Approximate Closed-Form Solutions to Finite-Horizon Optimal Control of Nonlinear Systems

Ali Heydari, Student Member, IEEE, and S. N. Balakrishnan, Member, IEEE

Abstract— The Hamilton-Jacobi-Bellman partial differential equation, which is needed to be solved for finite-horizon optimal control of nonlinear systems, is reduced to a state-dependent differential Riccati equation subject to a final condition through some approximations. Afterward, a method, called Finite-SDRE, is developed for finite-horizon near-optimal control synthesis. This technique allows for easier online implementation and its global stability is proved. Finally an approximate solution to the differential equation is given. Performance of the proposed controller in representative numerical examples demonstrates its excellent potential for use in nonlinear finite-horizon problems.

I. INTRODUCTION

FINITE-HORIZON optimal control of nonlinear systems is a challenging problem in the control field due to the added complexity of time-dependency of the solution compared to the infinite-horizon optimal control problems. A nonlinear partial differential equation called Hamilton-Jacobi-Bellman (HJB) is needed to be solved for the time-dependent solution and numerical methods give only an open loop solution depending on the pre-specified initial condition (IC) and time-to-go [1].

The available methods for this purpose in the literature can be classified to classical and intelligent methods. In the classical methods, one approach is calculating the open loop solution through some numerical method like a shooting method and then using techniques like Model Predictive Control (MPC) or Neighboring Optimal Control (NOC) for closing the control loop as done in [2] and [3] for MPC and NOC, respectively. The drawback is mainly the dependency of the solution to the pre-specified IC (or a neighborhood of that IC) and time-to-go. These methods are actually based on some pre-calculated states trajectories, and try to minimize the deviations from those trajectories.

A new method called approximate sequence of Riccati equation (ASRE) is developed in [4] which gives closed form solution to the problem, but, again, for some pre-specified IC and time-to-go. This method is based on the calculation of a sequence of Riccati equations until they converge, and then using the converged one for online control.

Another approach, which unlike usage of MPC and NOC in the cited works and ASRE, calculates the optimal solution online, is the Taylor series based methods, [5], which is based on finding a series based solution to the HJB equation. The drawback of this method is the limited domain of convergence of the series, like every other series based solution to the optimal control problems.

As for the intelligent methods for solving this problem, some researchers have used neural networks for finite-horizon optimal control of nonlinear problems, [6]-[9]. These methods are based on offline training of the weights and online usage of the neural network for calculation of the control. It is shown that they provide optimal solution for every initial condition as long as the resulted trajectory lies on the domain for which the neural network is trained.

Being motivated by the state-dependent Riccati equation (SDRE) method [10] for infinite-horizon optimal control of nonlinear systems, which gives a closed form control through the online solution of some state-dependent algebraic Riccati equation, a state-dependent differential Riccati equation (state dependent DRE) is introduced in this work and claimed to give a globally stabilizing approximate solution to the optimal control problem.

The approximations are done in skipping a term in transformation of the HJB equation to the state dependent DRE and in the control equation in which the partial derivative of the DRE solution with respect to the states are skipped for the sake of simplicity and avoiding mathematical intractability. Having done the approximations, the global stability of the resulted controller is proved.

The problem in solving the DRE is due to the fact that the states trajectory needs to be known ahead of time, which is impossible for an online implementation. To alleviate this problem, an approximate analytical approach is developed which is based on a change of variable that converts the DRE to a linear Lyapunov equation, [12] & [13], and evaluating the coefficients of the resulted equation based on the current state values at each time step and freezing them from current time to the final time. Then, the Lyapunov equation is solved in a closed form at each step during online implementation.

The rest of this paper is organized as follows; in section II the main results are given, and in section III a nonlinear system is simulated and analyzed using the developed controller. Finally the conclusions are given in section IV.
II. MAIN RESULTS

A. Introducing Finite-SDRE

The state equation of a nonlinear input-affine system is assumed to be of the form
\[ \dot{x}(t) = f(x) + B(x)u(t) \]  
where \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^m \) are the state and control vectors, respectively, and \( n \) and \( m \) are the order of the system and number of inputs, respectively. \( f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n \) and \( B(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{nxm} \) represent the dynamics of the system and \( t \) denotes the (continuous) time. For simplicity in the notation, the argument \( t \) is omitted in some places in the paper.

In finite-horizon optimal control, one in interested in finding the optimal control history \( u^*(t) \) for \( t_0 \leq t < t_f \) which minimizes a cost function \( J \), like the following
\[ J = \frac{1}{2} x^T(t_f)Sx(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^TQx + u^TRu)dt \]  
where \( S \) and \( Q \) are assumed to be positive definite or positive semi-definite matrices, and \( R \) is a positive definite matrix.

The approach developed here for solving finite-horizon optimal nonlinear problems is inspired by the State Dependent Riccati Equation (SDRE) method developed in [10]. In SDRE, one rewrites the state space equation (1) in the below linear form
\[ \dot{x}(t) = A(x)x(t) + B(x)u(t) \]  
where \( A(x)x(t) = f(x) \), and solves the following algebraic Riccati equation (ARE) with the state dependent coefficients at each time step during the real-time implementation to end up with \( P(x) \).

\[ P(x)A(x) + A^T(x)P(x) + Q - P(x)B(x)R^{-1}B(x)^T P(x) = 0 \]  

The solution, \( P(x) \), is then used for calculating the control through the below equation
\[ u(t) = -R^{-1}B(x)^TP(x)x(t) \]  

Interested readers are referred to [10] and for more information about forming the linear like form state equation (3) from (1) one may refer to [11].

Note that the SDRE is an approximate solution to the infinite-horizon optimal control problem, however, the problem to be solved here, belongs to the finite-horizon category. In the finite-horizon case, the feedback solution is time-dependent and a differential equation, rather than an algebraic one, needs to be solved to calculate the control. Following the method of solving a differential Riccati equation (DRE) for finite-horizon optimal control of linear systems, the below state dependent DRE is suggested to be solved for the approximate finite-horizon optimal solution.

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

where \( \dot{P}(x,t) \) denotes the total time-derivative of matrix \( P(x,t) \). The final condition is given by
\[ P(x,t_f) = S \]  

Note that in order for (7) to be satisfied at the boundary, the typical SDRE is not applicable here. The control is calculated in a feedback form as
\[ u(x,t) = -R^{-1}B(x)^TP(x,t)x(t) \]  

The following assumption is made for guaranteeing an stabilizing positive definite solution to DRE (6).

**Assumption 1:** Pairs \( \{A(x), B(x)\} \) and \( \{A(x), Q^{1/2}\} \) are pointwise stabilizable and observable for all \( x \in \mathbb{R}^n \), respectively, where the Cholesky decomposition of \( Q \) is denoted by \( Q^{1/2} \).

The closed loop controller given through Eq. (8) resulted from solving DRE (6) for the positive definite matrix \( P(x,t) \) is introduced as the Finite-SDRE method.

B. Supporting Theorems

As the first step, we show the relation between the proposed method and the exact optimal solution to the problem, i.e. the solution to the corresponding HJB equation. This is done through the following Theorem.

**Theorem 1:** Denoting the \( i \)th element of the state vector \( x \) by \( x_i \), \( 1 \leq i \leq n \), the partial derivative of \( P \) with respect to \( x_i \), i.e. \( P_{x_i} \), \( i \)th element of \( P(x,t) \) is calculated, the optimal control is given by
\[ u(x(t),t) = -R^{-1}B(x)^TP(x(t),t)x(t) \]  

\[ u(x(t),t) = -R^{-1}B(x)^TP(x(t),t)x(t) \]  

The solution to the optimal control of nonlinear system (3) subject to the cost function (2) is given by
\[ u^*(x(t),t) = -R^{-1}B(x)^T(P(x(t),t)x(t) + \Omega) \]  

where \( P(x(t),t) \) is the positive definite solution to the equation
\[ P(x(t),t)A(x) + A^T(x)P(x(t),t) + Q - P(x(t),t)B(x)R^{-1}B(x)^T P(x(t),t) + \Omega = -\dot{P}(x(t),t) \]  
with boundary condition (7).

**Proof:** It is well-known that the desired finite-horizon optimal control is given by the partial differential HJB equation given below, [1]
\[ -J^*_t = J^*_x A(x)x + \frac{1}{2} x^TQx - \frac{1}{2} J^*_x B(x)R^{-1}B(x)^T J^*_x \]  

with the final condition
\[ J^*(x(t_f),t_f) = \frac{1}{2} x^TSx \]  

where \( J^*(x(t),t) \) denotes the optimal cost-to-go and subscript \( t \) and \( x \) denote the corresponding partial derivatives of \( J^* \). Once \( J^* \) is calculated, the optimal control is given by
\[ u^*(x(t),t) = -R^{-1}B(x)^T J^*_x \]  

Since \( J^*(x(t),t) \) is a positive definite parameter, one can write it in the form of
\[ J^*(x(t),t) = \frac{1}{2} x^TP(x,t)x \]  

for some symmetric positive definite matrix \( P(x,t) \). For the sake of simplicity, in some places notation \( P \) is used instead of \( P(x,t) \). Having (16), leads to
\[ J^*_t = \frac{1}{2} x^T P_{x}x \]  

\[ J^*_x = P_{x} + \Omega \]  

Using (17) and (18) in the (13), leads to
\[ \frac{1}{2}(P_{x} + \Omega)B(x)R^{-1}B(x)^T(P_{x} + \Omega) \]  

which can be re-arranged to get
\[-\frac{1}{2}x^TP_x x = x^T \left( PA(x) + \frac{1}{2} Q - \frac{1}{2} PB(x) R^{-1} B(x)^T (x) P \right) x + \Pi^T \left( A(x)x - B(x) R^{-1} B(x)^T \left( P x + \frac{1}{2} \Pi \right) \right) \]  

(20)

Using (11) in (3) gives

\[ \dot{x} = A(x)x - B(x) R^{-1} B(x)^T (P x + \Pi) \]  

(21)

Using (21) in (20) leads to

\[-\frac{1}{2} x^T P_x x = x^T \left( PA(x) + \frac{1}{2} Q - \frac{1}{2} PB(x) R^{-1} B(x)^T (x) P \right) x + \Pi^T \dot{x} + \frac{1}{2} \Pi^T B(x) R^{-1} B(x)^T \Pi \]  

(22)

Since \( \dot{x}_i \) is a scalar, noting (9), one has

\[ \Pi^T \dot{x} = \frac{1}{2} \sum_{i=1}^{n} x_i \dot{x}_i = \frac{1}{2} x^T \left( \sum_{i=1}^{n} P_{x_i} \dot{x}_i \right) \]  

(23)

Using (23) in (22) and bringing it to the left hand side of the equation results in

\[-\frac{1}{2} x^T (P_t + \sum_{i=1}^{n} P_{x_i} \dot{x}_i) x = x^T \left( PA(x) + \frac{1}{2} Q - \frac{1}{2} PB(x) R^{-1} B(x)^T (x) P \right) x + \frac{1}{2} \Pi^T B(x) R^{-1} B(x)^T \Pi \]  

(24)

On the other hand, denoting the total derivative of \( P \) by \( \dot{P} \) one has

\[ \Pi^T \dot{B}(x) R^{-1} B(x)^T \Pi = \sum_{i=1}^{n} \sum_{j=1}^{n} \Pi_i (B(x) R^{-1} B(x)^T)^{ij} \Pi_j \]  

(26)

and since \( \Pi_i = \frac{1}{2} x^T P_{x_i} x \) one has

\[ \Pi^T B(x) R^{-1} B(x)^T \Pi = \frac{1}{4} x^T \left( \sum_{i=1}^{n} \sum_{j=1}^{n} P_{x_i} x (B(x) R^{-1} B(x)^T)^{ij} x^T P_{x_j} \right) x \]  

(27)

Using (25) and (27) in (24) leads to

\[-\frac{1}{2} x^T \dot{P} x = x^T (PA(x) + \frac{1}{2} Q - \frac{1}{2} PB(x) R^{-1} B(x)^T (x) P) \] 

\[ + \frac{1}{2} \Omega x \]  

(28)

and finally, for (28) to be valid for every \( x \in \mathbb{R}^n \), one needs

\[ -\dot{P} = PA(x) + A^T(x) P + Q - PB(x) R^{-1} B(x)^T (x) P + \Omega \]  

(29)

The boundary condition (14), using (16) is equivalent to

\[ P(x, t_f) = \mathbf{S} \]  

(30)

This proves that solving equation (12) with the final condition (7) solves the HJB equation (13) and gives the optimal solution to the nonlinear finite-horizon problem. □

Having proved the optimality of (11) resulted from (12), we proceed to perform some approximations by neglecting terms \( \Omega \) in (12) and \( \Pi \) in (11), which leads to the control given by (8) resulted from solving DRE (6). Using this approximation, the control will no longer be optimal, but, through the next theorem, we prove that it will still be globally stable.

**Definition:** A control \( \hat{u}(t) \in \mathbb{R}^m \), \( t_0 \leq t < t_f \), is called globally stabilizing for the nonlinear system of \( \dot{x} = F(x(t), u(t)) \) where \( F: \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \) denotes the dynamics of the system, if there is a positive definite Lyapunov function \( V \) which is radially unbounded and its total time derivative along the states trajectory of the closed loop system \( \dot{x} = F(\hat{x}(t), \hat{u}(t)) \) is negative definite for every \( x \in \mathbb{R}^n \) and \( t_0 \leq t < t_f \). Note that in this definition, if \( t_f \to \infty \), one has \( x \to 0 \) and the stability will be called asymptotic stability.

**Theorem 2:** The closed form control given through Finite-SDRE, i.e. Eq. (8) resulted from solving DRE (6) for the positive definite matrix \( P(x, t) \), makes the nonlinear system (3) a globally stable system.

**Proof:** Selecting the Lyapunov function \( V(x, t) = x^T P(x, t) x \), where \( P(x, t) \) is the symmetric positive definite solution to (6), and taking the total derivative of \( V(x, t) \) leads to

\[ \dot{V} = x^T \dot{P}(x, t) x + x^T P(x, t) \dot{x} + x^T \dot{P}(x, t) x \]  

(31)

Using controller (8) in (3) gives the closed loop dynamic of

\[ \dot{x} = (A(x) - B(x) R^{-1} B(x)^T P(x, t)) x \]  

(32)

Using above dynamics in (31) results in

\[ \dot{V} = x^T (P(x, t) A(x) + A^T(x) P(x, t)) x - 2P(x, t) B(x) R^{-1} B(x)^T (x) P(x, t) \]  

(33)

Since \( P(x, t) \) resulted from the Finite-SDRE satisfies DRE (6), using (6) in (33) leads to

\[ \dot{V} = x^T (P(x, t) B(x) R^{-1} B(x)^T P(x, t) - Q) x \]  

(34)

which leads to \( \dot{V} \leq 0 \). Noting that \( R > 0 \) and \( A(x), Q^{1/2} \) being observable, the only condition which \( V = 0 \) can happen is \( x = 0 \), hence, \( V \) is actually negative definite. Moreover, due to \( V \to \infty \) as \( x \to \infty \), the radial unboundedness condition is also satisfied, hence, the Finite-SDRE method is globally stable.

**Remark 1:** As a comment to Theorem 2, the reason for which this simple Lyapunov function cannot be used for proof of stability of SDRE method, is the existence of the third term in the right hand side of (31), otherwise, using SDRE, one has \( x^T P(x) x + x^T P(x) x < 0 \).

**C. Approximate Solution**

There is a problem in solving (6) with the final condition (7), that is, since the states in the future are not known ahead of time, one cannot calculate the state dependent coefficients to integrate (6) backward from \( t_f \) to \( t \) and end up with the unknown at time \( t \), i.e. \( P(x, t) \).

To remedy the problem, an approximate analytical approach is developed in this paper. The basis for this approach uses the closed form solution obtained by using a transformation with the solution to the ARE that converts the original nonlinear Ricatti equation to a Lyapunov equation, [12] & [13], then assuming constant coefficients from \( t \) to \( t_f \), evaluated based on the current state value at each time step, the Lyapunov equation can be solved in closed form.

As developed in Ref. [12] and [13] for linear system, in order to solve the DRE (6), one can follow these steps; Solve the ARE given below

\[ P_{s*(x)} A(x) + A^T(x) P_{s*(x)} - P_{s*(x)} B(x) R^{-1} B(x)^T (x) P_{s*(x)} + Q = 0 \]  

(35)

Subtracting (35) from (6) results in

\[ P(x, t) - P_{s*(x)} A(x) + A(x)^T P(x, t) - P_{s*(x)} \]

\[ - P(x, t) B(x) R^{-1} B(x)^T P(x, t) + P_{s*(x)} B(x) R^{-1} B(x)^T P_{s*(x)} = -\dot{P}(x, t) \]  

(36)
Using the change of variable of $K(x,t) \equiv (P(x,t) - P_{ss}(x))^{-1}$ in (36) leads to
\[ K(x,t) = A_{ct}(x)K(x,t) + K(x,t)A_{ct}(x)^T - B(x)R^{-1}B^T(x) \]
where $A_{ct}(x) \equiv A(x) - B(x)R^{-1}B^T(x)P_{ss}(x)$. The final condition of above equation is
\[ K(x,t_f) = (S - P_{ss}(x))^{-1} \]
Interestingly (37) is no longer a nonlinear equation and closed form solution for that can be found.

Here, the approximation takes place by evaluating the coefficients $A_{ct}(x)$ and $B(x)$ at current state value at each step and assuming that they are fixed for the rest of the horizon. To do this, define $A_{ct}$ as $A_{ct}(x)$ evaluated at current state vector, and $B$ as $B(x)$ evaluated at current state vector, i.e.,
\[ A_{ct} \equiv A_{ct}(x) \big|_{x=x(t)} \]
(39)
\[ B \equiv B(x) \big|_{x=x(t)} \]
(40)

Using $A_{ct}$ and $B$ in (37) instead of $A_{ct}(x)$ and $B(x)$, respectively, the result will be a constant coefficient differential Lyapunov equation whose solution, as shown in [14], is given by,
\[ K(x,t) = e^{A_{ct}(t-t_f)}(K(x,t_f) - D)e^{A_{ct}^T(t-t_f)} + D \]
(41)
where $D$ is the solution to the algebraic Lyapunov equation
\[ A_{ct}D + DA_{ct}^T = BR^{-1}B^T \]
(42)

In summary, these steps need to be done online at each time step: Solve ARE (35) and calculate (38), then (42) needs to be solved whose result will then be used in (41) to calculate $K(x,t)$. Having $K(x,t)$ one has
\[ P(x,t) = K^{-1}(x,t) + P_{ss}(x) \]
(43)

Finally, the control can then be calculated in the feedback form as
\[ u(x,t) = -R^{-1}B^T(x)P(x,t)x(t) \]
(44)

The computational effort needed to be done in real-time at each time step to end up with the closed loop solution mainly includes solving an ARE, calculating a matrix exponential, and performing two matrix inversions.

Note that, the assumed constant values for the Lyapunov equation’s coefficients are subject to evaluation online in each time step based on the current value of the state vector. The numerical analysis shows that having small time-steps, the approximation gives excellent results in the sense of bringing the desired states close to the origin in the fixed horizon.

It is known that, at least for linear systems, once $t_0 \ll t_f$ the solution to DRE (6) converges to that of the ARE (35), leading to singularity in matrix $(P(x,t) - P_{ss}(x))$, whose inverse is defined as the new variable $K(x,t)$ and used to convert (36) to (37). To avoid this singularity, for linear systems, Ref. [13] suggested to calculate the negative definite solution of ARE (35) instead of the positive definite one and use it in the rest of the process. Note that in this case $(P(x,t) - P_{ss}(x))$ is guaranteed to be positive definite, hence, its inverse always exists. This approach works for the nonlinear case as well. For calculation of negative definite solution to an ARE, it suffices to flip the sign of matrix $A(x)$ and solve the ARE using new $A(x)$ for the positive definite solution, then by flipping the sign of the resulted $P_{ss}(x)$, the negative definite solution to the original ARE will be resulted, [13].

Remark 2: Once the states are frozen, the DRE (6) is treated as a constant coefficient DRE as in linear finite-horizon solution and any other method available for solving such a DRE, e.g. Hamiltonian matrix method [14], can be used instead of the one selected in this paper. But the method selected here is seen to be much better in the implementation because of the existing remedy for the problem of long horizon, i.e., when $t_0 \ll t_f$. In case of using Hamiltonian matrix method for problems with long horizon, the matrix exponential of dimension $2n$ will have huge values which easily lead to computational problems and singularity in the required matrix inversion. Interested readers may refer to [14] for recent developments in Hamiltonian matrix method of solving a DRE.

III. NUMERICAL EXAMPLES

For numerical simulation and analysis, a benchmark nonlinear problem is selected; Van der Pol’s oscillator. The dynamics are given by
\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = (1 - x_1^2)x_2 - x_1 + u \]
(45)
and the selected weight matrices for the quadratic cost function (2) are
\[ Q = \text{diag}(10^3 10^3), R = 1, S = \text{diag}(10^9 10^9) \]
(46)
The following selection is made for the linear-like representation of the dynamics (45) to form (3)
\[ A(x) = \begin{bmatrix} 0 & 1 \\ -1 & 1 - x_1^2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]
(47)
The initial conditions are selected as $x_0 = [5, 5]^T$ to have a large nonlinearity in the dynamics.

The simulation is performed for three different final times of 1, 2, and 5 seconds and the resulting states trajectories of three different controllers are depicted in Fig. 1 to 3 for the final times, respectively. In these figures, the black plots denote the states trajectory of the finite-horizon controller developed in this paper, the blue plots denote the results of the (infinite horizon) SDRE one and this can be seen in Fig. 3 where the states trajectories for different initial conditions are depicted. As expected, once the horizon becomes larger, the Finite-SDRE results converge to the SDRE ones and this can be seen in Fig. 3 where the states trajectory of the Finite-SDRE lied on top of the SDRE one.

As described earlier, Finite-SDRE is an approximate solution to the problem. Comparing the black plots (Finite-
SDRE) with the red plots which reflect the optimal solution to the problem, the approximation error is not very large. Comparing the resulted cost-to-go of these two controllers for different final times of 1, 2, and 5 seconds, the cost-to-go of Finite-SDRE turned out to be 3%, 1% and 0.6% higher than the optimal cost-to-go, respectively.

Fig. 4 shows the control history generated using Finite-SDRE and applied on the system for the mentioned final times. Through this figure, it can be seen that how the control is time dependent and for different time-to-go’s, different control histories are generated in order to (approximately) minimize the cost function.

As far as the computational burden, propagating the states using Euler integration with the time step of $\Delta t = tf/1000$, the required time for simulation of the proposed controller for different final times which is composed of calculation of the approximate solution to the respective DRE for 1000 times with different state dependent values, tuned out to be around 1.0 sec. using MATLAB 2010 on a machine with Intel Core 2 Duo 2.66 GHz and 2 GB of RAM. This time for simulating the SDRE solution tuned out to be 0.6 sec. on this machine. Optimizing the code and using faster programming languages like C++ can decrease the required computational time by a considerable amount.

**IV. CONCLUSION**

Inspired by the great potential of SDRE for regulation of nonlinear systems, Finite-SDRE is developed here and shown to have a similar potential for finite horizon control of nonlinear systems. Many problems in control engineering fall in the category of finite-horizon control, e.g. guidance, path planning, and this method is a good candidate to be used in such applications.

**REFERENCES**


