and differential equations. In that sense, the algebraic equations represent the constraints to the solution of the differential part. These systems
are also known as degenerate, differential algebraic, descriptor or semi-state and generalized state space systems. The complex nature of singular system causes many difficulties in the analytical and numerical treatment of such systems, particularly when there is a need for their control. The system arises naturally as a linear approximation of system models or linear system models in many applications such as electrical networks, aircraft dynamics, neutral delay systems, chemical, thermal and diffusion processes, large-scale systems, robotics, biology, etc. [11,12,22]. As the theory of optimal control of linear systems with quadratic performance criteria is well developed, the results are most complete and close to use in many practical designing problems. The theory of the quadratic cost control problem has been treated as a more interesting problem and the optimal feedback with minimum cost control has been characterized by the solution of a Riccati equation. Da Prato and Ichikawa [15] showed that the optimal feedback control and the minimum cost are characterized by the solution of a Riccati equation. Solving the matrix riccati differential equation (MRDE) is the central issue in optimal control theory.

Although parallel algorithms can compute the solutions faster than sequential algorithms, there have been no report on ACP solutions for MRDE. This article focuses on the implementation of ACP approach for solving MRDE in order to get the optimal solution. The optimal control obtained by this approach is called singular optimal control since the matrix $R$ is singular in the performance index $J$.

This article is organized as follows. In Section 2, the statement of the problem is given. In Section 3, solution of the MRDE is presented. In Section 4, numerical example is discussed to illustrate the proposed method. The final conclusion section demonstrates the efficiency of the method.

2. Statement of the problem

Consider the linear dynamical singular system that can be expressed in the form:

$$ F \, dx(t) = [Ax(t) + Bu(t)]dt + Du(t)dW(t), \quad x(0) = x_0, \quad t \in [0, t_f], \quad (1) $$

where the matrix $F$ is possibly singular, $x(t) \in \mathbb{R}^n$ is a generalized state space vector, $u(t) \in \mathbb{R}^m$ is a control variable and it takes value in some Euclidean space, $W(t)$ is a Brownian motion and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $D \in \mathbb{R}^{n \times m}$ are known coefficient matrices associated with $x(t)$ and $u(t)$, respectively, $x_0$ is given initial state vector, $t_f$ is the final time and $m \leq n$.

In order to minimize both state and control signals of the feedback control system, a quadratic performance index is usually minimized:

$$ J = E \left\{ \frac{1}{2} x^T(t_f)F^T S F x(t_f) + \frac{1}{2} \int_0^{t_f} [x^T(t)Q x(t) + u^T(t) R u(t)] \, dt \right\}, $$

where the superscript $T$ denotes the transpose operator, $S \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times n}$ are symmetric and positive definite (or semi-definite) weighting matrices for $x(t)$, $R \in \mathbb{R}^{m \times m}$ is a singular weighting matrix for $u(t)$. It will be assumed that $|sF - A| \neq 0$ for some $s$. This assumption guarantees that any input $u(t)$ will generate one and only one state trajectory $x(t)$.

If all state variables are measurable, then a linear state feedback control law

$$ u(t) = -(R + D^T K(t) D)^{-1} B^T \lambda(t) $$

can be obtained for the system described by Equation (1), where

$$ \lambda(t) = K(t) F x(t), \quad K(t) \in \mathbb{R}^{n \times n} $$

is a symmetric matrix and the solution of MRDE.
The relative MRDE for the stochastic linear singular system (1) is
\[ F^T \dot{K}(t)F + F^T K(t)A + A^T K(t)F + Q - F^T K(t)B(R + D^T K(t)D)^{-1}B^T K(t)F = 0 \] (2)
with terminal condition (TC) \( K(t_f) = F^T SF \) and \( (R + D^T K(t)D) > 0 \).

After substituting the appropriate matrices in the above equation, it becomes a differential algebraic equation (DAE) of index one. Therefore, solving MRDE is equivalent to solving the DAE of index one.

3. Solution of MRDE

Consider the DAE for Equation (2)
\[
\begin{align*}
\dot{k}_{ij}(t) &= \phi_{ij}(k_{ij}(t)), \quad k_{ij}(t_f) = A_{ij} \quad (i, j = 1, 2, \ldots, n - 1) \\
K_{1n}(t) &= \psi(k_{ij}(t)), \quad K_{1n}(t_f) = A_{1n}.
\end{align*}
\] (3)

3.1 Runge Kutta method

Numerical integration is one of the oldest and most fascinating topics in numerical analysis. It is the process of producing a numerical value for the integration of a function over a set. Numerical integration is usually utilized when analytic techniques fail. Even if the indefinite integral of the function is available in a closed form, it may involve some special functions, which cannot be computed easily. In such cases we can also use numerical integration. Runge Kutta (RK) algorithms have always been considered as the best tool for the numerical integration of ordinary differential equations (ODEs). The DAE can be changed into system of nonlinear differential equation by differentiating the algebraic equation one time, since the DAE is of index one type. The system (3) contains \( n^2 \) first-order ODEs with \( n^2 \) variables. In particular \( n = 2 \), the system will contain four equations. Since the matrix \( K(t) \) is symmetric and the system is singular, \( k_{12} = k_{21} \) and \( k_{22} \) is free (let \( k_{22} = 0 \)). Finally, the system will have two equations with two variables. Hence, RK method is explained for a system of two first-order ODEs with two variables.

\[
\begin{align*}
k_{11}(i + 1) &= k_{11}(i) + \frac{1}{6}(k1 + 2k2 + 2k3 + k4) \\
k_{12}(i + 1) &= k_{12}(i) + \frac{1}{6}(l1 + 2l2 + 2l3 + l4),
\end{align*}
\]

where
\[
\begin{align*}
k1 &= h \ast \phi_{11}(k_{11}, k_{12}) \\
l1 &= h \ast \phi_{12}(k_{11}, k_{12}) \\
k2 &= h \ast \phi_{11}\left(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2}\right) \\
l2 &= h \ast \phi_{12}\left(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2}\right) \\
k3 &= h \ast \phi_{11}\left(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2}\right) \\
l3 &= h \ast \phi_{12}\left(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2}\right)
\end{align*}
\]
\begin{align*}
k_4 &= h \ast \phi_{11}(k_{11} + k_3, k_{12} + l_3) \\
l_4 &= h \ast \phi_{12}(k_{11} + k_3, k_{12} + l_3).
\end{align*}

In the similar way, the original system (2) can be solved for \( n^2 \) first-order ODE’s.

### 3.2 ACP method

In this approach, ACP is used to obtain a set of expressions. If the required number expressions satisfy the fitness function, it will be the optimal solution of Equation (3). The scheme of computing optimal solution is given in Figure 2.

According to Boryczka and Wiezorek [9], the following four preparatory steps are essential for a searching process.

- Choice of terminals and functions
- Construction of graph
- Defining fitness function
- Defining terminal criteria.

![Flow chart](attachment:flow_chart.png)

Figure 2. Flow chart.
3.2.1 Choice of terminals and functions

A terminal symbol $t_i \in T$ can be a constant or a variable. Every function $f_i \in F$ can be an arithmetic operator \{+,-,\*,/\}, an arithmetic function (sin, cos, exp, log) and an arbitrarily defined function appropriate to the problem under consideration. The terminal symbols and functions have chosen such that they provided sufficient expressive power to express the solution to a problem. This means that the problem must be solved by a composition of functions and terminals specified. For solving the DAE (3), terminal set and function set are taken as $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, t\}$ and $F = \{+,-,\*,/\sin, \cos, \exp, \log\}$.

3.2.2 Construction of graph

In ACP technique, the search space consists of a graph with $\ell$ nodes where the nodes are the functions or terminals and edges are weighted by pheromone. The examples of such a graph are given in Figures 3 and 4. Each node in the graph holds either a function or a terminal. This graph is generated by a randomized process.

Figure 3. Graph with functions and terminals.

Figure 4. Graph with functions and terminals.
3.2.3 Fitness function

The aim of the fitness function is to provide a basis for competition among available solutions and to obtain the optimal solution. Hence, the fitness function for Equation (3) is defined as

\[ E_r = (k_{1n}(t_m) - \varphi(k_{ij}(t_m)))^2 + \sum_{i,j=1}^{n-1} (\dot{k}_{ij}(t_m) - \phi_{ij}(k_{ij}(t_m)))^2, \quad (m = 0, 1, 2, \ldots, t_f), \] (4)

where \( m \) represents the equidistance points in the relevant range \([0, t_f]\).

3.2.4 Terminal criteria

The group of ants and their collective tours form a generation. In each generation, a set of expressions are generated by the artificial ants. If the required number of expressions minimize the fitness function, \( E_r \) tends to zero or very close to zero and they satisfy the terminal conditions, the process may be stopped; otherwise continue the ACP approach.

3.2.5 ACP methodology

Artificial ants build solutions by performing randomized tours on the completely connected graph \( G(V, E) \). In the graph, vertices \((V)\) are represented by functions and terminals and the set \((E)\) of edges connect the vertices. The ants move on the graph by applying a stochastic local decision policy that makes use of pheromone trials and heuristic information. In this way, ants incrementally build solutions to the given problem.

In the first generation, all edges are initialized by equal pheromone weight. Send \( k \) (< \( \ell \)) ants through the graph from \( k \) starting points in a random fashion. Each ant is initially put on a randomly chosen start node. Each ant is moving from the node \( r \) to node \( s \) in the graph at time \( t \) according to the following probability law [9]

\[ p_{rs}(t) = \frac{\tau_{rs}(t) \cdot [\gamma_s]^\beta}{\sum_{i \in J_k} [\tau_{ri}(t)] \cdot [\gamma_i]^\beta}, \]

where \( \gamma_s = (1/(2 + \pi_s))^d \), \( \pi_s \) is the power of symbol \( s \) that can be either a terminal symbol or a function, \( d \) is the current length of the arithmetic expression, \( \beta \) is a parameter that controls the relative weight of the pheromone trail and visibility and \( J_k \) is the set of unvisited nodes. The power of the symbols can be calculated from Table 1. When an ant reaches a node, it determines if the node is a terminal or a function node. If the ant is on a terminal node, the end of the tour has been reached for that ant.

After having found a tour of an ant, the ant deposits pheromone information on the edges through which it travelled. It constitutes a local update of the pheromone trial, which also comprises partial

<table>
<thead>
<tr>
<th>Terminal symbol or function</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant, variable</td>
<td>-1</td>
</tr>
<tr>
<td>Functions</td>
<td>1</td>
</tr>
</tbody>
</table>

Downloaded By: [Kumaraswarup, N. T. At: 06:45 22 November 2010]
evaporation of the trial. The local update process is carried out according to the formula:
\[
\tau_{ij}(t + 1) = (1 - \rho).\tau_{ij}(t) + \rho.\tau_0,
\]
where \( (1 - \rho), \rho \in (0, 1], \) is the pheromone decay coefficient (\( \rho \) is the concentration of pheromone on edge over time, \( \rho > 0.5 \) yields good solution), \( \tau_{ij} \) is the amount of pheromone trial on edge \((i, j)\) and \( \tau_0 \) is the initial amount of pheromone on edge \((i, j)\).

Each ant has a working memory that stores data about its tour. The ant’s memory is represented programmatically by a parse tree structure. In this tree, the root and branches are functions and leaves are terminals. The depth of the memory tree is limited according to the nature of the problem. The tour \( e * t + 1 * +5 \) of an ant is represented as parse tree in Figure 5. The tour \( e * t/7 * +4/5 \) of another ant is represented as parse tree in Figure 6. The tours of the ants and their corresponding expressions extracted from the parse trees are given in Table 2. Some tours of the ants cannot be represented as the parse trees. Such type of tours are given in Table 3. They are discarded when the parse tree construction process is carried out for the tours of the ants. This parse tree construction is helpful to converge the solution quickly and also reduces the computation time by discarding the unnecessary tours. After each generation, a global update of pheromone trail is taken place. The level of pheromone is then changed as follows:
\[
\tau_{ij}(t + g) = (1 - \rho).\tau_{ij}(t) + \rho.\frac{1}{L},
\]
Table 2. Tours and expressions.

<table>
<thead>
<tr>
<th>Tours of ants</th>
<th>Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>e * t</td>
<td>e^t</td>
</tr>
<tr>
<td>e * t + 1 * + 5</td>
<td>e^{t+1} + 5</td>
</tr>
<tr>
<td>e * t/5 * + 0</td>
<td>e^{t/5} + 0</td>
</tr>
<tr>
<td>e * t/7 * + 4/5</td>
<td>e^{t/7} + 4/5</td>
</tr>
<tr>
<td>e * 3 * t - 2 * * + 5/2</td>
<td>e^{3(t-2)} + 5/2</td>
</tr>
<tr>
<td>e * 3 * t * + 5/2 - e * t * / 7</td>
<td>e^{3t/2 - e^{t/7}}</td>
</tr>
</tbody>
</table>

Table 3. Discarded tours.

<table>
<thead>
<tr>
<th>Tours of ants</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e * t * + 1</td>
</tr>
<tr>
<td>e * t + 1 * + - 5</td>
</tr>
<tr>
<td>e * t/ - * 2</td>
</tr>
<tr>
<td>- * e * t/ / 7 * + - / + 0</td>
</tr>
</tbody>
</table>

Figure 7. Optimal tour of the ACP.

where \( g \) is the number of generations, edges \((i, j)\) belong to the optimal tour found so far and \( L \) is the length of this tour. The aim of the pheromone value update rule is to increase the pheromone values on the solution path. The update rule reduces the size of the searching region in order to find high-quality solution with reasonable computation time. On the updated graph, the consecutive cycles of the ant colony algorithm are carried out by sending the ants through the best tour of the previous generation. The procedure is repeated until the fitness function (4) becomes zero or very close to zero. The optimal tour of the ACP and its corresponding tree are given in Figures 7 and 8, respectively.
3.2.6 ACP algorithm

Step 1 Construct a graph with \( \ell \) nodes.

Step 2 Initialize the equal weight of pheromone in each edge of the graph.

Step 3 Pass \( k \) ants through the graph from \( k \) starting points and they move to the next node according to the probability law.

Step 4 Apply local update rule after the tour of each ant.

Step 5 Construct parse trees from the tours of \( k \) ants.

Step 6 Extract the expressions from the trees.

Step 7 Evaluate the fitness function.

Step 8 If \( E_r \to 0 \) and satisfy the terminal conditions, then stop. Otherwise apply global update rule.

Step 9 Identify the best tour of the previous generation.

Step 10 Pass the same \( k \) ants through the best tour and go to Step 4.

4. Numerical examples

4.1 Example 1

Consider the optimal control problem:

Minimize

\[ J = E \left\{ \frac{1}{2} x^T(t_f) F^T S F x(t_f) + \frac{1}{2} \int_0^{t_f} [x^T(t) Q x(t) + u^T(t) R u(t)] dt \right\} \]

subject to the stochastic linear singular system

\[ F \, dx(t) = [A x(t) + B u(t)] dt + D u(t) dW(t), \quad x(0) = x_0, \]
where

\[
S = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & 2 \\ 0 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad R = 0,
\]

\[
Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \end{bmatrix}.
\]

The numerical implementation could be adapted by taking \(t_f = 2\) for solving the related MRDE of the above linear singular system. The appropriate matrices are substituted in Equation (2), and the MRDE is transformed into DAE in \(k_{11}\) and \(k_{12}\). In this problem, the value of \(k_{22}\) of the symmetric matrix \(K(t)\) is free and let \(k_{22} = 0\). Then the optimal control of the system can be found out by the solution of MRDE.

4.1.1 Solution obtained using ACP

The graph is generated randomly with 15 nodes. Let \(\rho = 0.8\) and \(\beta = 2\). Each edge is initialized by a pheromone weight of 1.0. Ten ants are taken for sending through the graph from 10 different points. The equidistance points of the interval \([0, 2]\) are taken as \(m = 0, 1, 2\).

As the ACP is carried out continuously, the solution of each generation will be improved by the pheromone updating rules. The construction of parse tree for the tour of the ants will converge the optimal solution quickly by discarding some unnecessary tours.

After 75 generations, the value of the fitness function is 78.0398 and the corresponding tours and expressions are given below.

\[
\text{Tours} : k_{11} = e \ast t - 2 \ast -4, \quad k_{12} = e \ast t - 2 \ast -2
\]

\[
\text{Expressions} : k_{11} = e^{(t-2)} - 4, \quad k_{12} = e^{(t-2)} - 2.
\]

After 150 generations, the following expressions satisfy the fitness function and the value of the fitness function tends to 0. The expressions also satisfy the terminal conditions. Hence, the solution of the DAE/MRDE is obtained.

\[
\text{Tours} : k_{11} = e \ast 7/4 \ast 2 - t \ast \ast \ast 18/7 - 4/7, \quad k_{12} = e \ast 7/4 \ast 2 - t \ast \ast \ast 2/7 - 2/7
\]

\[
\text{Expressions} : k_{11} = \frac{18}{7} e^{((7/4)(2-t))} - 4, \quad k_{12} = \frac{2}{7} e^{((7/4)(2-t))} - 2/7.
\]

After 150 generations, the solutions are obtained from the graph. The numerical solutions of MRDE are calculated and displayed in Table 4 using the ACP. Since this problem is having explicit solution, the ACP solution is equivalent to exact solution of the DAE. ACP solution curves are shown in Figures 9 and 10. In Figure 9, the values of \(t\) are noted in the \(X\) axis and the values of \(k_{11}\) are placed in the \(Y\) axis. Similarly in Figure 10, the values of \(t\) and \(k_{12}\) are in \(X\) and \(Y\) axes, respectively. The parse trees for the solutions are given in Figures 11 and 12.

4.2 Example 2

Consider the optimal control problem:

\[
\text{Minimize} \quad J = E \left\{ \frac{1}{2} x^T(t_f)F^TSF x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \right\}
\]
Table 4. Solutions of MRDE after 150 generators.

<table>
<thead>
<tr>
<th>t</th>
<th>$k_{11}$</th>
<th>$k_{12}$</th>
<th>$k_{11}$</th>
<th>$k_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>84.5805</td>
<td>9.1756</td>
<td>84.5826</td>
<td>9.1758</td>
</tr>
<tr>
<td>0.2</td>
<td>59.4343</td>
<td>6.3815</td>
<td>59.4356</td>
<td>6.3817</td>
</tr>
<tr>
<td>0.4</td>
<td>41.7140</td>
<td>4.4126</td>
<td>41.7148</td>
<td>4.4128</td>
</tr>
<tr>
<td>0.6</td>
<td>29.2266</td>
<td>3.0251</td>
<td>29.2272</td>
<td>3.0252</td>
</tr>
<tr>
<td>0.8</td>
<td>20.4269</td>
<td>2.0474</td>
<td>20.4273</td>
<td>2.0475</td>
</tr>
<tr>
<td>1.0</td>
<td>14.2259</td>
<td>1.3584</td>
<td>14.2261</td>
<td>1.3585</td>
</tr>
<tr>
<td>1.2</td>
<td>9.8561</td>
<td>0.8729</td>
<td>9.8562</td>
<td>0.8729</td>
</tr>
<tr>
<td>1.4</td>
<td>6.7767</td>
<td>0.5307</td>
<td>6.7768</td>
<td>0.5308</td>
</tr>
<tr>
<td>1.6</td>
<td>4.6067</td>
<td>0.2896</td>
<td>4.6068</td>
<td>0.2896</td>
</tr>
<tr>
<td>1.8</td>
<td>3.0775</td>
<td>0.1197</td>
<td>3.0776</td>
<td>0.1197</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0000</td>
<td>0.0000</td>
<td>2.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Figure 9. Solution curve for $k_{11}$.

Figure 10. Solution curve for $k_{12}$. 
subject to the stochastic linear singular system

\[ F \, dx(t) = [Ax(t) + Bu(t)]dt + Du(t)dW(t), \quad x(0) = x_0, \]

where

\[
S = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},
\]

\[
R = 0, \quad Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.
\]

The numerical implementation could be adapted by taking \( t_f = 2 \) for solving the related MRDE of the above linear singular system. The appropriate matrices are substituted in Equation (2), and the MRDE is transformed into DAE in \( k_{11}, k_{12} \) and \( k_{13} \). In this problem, the value of \( k_{22}, k_{32}, k_{23} \)
Table 5. Solutions of MRDE after 130 generations.

<table>
<thead>
<tr>
<th>t</th>
<th>$k_{11}$</th>
<th>$k_{12}$</th>
<th>$k_{13}$</th>
<th>$k_{11}$</th>
<th>$k_{12}$</th>
<th>$k_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.4617</td>
<td>0.2197</td>
<td>0.2197</td>
<td>0.4617</td>
<td>0.2198</td>
<td>0.2198</td>
</tr>
<tr>
<td>0.2</td>
<td>0.4715</td>
<td>0.2183</td>
<td>0.2183</td>
<td>0.4715</td>
<td>0.2184</td>
<td>0.2184</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4869</td>
<td>0.2161</td>
<td>0.2161</td>
<td>0.4869</td>
<td>0.2162</td>
<td>0.2162</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5111</td>
<td>0.2126</td>
<td>0.2126</td>
<td>0.5111</td>
<td>0.2127</td>
<td>0.2127</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5489</td>
<td>0.2072</td>
<td>0.2072</td>
<td>0.5490</td>
<td>0.2073</td>
<td>0.2073</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6084</td>
<td>0.1987</td>
<td>0.1987</td>
<td>0.6084</td>
<td>0.1988</td>
<td>0.1988</td>
</tr>
<tr>
<td>1.2</td>
<td>0.7015</td>
<td>0.1854</td>
<td>0.1854</td>
<td>0.7016</td>
<td>0.1855</td>
<td>0.1855</td>
</tr>
<tr>
<td>1.4</td>
<td>0.8477</td>
<td>0.1646</td>
<td>0.1646</td>
<td>0.8477</td>
<td>0.1646</td>
<td>0.1646</td>
</tr>
<tr>
<td>1.6</td>
<td>1.0769</td>
<td>0.1318</td>
<td>0.1318</td>
<td>1.0769</td>
<td>0.1319</td>
<td>0.1319</td>
</tr>
<tr>
<td>1.8</td>
<td>1.4363</td>
<td>0.0805</td>
<td>0.0805</td>
<td>1.4363</td>
<td>0.0805</td>
<td>0.0805</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>2.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

and $k_{33}$ of the symmetric matrix $K(t)$ are free and let $k_{22} = 0, k_{32} = 0, k_{23} = 0$ and $k_{33} = 0$. Then the optimal control of the system can be found out by the solution of MRDE.

4.2.1 Solution obtained using ACP

The graph is generated randomly with 15 nodes. Let $\rho = 0.8$ and $\beta = 2$. Each edge is initialized by a pheromone weight of 1.0. Ten ants are taken for sending through the graph from 10 different points. The equidistance points of the interval $[0, 2]$ are taken as $m = 0, 1, 2$.

As the ACP is carried out continuously, the solution of each generation will be improved by the pheromone updating rules. The construction of parse tree for the tour of the ants will converge the optimal solution quickly by discarding some unnecessary tours.

After 200 generations, the value of the fitness function is 52.6695 and the corresponding tours and expressions are given below.

**Tours**: $k_{11} = e \times t - 9 - 1$, $k_{12} = e \times t - 9 \times 2$

**Expressions**: $k_{11} = e^{(t - 9)} - 1$, $k_{12} = e^{(t - 9)} - 2$
After 300 generations, the following expressions satisfy the fitness function and the value of the fitness function tends to 0. The expressions also satisfy the terminal conditions. Hence, the solution of the DAE/MRDE is obtained.

**Tours:**
\[ k_{11} = e \cdot \frac{9}{4} \cdot t - 2 \ast \ast \ast 14/9 + 4/9, \quad k_{12} = 2/9 - e \cdot \frac{9}{4} \cdot t - 2 \ast \ast \ast 2/9 \]

**Expressions:**
\[ k_{11} = \frac{14}{9} e^{(9/4)(t-2))} - \frac{4}{9}, \quad k_{12} = \frac{2}{9} e^{(9/4)(t-2))} - \frac{2}{9}. \]

After 300 generations, the solutions are obtained from the graph. The numerical solutions of MRDE are calculated and displayed in Table 5 using the ACP. Since this problem is having explicit solution, the ACP solution is equivalent to exact solution of the DAE. ACP solution curves are shown in Figures 13 and 14. In Figure 13, the values of \( t \) are noted in the \( X \) axis and the values
of \( k_{11} \) are placed in the \( Y \) axis. Similarly in Figure 14, the values of \( t \) and \( k_{12} \) and \( k_{13} \) are in \( X \) and \( Y \) axes, respectively. The parse trees for the solutions are given in Figures 15 and 16.

5. Conclusion

The singular optimal control for the stochastic linear singular system can be found by the ACP approach. To obtain the optimal control, the solution of MRDE is computed by solving DAE using a novel and nontraditional ACP approach. The obtained solution in this method is equivalent to the exact solution of the problem. Accuracy of the solution computed by the ACP approach to the problem is qualitatively better when it is compared with RK solution. A numerical example is given to illustrate the derived results.

Acknowledgements

The authors are very much thankful to the referees for the valuable comments and suggestions for improving this manuscript. The work of the authors was supported by the Department of Science and Technology, Government of India, New Delhi under SERC Project No. SR/S4/MS- 485/07 dt. 21.04.2008.

References


