Solving the Hamilton–Jacobi–Bellman equation using Adomian decomposition method

A. Fakhariana, M.T. Hamidi Beheshtia and A. Davari\textsuperscript{b}\textsuperscript{*}

\textsuperscript{a}Faculty of Engineering, Department of Electrical Engineering, Control Group, Tarbiat Modares University, Tehran, Iran; \textsuperscript{b}Department of Mathematics, University of Isfahan, Isfahan 81746–73441, Iran

\textit{(Received 26 August 2008; resubmitted 7 January 2009; revised 18 January 2009; accepted 27 January 2009)}

The aim of this research is to solve the Hamilton–Jacobi–Bellman equation (HJB) arising in nonlinear optimal problem using Adomian decomposition method. First Riccati equation with matrix variable coefficients, arising in linear optimal and robust control approach, is considered. By using the Adomian method, we consider an analytical approximation of the solution of nonlinear differential Riccati equation. An application in optimal control is presented. The solution in different order of approximations and different methods of approximation will be compared with respect to accuracy. Then the HJB equation, obtained in nonlinear optimal approach, is considered and an analytical approximation of the solution of it, using Adomian method, is presented.

\textbf{Keywords:} Adomian decomposition method; ADM; Riccati differential equation; optimal control; Hamilton–Jacobi–Bellman equation; HJB

\textbf{2000 AMS Subject Classification:} 49J15

1. Introduction

The Adomian decomposition method has been applied to a wide class of stochastic and deterministic problems involving algebraic, differential, integro-differential, differential-delay and partial differential equations and systems, [2,3,6,13]. This method leads to computable, accurate, approximate convergent solutions to linear and non-linear deterministic and stochastic operator equations [1,5,15]. The solutions can be verified at any stage of approximation. It is worth mentioning that several authors have treated many concepts related to the Adomian method such as the convergence concept and made comparisons with the other existing numerical techniques. The most significant works about convergence have been carried out by Cherruault [11] and Cherruault \textit{et al.} [12] based on the fixed-point theorems or substituting results in function series. Adomian decomposition method is an approximated approach for solving nonlinear differential equations by substitution of nonlinear parts of equation with Adomian polynomials and uses a step-by-step method for finding solutions [4]. This method is a powerful approach in nonlinear differential equations and its accuracy depends on the number of used partial solutions. Also, the solution of

*Corresponding author. Email: a_davari@sci.ui.ac.ir
this method generally has a fast convergence to the exact solution. In recent years, some modifications on this method have been presented [16,17]. Modification of the method is in quality of computation of Adomian polynomials. These modifications affect on convergence of the method. In [10,14], Adomian decomposition method used in nonlinear optimal control and non-quadratic cost functions optimal control. In [9] Adomian method used for optimal control of bilinear systems and application of this method for a type of control systems have been shown. Riccati equation arises in optimal and robust control theory, and it is a nonlinear, time-variant matrix coefficient equation. To solve this equation no analytical method exists. A method for solving this equation numerically is discretization of it in time domain and substitution of derivation operator with discrete approximation and finding solution in each iteration. But this method is very sensitive to sample time $\Delta T$ in discretization and may be unstable for some $\Delta T$. On the other hand, Hamilton–Jacobi–Bellman (HJB) equations were obtained in nonlinear optimal control and there are two approaches to solve them. First, we discretize the given system and using dynamic programming, we find optimal control signal. Second, we take advantage of given continuous system and come to HJB equation. Then we use approximation to solve this equation and to find optimal control signal. In this study, we consider linear, time-invariant system and apply optimal control to this system. Then using calculus of variations, we come to a Riccati equation. Then we apply Adomian method for solving this equation and compare solutions with different order of approximations. At next stage we consider a nonlinear system and apply nonlinear optimal control to this system. By using dynamic programming, we reach to HJB equation. Finally, we apply the Adomian method for solving this equation.

This article is organized as follows: Section 1, a brief description of optimal control is presented. Section 2, a brief description of nonlinear optimal control is discussed. Third section, Adomian decomposition method for solving differential equations will be described. Then, we apply Adomian method to Riccati equation and find the solution. Some examples are illustrated relating to their application and the solutions are compared. Also, we apply Adomian decomposition method to the HJB equation and find the solution. We come to the conclusion and suggestion, for future works. Our contribution in this paper is using modified Adomian method for solving both Riccati and the HJB equations with a prescribed accuracy.

2. Linear optimal control

In this section we have a brief description of optimal control. First we consider the following linear time-invariant system in state space realization [8]:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0,$$
$$y(t) = C_x x(t).$$

(1)

The system has no disturbance input. Suppose that $A$ is a $n \times n$ matrix, $x(t)$ a $n \times 1$ state space vector, $y(t)$ output vector and $u(t)$ is control signal. Our purpose is to control the above system and to find control signals subject to minimize the following cost function:

$$J(u, y) = \frac{1}{2} y^T(t_f) H_y y(t_f) + \frac{1}{2} \int_0^{t_f} (y^T(t) Q_y y(t) + u^T(t) R u(t)) dt.$$  

(2)

In this cost function, $Q_y$, $R$ and $H_y$ are positive definite and symmetric with appropriate dimensions. Now we want to rewrite $J(u, y)$ according to $x(t)$. Substitution of $y(t) = C_x x(t)$ in
Equation (2) results:

\[ J(u, x) = \frac{1}{2} x^T(t_f) H x(t_f) + \frac{1}{2} \int_0^{t_f} (x^T(t) Q x(t) + u^T(t) R u(t)) \, dt. \]  

(3)

That in Equation (3) we have

\[ H = C_y^T H_y C_y, \quad Q = C_y^T Q_y C_y. \]

That \( H \) and \( Q \) are positive semi-definite and symmetric matrices. Therefore, a constrained optimization problem is obtained with system dynamic equations’ constrains. By using Lagrange coefficients method and adding a constrained equation to cost function (3), we convert it to an unconstrained problem as follows:

\[ J_a(x, u, p(t)) = J(x, u) + \int_0^{t_f} p^T(t) (A x(t) + B_u u(t) - \dot{x}(t)) \, dt. \]

(4)

In Equation (4) \( p(t) \) is a Lagrange coefficient vector or a co-state. By using calculus of variations and simplifying the problem the following equations result:

\[
\begin{align*}
\dot{p}(t) &= -x^T(t) Q - p^T(t) A, \\
u(t) &= -R^{-1} B_u^T p(t), \\
\dot{x}(t) &= A x(t) + B_u u(t).
\end{align*}
\]

(5)

If we delete \( u(t) \) in Equation (5), then we have

\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{p}(t)
\end{bmatrix} = 
\begin{bmatrix}
A & -B_u R^{-1} B_u^T \\
-Q & -A
\end{bmatrix}
\begin{bmatrix}
x(t) \\
p(t)
\end{bmatrix} = Z
\begin{bmatrix}
x(t) \\
p(t)
\end{bmatrix},
\]

where

\[
x(0) = x_0, \quad p(t_f) = H x(t_f).
\]

(6)

The above system is the corresponding Hamiltonian system for Equations (1) and (3). Solution of Equation (6) in \( t = t_f \) with using state transient matrix will be

\[
\begin{bmatrix}
x(t_f) \\
p(t_f)
\end{bmatrix} = e^{Z(t_f-t)}
\begin{bmatrix}
\phi_{11}(t_f-t) & \phi_{12}(t_f-t) \\
\phi_{21}(t_f-t) & \phi_{22}(t_f-t)
\end{bmatrix}
\begin{bmatrix}
x(t) \\
p(t)
\end{bmatrix}.
\]

(7)

We have

\[
p(t) = [\phi_{22}(t_f-t) - H \phi_{12}(t_f-t)]^{-1} [H \phi_{11}(t_f-t) - \phi_{21}(t_f-t)] x(t)
\]

(8)

or

\[
p(t) = P(t) x(t).
\]

(9)

That

\[
P(t) = [\phi_{22}(t_f-t) - H \phi_{12}(t_f-t)]^{-1} [H \phi_{11}(t_f-t) - \phi_{21}(t_f-t)].
\]

(10)

If we derive from Equation (9) and substitute from Equation (6) then simplify it, we have

\[
-\dot{P}(t) = P(t) A + A^T P(t) + Q - P(t) B_u R^{-1} B_u^T P(t),
\]

\[
P(t_f) = H.
\]

(11)

This equation called ‘Riccati Equation’ and is a nonlinear time-variant differential equation. Because \( Q, R \geq 0 \) and are symmetric, global existence of solutions is guaranteed. It has two
solutions that positive semi-definite solution \((P(t) \geq 0)\) is desirable. Optimal control signal is obtained as follows:

\[
\begin{align*}
\mathbf{x}_{\text{opt}}(t) &= -R^{-1}B_u^T P(t) \mathbf{x}(t) = -K(t) \mathbf{x}(t), \\
K(t) &= R^{-1}B_u^T P(t).
\end{align*}
\]  
(12)

**Example 2.1**  Consider the following linear scalar time-invariant system:

\[
\dot{x}(t) = x(t) + u(t).
\]

We want to find \(u(t)\) such that minimize the following cost function:

\[
J = \frac{1}{2} 8x^2(10) + \frac{1}{2} \int_0^{10} (3x^2(t) + u^2(t)) \, dt.
\]

First we have \(A = 1, B_u = 1, t_f = 10, H = 8, Q = 3, R = 1\).

Organize Hamiltonian matrix \(Z\):

\[
Z = \begin{bmatrix} 1 & -1 \\ -3 & -1 \end{bmatrix}.
\]

State the transient matrix obtained as

\[
e^{Zt} = \begin{bmatrix}
\frac{3}{4}e^{2t} + \frac{1}{4}e^{-2t} & -\frac{1}{4}e^{2t} + \frac{1}{4}e^{-2t} \\
-\frac{3}{4}e^{2t} + \frac{3}{4}e^{-2t} & \frac{1}{4}e^{2t} + \frac{3}{4}e^{-2t}
\end{bmatrix}.
\]

Using Equations (10)–(12) will result in the following optimal gain:

\[
K(t) = \frac{27e^{2(10-t)} + 5e^{-2(10-t)}}{9e^{2(10-t)} - 5e^{-2(10-t)}}.
\]

It is clear that optimal feedback law is a nonlinear time-variant vector. If time horizon tends to infinity, optimal gain will be tending to 27/9. This is a steady state value of optimal gain.

### 3. Nonlinear optimal control

In this section we have a brief description of nonlinear optimal control. First consider the following nonlinear system in state space realization:

\[
\dot{x}(t) = a(x(t), u(t), t).
\]

(13)

In the above system, \(x(t)\) is a state vector; \(u(t)\) is a control signal. Our purpose is to control the system and to find the control signal such that it minimizes the following cost function:

\[
J = h(x(t_f), t_f) + \int_0^{t_f} g(x(\tau), u(\tau), \tau) \, d\tau.
\]

(14)
In this cost function, \( h \) and \( g \) are arbitrary convex functions and \( t_f \) is final time of system operation. Using dynamic programming approach, we introduce a new variable as

\[
j(x(t), t, u(\tau)) = h(x(t_f), t_f) + \int_0^{t_f} g(x(\tau), u(\tau), \tau) \, d\tau,
\]

\( t \leq \tau \leq t_f, \quad t_0 \leq t \leq t_f. \) \hfill (15)

Suppose that we have

\[
V(x(t), t) = J^*(x(t), t) = \min_{u(t)} \left\{ h(x(t_f), t_f) + \int_t^{t_f} g(x(\tau), u(\tau), \tau) \, d\tau \right\}.
\]

(16)

Therefore, we have

\[
V(x(t), t) = \min_{u(t)} \left\{ \int_t^{t+\Delta t} g(x(\tau), u(\tau), \tau) \, d\tau + V(x(t+\Delta t), t+\Delta t) \right\}.
\]

(17)

According to the principle of optimality, we have

\[
V(x(t), t) = \min_{u(t)} \left\{ \int_t^{t+\Delta t} g(x(\tau), u(\tau), \tau) \, d\tau + \frac{\partial V}{\partial t} \Delta t + \frac{\partial V}{\partial x} a(x, u, t) \Delta t + O(\Delta t) \right\}.
\]

(19)

If we suppose that \( \Delta t \) is small enough then \( \tau \to t \) and we have

\[
V(x(t), t) = \min_{u(t)} \left\{ g(x(t), u(t), t) + \frac{\partial V}{\partial x} a(x, u, t) \Delta t + O(\Delta t) \right\}.
\]

(20)

By dividing both sides of Equation (20) by \( \Delta t \), we have

\[
- \frac{\partial V}{\partial t} = \min_{u(t)} \left\{ g(x(t), u(t), t) + \frac{\partial V}{\partial x} a(x, u, t) \right\}.
\]

(21)

This nonlinear time-variant differential equation called ‘HJB equation’. We have the following boundary condition:

\[
J^*(x(t_f), t_f) = V(x(t_f), t_f) = h(x(t_f), t_f).
\]

(22)

By introducing the Hamiltonian function

\[
H(x, u, V_x, t) = g(x, u, t) + \frac{\partial V}{\partial x} a(x, u, t),
\]
we have
\[ H(x, u^*, V_x, t) = \min_{u(t)} H(x, u, V_x, t). \] (23)

Therefore by substitution of Hamiltonian function (23) in Equation (21), we have
\[ -\frac{\partial V}{\partial t} = H(x, u^*(x, V_x, t), V_x, t). \] (24)

**Example 3.1** Consider the following system:
\[ \dot{x}(t) = x(t) + u(t). \]

Suppose that we consider the following cost function for this system:
\[ J = \frac{1}{4} x^2(T) + \int_0^T \frac{1}{4} u^2(t) \, dt. \]

The corresponding Hamiltonian function will be
\[ H(x, u, V_x, t) = \frac{1}{4} u^2(t) + V(x, t)[x + u]. \]

For finding \( u^* \), we have
\[ \frac{\partial H}{\partial u} = \frac{1}{2} u + V(x, t) = 0. \]

Therefore we obtain
\[ u^* = -2 V_x(x, t). \]

Because \( \partial^2 H / \partial u^2 = 1/2 > 0 \), \( u^* \) is a minimum and acceptable. Now, by substitution \( u^* \) in HJB equation, we have the following equation:
\[ -V_t = V_x^2 + V_x x; \quad V(x(T), T) = \frac{1}{4} x^2(T). \]

Our goal in Section 4 is solving this equation using the Adomian decomposition method and then finding optimal control signal \( u^* \).

### 4. Adomian decomposition method

In this section we will have a brief description of Adomian method. Suppose that we have a nonlinear differential equation [4] in the form of
\[ Lu + Ru + Nu = g(x), \] (25)

where \( L \) is the highest order derivative assumed to be easily invertible, \( R \) the linear differential operator of less order than \( L \), \( N \) represents the nonlinear parts and \( g \) is the input part. Using inverse operator \( L^{-1} \) to both the sides of Equation (25), we obtain
\[ u = f(x) - L^{-1}(Ru) - L^{-1}(Nu) \] (26)

which will be produced after integration from \( g(x) \) and using given initial conditions. The Adomian decomposition method defines the solution \( u(x) \) by the series \( u(x) = \sum_{n=0}^{\infty} u_n \). In this regard,
the nonlinear operator \( N(u) \) is usually represented by an infinite series as follows:

\[
N(u) = \sum_{j=0}^{\infty} A_j.
\]

That components \( u_0, u_1, u_2, \ldots \) are usually determined recursively from the following equations:

\[
\begin{align*}
    u_0 &= f(x), \\
    u_{n+1} &= -L^{-1}(Ru_n) - L^{-1}(A_n).
\end{align*}
\]

(27)

The polynomials \( A_j \) are called Adomian polynomial and produced for all of nonlinearities so that \( A_0 \) depends only on \( u_0 \), \( A_1 \) depends on \( u_0 \) and \( u_1 \), and so on. \( A_j \) is obtained by

\[
A_j = \frac{1}{j!} \frac{d}{d\lambda} \left[ N \left( \sum_{i=0}^{\infty} u_i \lambda^i \right) \right]_{\lambda=0}, \quad j = 0, 1, 2, \ldots.
\]

(28)

And we have [7]

\[
\begin{align*}
    A_0 &= N(u_0), \\
    A_1 &= u_1 N'(u_0), \\
    A_2 &= u_2 N'(u_0) + \frac{u_1^2}{2!} N''(u_0), \\
    A_3 &= u_3 N'(u_0) + u_1 u_2 \frac{u_1^2}{2!} N''(u_0) + \frac{u_1^3}{3!} N'''(u_0).
\end{align*}
\]

(29)

And so on.

**Example 4.1** Consider the following nonlinear differential equation [4]:

\[
\frac{du}{dt} - u^2 = 0, \quad u(0) = 1,
\]

where \( L = d/dt, \) \( Nu = u^2, \) and the inverse operator \( L^{-1} \) defined as \( L^{-1} = \int_{0}^{t}(\cdot) \, dt. \) We use Adomian method for this problem. It is found that

\[
\begin{align*}
    u &= \sum_{n=0}^{\infty} u_n = u_0 + L^{-1} \sum_{n=0}^{\infty} A_n, \\
    u_0 &= u(0) = 1, \\
    u_1 &= L^{-1}(A_0) = L^{-1}(1) = t, \\
    u_2 &= L^{-1}(A_1) = L^{-1}(u_1 N'(u_0)) = L^{-1}(2t) = t^2
\end{align*}
\]

and \( u = \sum_{n=0}^{\infty} t^n = 1/(1 - t) \) is the exact solution.

5. **Brief description of method**

In this section we describe the application of Adomian method for solving Riccati and HJB equations. In Section 4.1, we describe the method in linear optimal control case and in Section 4.2, the nonlinear optimal control case will be considered.
5.1 Linear optimal control case

According to Equation (11), we have a nonlinear matrix equation with two solutions. The positive definite solution is acceptable. First, we introduce variables \( t = t_f - t \) and \( B = B_n R^{-1} B_n^T \). Then we have

\[
\dot{P}(\tau) = P(\tau)A + A^T P(\tau) + Q - P(\tau)BP(\tau),
\]

\[ P(0) = H. \]  

(30)

Let now \( L = \frac{d}{dt} \), so we have \( LP = \dot{P} \) and \( NP = PBP \), that \( N \) is the nonlinear operator. With substitution of mentioned parameters, Equation (30) becomes:

\[
\dot{P} = PA + A^T P + Q - NP. \]  

(31)

and in terms of inverse operator \( L^{-1} = \int_{0}^{\tau} [.] \, dx \):

\[
P = L^{-1}(PA + A^T P) + L^{-1}Q - L^{-1}NP. \]  

(32)

Now, we can apply the Adomian decomposition method mentioned in the previous section to Equation (32) and find the solution. Suppose that solution is \( P = \sum_{n=0}^{\infty} P_n \) and writing nonlinear part in the form of the below Adomian polynomials:

\[
NP = \sum_{n=0}^{\infty} A_n. \]  

(33)

Therefore Equation (32) becomes

\[
\sum_{n=0}^{\infty} P_n = H + L^{-1} \left( \sum_{n=0}^{\infty} P_n A + A^T \sum_{n=0}^{\infty} P_n \right) + L^{-1}Q - L^{-1} \sum_{n=0}^{\infty} A_n. \]  

(34)

Thus we can find the components of solution \( (P_n) \) as

\[
P_0 = H + L^{-1}Q,
\]

\[
P_1 = L^{-1}(P_0 A + A^T P_0) - L^{-1}A_0,
\]

\[
P_2 = L^{-1}(P_1 A + A^T P_1) - L^{-1}A_1,
\]

\[
\vdots
\]

\[
P_n = L^{-1}(P_{n-1} A + A^T P_{n-1}) - L^{-1}A_{n-1}. \]  

(35)

Now, we shall produce \( A_n \) polynomials for completion of the method. There is a step-by-step method for finding \( A_n \) as follows:

\[
A_0 = P_0 B P_0,
\]

\[
A_1 = P_1 B P_0 + P_0 B P_1,
\]

\[
A_2 = P_2 B P_0 + P_1 B P_1 + P_0 B P_2.
\]

\[
\vdots
\]

\[
A_n = \sum_{i=0}^{n} P_i B P_{n-i}. \]  

(36)
Thus with substitution of Equation (36) in Equation (35) we have the following step-by-step equations:

\[ P_0 = H + L^{-1}Q, \]
\[ P_1 = L^{-1}(P_0A + A^TP_0) - L^{-1}(P_0BP_0), \]
\[ P_2 = L^{-1}(P_1A + A^TP_1) - L^{-1}(P_1BP_0 + P_0BP_1), \]
\[ \vdots \]
\[ P_{n+1} = L^{-1}(P_nA + A^TP_n) - L^{-1}\left\{ \sum_{i=0}^{n} P_iBP_{n-i} \right\}. \]  

(37)

Therefore with computing partial solution \( P_n \) and calculation of the sum of them, we can find approximated response with a desirable accuracy. It is clear that when we use more terms of partial solution, the obtained response is more accurate. Now, we use the mentioned algorithm for a typical example.

**Example 5.1** Consider the following differential equation:

\[ \dot{P}(t) = -P^2 + 1, \]
\[ P(0) = 0. \]

In this equation we have \( A = 0, B = -1, Q = 1, H = 0. \)

Exact solution of this equation is \( P(t) = (e^{2t} - 1)/(e^{2t} + 1). \) If we use the mentioned method introduced in this section, we have

\[ P_0 = H + L^{-1}Q = t \]
\[ P_1 = L^{-1}(P_0A + A^TP_0) - L^{-1}(P_0BP_0) = L^{-1}(-t^2) = -\frac{1}{3}t^3 \]
\[ P_2 = L^{-1}(P_1A + A^TP_1) - L^{-1}(P_1BP_0 + P_0BP_1) = \frac{2}{15}t^5 \]

and so on. Therefore, we consider \( \phi_n = \sum_{i=0}^{n-1} P_i \) as partial solution of Riccati equation. So

\[ \phi_1 = t \]
\[ \phi_2 = t - \frac{1}{3}t^3 \]
\[ \phi_3 = t - \frac{1}{3}t^3 + \frac{2}{15}t^5. \]

We plot exact and approximated solutions of the Riccati equation in Figure 1. Also the approximated solution of this equation with the discretization method is plotted in this figure. From Figure 1, it is clear that by increasing the number of Adomian partial solutions, accuracy of solution increases. Now we calculate error of solutions in \( t = 0.8 \) s for comparing them. We have absolute error in different cases as follows:

- Case 1: 0.1360.
- Case 2: 0.0347.
- Case 3: 0.009.
- Case 4 (Discretization method): 0.0443.
Figure 1. Exact and approximated solutions of Riccati equation.

This confirms that by increasing the partial sum of the solution, error reduces. Also it is considered that the Adomian method is better than the discretization method because, error of approximation in cases 2 and 3 is less than error of discretization method. Now we consider another example for this purpose.

**Example 5.2** Consider the following Ricatti equation:

\[
\dot{P}(t) = 2P(t) - P^2(t) + 1,
\]

\[
P(0) = 0.
\]

In this case we have \( A = 0, B = -1, Q = 1, H = 0. \)

The exact solution of this equation is

\[
P(t) = 1 + \sqrt{2} \tanh \left( \sqrt{2} t + \frac{1}{2} \log \left( \frac{\sqrt{2} - 1}{\sqrt{2} + 1} \right) \right).
\]

If we use Adomian decomposition method for this equation, we have

\[
P_0 = H + L^{-1} Q = t,
\]

\[
P_1 = L^{-1}(P_0A + A^TP_0) - L^{-1}(P_0BP_0) = L^{-1}(2t) + L^{-1}(-t^2) = t^2 - \frac{1}{3} t^3,
\]

\[
P_2 = L^{-1}(P_1A + A^TP_1) - L^{-1}(P_1BP_0 + P_0BP_1) = \frac{2}{3} t^3 - \frac{2}{3} t^4 + \frac{2}{15} t^5.
\]

Therefore we consider \( \phi_n = \sum_{i=0}^{n-1} P_i \) as a partial solution of Riccati equation. So

\[
\phi_1 = t,
\]

\[
\phi_2 = t + t^2 - \frac{1}{3} t^3,
\]

\[
\phi_3 = t + t^2 + \frac{1}{3} t^3 - \frac{2}{3} t^4 + \frac{2}{15} t^5.
\]
We plot exact and approximated solutions of the Riccati equation in Figure 2. Also the approximated solution of this equation with discretization method is plotted in this figure. From Figure 2 it is clear that the accuracy of the solution increases with using more number of Adomian partial solutions in our response. Now, we calculate the absolute error of solution for comparing them. We have absolute error in different cases as follows:

- Case 1: 0.5464.
- Case 2: 0.0771.
- Case 3: 0.0349.
- Case 4: (Discretization method): 0.1402.

This confirms that by increasing partial sum of solution, error reduces. Also error in cases 2 and 3 is less than relative error in the discretization method. Therefore the Adomian method has a better accuracy than discretization method. Therefore, we used introduced method for solving the Riccati equation and considered the application of it. Also the effect of the number of partial solutions is considered in accuracy of the solution.

**Example 5.3** Consider the following state space system:

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ 3 & 2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.
\]

Suppose that relevant weight matrices in Riccati equation are

\[ H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q = R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \]
Therefore using the method proposed in Section 4, we have the following results:

\[ P_0 = H + L^{-1}Q = \begin{bmatrix} t & 0 \\ 0 & t \end{bmatrix}, \]
\[ P_1 = L^{-1}(P_0A + A^TP_0) - L^{-1}(P_0BP_0) = \begin{bmatrix} 0 & 2t^2 \\ 2t^2 & 0 \end{bmatrix}, \]
\[ P_2 = L^{-1}(P_1A + A^TP_1) - L^{-1}(P_1BP_0 + P_0BP_1) = \begin{bmatrix} 4t^3 & 10t^3 + 3t^4 \\ 10t^3 + 3t^4 & 4t^3 + 4t^4 + 2t^5 \end{bmatrix}, \]

and so on. Therefore we consider \( \phi_n = \sum_{i=0}^{n-1} P_i \) as a partial solution of Riccati equation.

\[ \phi_1 = P_0 = \begin{bmatrix} t & 0 \\ 0 & t \end{bmatrix}, \]
\[ \phi_2 = P_0 + P_1 = \begin{bmatrix} t + 2t^2 \\ 2t^2 & t \end{bmatrix}, \]
\[ \phi_3 = P_0 + P_1 + P_2 = \begin{bmatrix} t + 4t^3 & 2t^2 + 10t^3 + 3t^4 \\ 2t^2 + 10t^3 + 3t^4 & t + 2t^2 + 4t^3 + 4t^4 + 2t^5 \end{bmatrix}. \]

We plot exact and approximated solutions of Riccati equation \( P_{22} \) in Figure 3. Also eigenvalues of exact and approximated solutions are plotted in Figures 4 and 5. From Figure 3, it is clear that the accuracy of the solution increased by adding more terms of Adomian’s polynomials. Also from Figures 4 and 5, it is considered that eigenvalues of approximated solution of Riccati equation tend to eigenvalues of exact solution of this equation by increasing terms of Adomian’s polynomials in solution. The values of exact and approximated solutions of Riccati equation in \( t = 0.2 \) s are

\[ P_{|t=0.2\,s} = \begin{bmatrix} 0.2484 & 0.1230 \\ 0.1230 & 0.3326 \end{bmatrix}, \]
\[ \phi_{1\,|t=0.2\,s} = \begin{bmatrix} 0.2000 & 0.0800 \\ 0.0800 & 0.2827 \end{bmatrix}, \]
\[ \phi_{2\,|t=0.2\,s} = \begin{bmatrix} 0.2320 & 0.1079 \\ 0.1079 & 0.3168 \end{bmatrix}. \]

And values of eigenvalues of exact and approximated solutions of Riccati equation in \( t = 0.2 \) s are

\[ \lambda(P(t))_{|t=0.2\,s} = \begin{bmatrix} 0.1609 & 0.4209 \end{bmatrix}, \]
\[ \lambda\phi_1((t))_{|t=0.2\,s} = \begin{bmatrix} 0.1513 & 0.3314 \end{bmatrix}, \]
\[ \lambda\phi_2((t))_{|t=0.2\,s} = \begin{bmatrix} 0.1585 & 0.3903 \end{bmatrix}. \]

This is clear that by increasing the order of approximation, we have a more accurate solution.
5.2 Nonlinear optimal control case

According to Equation (24) we have a nonlinear, time variant differential equation in regard to $V(x, t)$. First, we introduce variable $\tau = t_f - t$. Therefore, we have

$$\frac{\partial V}{\partial \tau} = H(x, u^*(x, V_x, \tau), V_x, \tau)$$

(38)
and consequently, initial condition will be as follows:

\[ J^*(x(0), 0) = V(x(0), 0) = h(x(0), 0). \]  

(39)

Let now \( L_\tau = \partial/\partial \tau \), so we have

\[ L_\tau V = H(x, u^*(x, V_x, \tau), V_x, \tau). \]  

(40)

If we find \( u^* \) from Equation (23), we can suppose that

\[ u^*(x, V_x, \tau) = f(V_x, x, \tau). \]  

(41)

By substituting Equation (41) into Equation (40), we have

\[ L_\tau V = H(x, f(V_x, x, \tau), V_x, \tau). \]  

(42)

Therefore, we can rewrite Equation (42) as follows:

\[ L_\tau V = R(V_x) + N(V_x). \]  

(43)

That \( R(V_x) \) is linear part and \( N(V_x) \) is nonlinear part of \( H(x, f(V_x, x, \tau), V_x, \tau) \). We apply inverse operator \( L_\tau^{-1} = \int_0^\tau [.] \, d\tau \) to both sides of Equation (43). Therefore, we have

\[ L_\tau^{-1} L_\tau V = L_\tau^{-1} R(V_x) + L_\tau^{-1} N(V_x). \]  

(44)

Then, we consider \( V(x, \tau) \) as follows:

\[ V(x, \tau) = \sum_{n=0}^{\infty} V_n. \]  

(45)

Also, the nonlinear term of Equation (44) will be equated to \( \sum_{n=0}^{\infty} A_n \), that \( A_n \) are Adomian polynomial and shall be computed according to nonlinear part format. By substitution of mentioned...
terms in Equation (44), we have
\[ \sum_{n=0}^{\infty} V_n = V_0 + L_{\tau}^{-1} L_x R \left( \sum_{n=0}^{\infty} V_n \right) + L_{\tau}^{-1} \left( \sum_{n=0}^{\infty} A_n \right). \] (46)

In Equation (46), \( V_0 \) identified as follows:
\[ V_0 = h(x(0), 0). \] (47)

By using Equation (28), we can compute \( V_i \) as follows:
\[ V_1 = L_{\tau}^{-1} L_x R(V_0) + L_{\tau}^{-1} A_0, \]
\[ V_2 = L_{\tau}^{-1} L_x R(V_1) + L_{\tau}^{-1} A_1, \]
\[ V_3 = L_{\tau}^{-1} L_x R(V_2) + L_{\tau}^{-1} A_2, \]
\[ \vdots \]
\[ V_{n+1} = L_{\tau}^{-1} L_x R(V_n) + L_{\tau}^{-1} A_n. \] (48)

And Adomian polynomial \( A_n \) can be computed as follows:
\[ A_0 = N(V_{0x}), \]
\[ A_1 = V_{1x} N'(V_{0x}), \]
\[ A_2 = V_{2x} N'(V_{0x}) + \frac{V_{1x}^2}{2!} N''(V_{0x}), \]
\[ A_3 = V_{3x} N'(V_{0x}) + V_{1x} V_{2x} N''(V_{0x}) + \frac{V_{1x}^3}{3!} N'''(V_{0x}) \] (49)
and so on. Therefore, \( V_n(x, \tau) = \sum_{i=0}^{n} V_i \) is partial solution of HJB equation and we can improve accuracy of solution by increasing the number of partial solution.

**Example 5.4** Suppose that we have the following HJB equation:
\[ V_t = x^2 - \frac{1}{4} (V_x)^2, \]
\[ V(x, 0) = 0. \]

If we use the mentioned method for this equation, we have
\[ L_t V = x^2 - \frac{1}{4} (V_x)^2. \]

And then, we have:
\[ L_{\tau}^{-1} L_t V = L_{\tau}^{-1} x^2 - \frac{1}{4} L_{\tau}^{-1} (V_x)^2. \]

Since the left side is \( V(x, t) - V(x, 0) = V(x, t) \), we have
\[ \sum_{n=0}^{\infty} V_n = V_0 - \frac{1}{4} L_{\tau}^{-1} \sum_{n=0}^{\infty} A_n. \]
We let $V_0(x, t) = \sum_{i=0}^{n} V_i$, that $V_0 = L_t^{-1}x^2 = x^2t$. It is clear that nonlinear part is $(V_x)^2$. Therefore, we can compute Adomian polynomials using Equation (49) as follows:

$$A_0 = V,$$
$$A_1 = 2V_0V_1,$$
$$A_2 = V_1^2 + 2V_0V_2.$$ 

And so on. Consequently, we have

$$V_1 = -\frac{1}{4}L_t^{-1}(V_0x)^2 = -\frac{1}{4}L_t^{-1}(4x^2t^2) = -\frac{1}{3}x^2t^3,$$
$$V_2 = -\frac{1}{4}L_t^{-1}(2V_0V_1x) = \frac{2}{15}x^3t^5,$$
$$V_3 = -\frac{1}{4}L_t^{-1}(V_1^2 + 2V_0V_2x).$$

So we can compute closed form for solution as follows:

$$V(x, t) = x^2 \left( t - \frac{1}{3}t^3 + \frac{2}{15}t^5 - \cdots \right)$$
$$V(x, t) = x^2 \tanh t, \quad |t| < \frac{\pi}{2}.$$ 

And therefore we calculated the solution of HJB equation using the mentioned method.

6. Conclusion

We introduced a new method for solving Riccati and HJB equations using the Adomian decomposition method. It was considered that increase in the number of partial solutions causes decrease in error of approximated solution. For future works, we can use this method with some modification for solving Hamilton–Jacobi–Isaacs equations (HJI). Also, we can use this method for analysing singular perturbation systems.

References

International Journal of Computer Mathematics