A Study on Numerical Solutions of Hamilton-Jacobi-Bellman Equations Based on Successive Approximation Approach

Ichiro MARUTA*, Shuhei NISHIDA*, and Kenji FUJIMOTO*

Abstract: This paper presents a numerical approach to solve the Hamilton-Jacobi-Bellman (HJB) equation, which arises in nonlinear optimal control. In this approach, we first use the successive approximation to reduce the HJB equation, a nonlinear partial differential equation (PDE), to a sequence of linear PDEs called a generalized-Hamilton-Jacobi-Bellman (GHJB) equation. Secondly, the solution of the GHJB equation is decomposed by basis functions whose coefficients are obtained by the collocation method. This step is conducted by solving quadratic programming under the constraints which reflect the conditions that the value function must satisfy. This approach enables us to obtain a stabilizing solution of problems with strong nonlinearity. The application to swing up and stabilization control of an inverted pendulum illustrates the effectiveness of the proposed approach.

Key Words: nonlinear optimal control, Hamilton-Jacobi-Bellman equation, successive approximation.

1. Introduction

It is well known that the feedback solution to an optimal control problem is obtained by solving a Hamilton-Jacobi-Bellman (HJB) equation [1],[2]. If the plant dynamics is expressed by linear state equations and the cost functional is quadratic in the state and control, the HJB equation reduces to a Riccati equation. In this case, the corresponding optimal control problems are called linear quadratic (LQ) optimal control problems. Since the algorithms for solving Riccati equations are well established, the feedback solution to LQ optimal control problems can be easily obtained by means of existing solvers. On the other hand, in the case of nonlinear state equations, or non-quadratic cost functional, the corresponding HJB equation is a nonlinear PDE, which cannot be solved except for some special cases. Moreover, the solutions of HJB equations are not unique in general, which means that even if we obtain a solution, it does not necessarily correspond to the optimal feedback control. Because of these difficulties, nonlinear optimal control techniques are rarely applied to practical systems compared to the LQ case.

There have been numerous attempts to approximate the solutions of the HJB equations numerically. One of the approaches is the use of neural networks (NN), in which the residuals of the HJB equations are directly minimized by gradient-based optimization techniques [3]–[5]. Although these approaches are potentially capable of approximating the solution with high accuracy by exploiting the rich feature representation of NNs, it often falls into local minima and takes a great effort to tune the network properly, because of the nonlinearity of both the network and the HJB equation.

The stable manifold method is a promising approach to obtain stabilizing solutions to the HJB equation, which has been adopted to a wide range of optimal control problems [6]–[9]. Instead of directly approximating the solutions to the HJB equation, they practically simulate a flow of the associated Hamiltonian system backward from the neighborhood of origin to obtain points on the stable manifold and corresponding solutions to the HJB equation. An advantage of reformulating the optimal control problem to the computation of the stable manifold is that it can selectively obtain points on the solution that stabilizes the origin. However, the backward integration lacks numerical stability, and to generate trajectories that pass the desired states is not an easy task. Moreover, when the stabilizing solutions to the HJB equation are not unique, each of the points obtained by the stable manifold method is on one of the multiple stabilizing solutions. In such cases, we need to exclude points on stabilizing but not the optimal solutions, which is not an easy task. Because of these reasons, there still exist demands for methods to obtain approximation function directly from the equation.

The successive approximation approach (SAA) is another major approach, in which the solution is obtained as a parametric approximation function, and the nonlinearity of the HJB equation is abbreviated by an iterative scheme [10]–[23]. By employing the SAA, the problem to solve an HJB equation is reduced to recursively solving generalized Hamilton-Jacobi-Bellman (GHJB) equations, which are linear PDEs. The SAA has several desirable properties, which include the stability of the solutions obtained at each iterative step and uniform convergence to the stabilizing solutions [10]–[12]. Moreover, it is shown that the convergence rate of this algorithm can be quadratically fast under certain conditions in [22]. Finally, we would like to point out that the successive approximation based algorithm is also actively researched in reinforcement learning community, where the algorithms are called adaptive dynamic programming [24]–[26]. Among various solution methods of the GHJB equations, including the sum-of-squares approach [23] and the neural network least squares approach [20]–[22], the most major
one would be the Galerkin approach [13]–[19], in which the Galerkin method is applied to solve the GHJB equations. The Galerkin method enables us to obtain approximate solutions to GHJB equations as linear combinations of basis functions by solving algebraic equations. It is proved in [13] and [14] that as the number of basis function goes to infinity, an approximate solution will converge to the true solution of the GHJB equation. Moreover, numerically efficient modifications of this method are proposed to abbreviate the burden of numerical integration and thus enable application to systems with higher dimensional states [16]–[19]. However, this approach is based on the premise that approximation errors of GHJB equations are negligible, which is often not true. Especially, if the dynamics of a system have strong nonlinearity, the solutions of the GHJB equations are likely to be complex functions, which could hardly be approximated by a limited number of basis functions. In such cases, the numerical error at each step of the SAA accumulates and makes it difficult to obtain a meaningful solution.

In the proposed approach, the conventional method is modified in order to overcome these difficulties. In particular, problems of solving GHJB equations are reduced to constrained quadratic programming. Here the constraints reflect the conditions which the value function must satisfy, such as positive definiteness and relationships between the value function of the linearized system. By doing so, we selectively obtain numerical solutions with desirable properties so that the output of the iteration scheme will be a useful approximation. The proposed approach enables us to apply control synthesis via the HJB equation to problems with strong nonlinearity. This fact is illustrated by the application to swing-up and stabilization of an inverted pendulum.

The paper is organized as follows. In Section 2, the optimal control problem which we address throughout this paper is introduced, and we point out that the solution can be obtained by solving the HJB equation. In Section 3, we illustrate how the successive approximation scheme reduces the nonlinear HJB equation to a sequence of linear GHJB equations. Section 4 is devoted to the explanation of the proposed method, where we solve the constrained quadratic programming to obtain the solution of GHJB equations. In Section 5, the proposed method is applied to swing-up and stabilization task of an inverted pendulum. Finally, we make some concluding remarks in Section 6.

2. Problem Statement

Consider an input affine nonlinear dynamical system

\[ \dot{x} = f(x) + g(x)\eta(t), \]

where \( x \in \mathbb{R}^n \) is the state variable; \( f : \mathbb{R}^n \to \mathbb{R}^n \) and \( g : \mathbb{R}^n \to \mathbb{R}^{n \times m} \) are functions which characterize the dynamics; and \( \eta : \mathbb{R} \to \mathbb{R}^m \) is the control input. The cost functional is defined as

\[ J(x_0, \eta(\cdot)) = \int_0^\infty l(\phi(t; x_0, t_0, \eta(\cdot))) + ||\eta(t)||_R^2 \, dt, \]

where \( \phi(t; x_0, t_0, \eta) \) is the trajectory for the system equation (1) when starting from state \( x_0 \) at time \( t_0 \) and using input \( \eta \). For the simplicity of notation, we write \( \phi(t) = \phi(t; x_0, t_0, \eta(\cdot)) \) if \( x_0, t_0, \) and \( \eta \) are clear. Moreover, we assume that \( f \) is positive semi-definite and \( ||\eta||_R^2 = \eta^T R \eta, \) where \( R \in \mathbb{R}^{m \times m} \) is a positive definite symmetric matrix.

Let us suppose that the input \( \eta(t) \) is determined by a state feedback control law \( u, \) as \( \eta(t) = u(\phi(t)) \). By substituting it into (1), the state equation is given as

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

When the feedback control law \( u : \mathbb{R}^n \to \mathbb{R}^m \) is given, the cost functional (2) depends only on the initial state. Therefore, we redefine the cost functional which corresponds to control \( u \) as a function of the initial state:

\[ V(x; u) := \int_0^\infty l(\phi(t)) + ||u(\phi(t))||_R^2 \, dt, \]

where \( \phi(t) \) represents \( \phi(t; x, t_0, u(\phi(t))) \). Here, \( V(x; u) \) is called a value function. The optimal value function is defined as

\[ V^*(x) := V(x; u^*), \]

where \( u^* \) is the optimal control law. It is derived from the theory of dynamic programming that the optimal value function can be obtained by solving the HJB equation

\[ 0 = \min_{u \in \mathcal{U}} \left\{ \frac{\partial V^*}{\partial x} (f(x) + g(x)u(x)) + l(x) + ||u(x)||_R^2 \right\}, \]

Moreover, the relation between the optimal value function and the optimal control law is known as follows:

\[ u^* = \arg \min_{u \in \mathcal{U}} \left\{ \frac{\partial V^*}{\partial x} (f(x) + g(x)u(x)) + l(x) + ||u(x)||_R^2 \right\}, \]

By substituting (7) to (6), we obtain

\[ \frac{\partial V^*}{\partial x} f(x) - \frac{1}{4} \frac{\partial V^*}{\partial x} g(x)R^{-1}g(x)^T \frac{\partial V^*}{\partial x} + l(x) = 0. \]

Hence, the HJB equation could be written as a nonlinear PDE with respect to \( V^* \). The optimal value function \( V^* \) is a positive definite solution to nonlinear PDE (8), and the optimal control \( u^* \) can be calculated by (7) from \( V^* \).

3. Successive Approximation Approach [10]–[12]

In the SAA, a sequence of the value function, which converges to a solution of the HJB equation, is generated from an arbitrary admissible control by iterating value estimation and policy improvement, which corresponds to (6) and (7) respectively. A control \( u \) is defined to be admissible in \( \mathbb{R}^m \) if it satisfies the following properties:

- \( u \) is continuous on \( \mathbb{R}^n \)
- \( u(0) = 0 \)
- the origin of (3) is asymptotically stable
- \( J(x_0, u(\phi(\cdot))) < \infty \) for all \( x_0 \in \mathbb{R}^n \)

Assume that an admissible control \( \alpha \) is given. It is proved in [11] that the value function (4) which corresponds to an admissible control \( \alpha \) is the positive definite solution to the GHJB equation (9):

\[ 0 = \frac{\partial V}{\partial x} \left( f(x) + g(x)u(x) \right) + l(x) + ||u(x)||_R^2, \]
which is obtained by substituting $u'$ instead of $u^*$ to the HJB equation (6). We call this manipulation, where we obtain the value function $V^i$ which corresponds to $u'$. The past study shows that if we improve policy by (10).

Next, we analyze the computational complexity of the method. We have obtained the following equation for the numerical solution of the HJB equation in the region of interest $\Omega \subset \mathbb{R}^n$.

\[ V(x) = c^T \Phi(x) \] (11)

where $\Phi = (\phi_1, \phi_2, ..., \phi_M)^T$ is a basis vector and $c = (c_1, c_2, ..., c_M)^T$ is a coefficient vector. Let $\xi_k : \Omega \rightarrow \mathbb{R}$, $k = 1, 2, ..., M$ be linearly independent weight functions. By substituting (11) to the GHJB equation and integrating it with weight $\chi_k$, we obtain

\[
\int_\Omega \int_\Delta \left[ c^T \frac{\partial \Phi(x)}{\partial x} f(x) + c^T \frac{\partial \Phi(x)}{\partial x} g(x)u(x) \right] \, dx = 0 \quad (k = 1, 2, ..., M). \tag{12}
\]

Note that these are linear algebraic equations with respect to the coefficient vector $c$. Consequently, the coefficient vector $c$ is determined uniquely by solving this equation. Here, the linear independence of weight functions ensures the existence of the solution to the linear equation. In the conventional successive Galerkin approach (SGA), the weight functions are chosen as $\chi_k = \phi_k$.

Although the Galerkin method is empirically known to be more accurate than other weighted residual methods, it often fails to obtain an admissible solution when applied to hard problems, such as problems with highly nonlinear dynamics. In these cases, we may want to obtain a solution that could be less accurate but has a desirable property such as admissibility. Therefore, in the proposed approach, we will reformulate the equation of weighted residual (12) into a minimization problem so that we could obtain the solution with least residuals among those that satisfy the constraints that reflect necessary conditions of admissibility. For this purpose, instead of the Galerkin method, we employ the least squares method where the weight functions are chosen as $\chi_k = \phi_k$.

Algorithm 1: Successive Approximation Algorithm

**Input:** Initial controller $u^0$  
**Output:** $V^0$, $u^0$  
for $i = 0$ to $N - 1$ do  
$V^i \leftarrow G u'$  
$u^{i+1} \leftarrow S V^i$  
end for

In this case, the equation of weighted residual (12) is the first order necessary conditions for a solution of minimization problem

\[
\min_c \int_\Omega \left[ c^T \frac{\partial \Phi(x)}{\partial x} f(x) + c^T \frac{\partial \Phi(x)}{\partial x} g(x)u(x) \right] \, dx + l(x) + \|u(x)\|_R^2 \tag{14}
\]

Moreover, in order to abbreviate the computational burden, we replaced the integral by the sum of the values on sample points $x_k$, $k = 1, 2, ..., m$.

\[
\min_c \sum_{k=1}^m \left[ c^T \frac{\partial \Phi(x_k)}{\partial x} f(x_k) + c^T \frac{\partial \Phi(x_k)}{\partial x} g(x_k)u(x_k) \right] + l(x_k) + \|u(x_k)\|_R^2. \tag{15}
\]

For the simplicity of notation and to clarify that it is a quadratic programming problem, we reformulate (15) as

\[
\min_c (\Xi c + Z)^2, \tag{16}
\]

where

\[
\Xi = \begin{pmatrix} \xi_{11} & \cdots & \xi_{1M} \\ \vdots & \ddots & \vdots \\ \xi_{m1} & \cdots & \xi_{mM} \end{pmatrix}, \quad Z = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_M \end{pmatrix}. \tag{17}
\]

\[
\xi_{k1} = \frac{\partial \phi(x_k)}{\partial x} f(x_k) + g(x_k)u(x_k), \quad \xi_{k2} = l(x_k) + \|u(x_k)\|_R^2, \tag{18}
\]

In order to approximate the solution in a way that the corresponding control satisfies the condition for admissible control, we imposed constraints to the aforementioned quadratic programming. Firstly, the approximated value function $c^T \Phi(x)$ needs to be positive definite, which is practically implemented as

\[
c^T \Phi(0) = 0 \tag{20}
\]

and

\[
\Gamma c \geq 0, \tag{21}
\]

where $\Gamma$ represents

\[
\Gamma = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_M) & \cdots & \phi_M(x_M) \end{pmatrix}. \tag{22}
\]

Secondly, we make use of the fact that in the neighborhood of the origin, the optimal value function of the nonlinear system is equal to that of the linearized system. In particular, for the gain matrix $P$ obtained by solving the Riccati equation of the linearized system and the value of hessian of $V^*$ at the origin, the relation

\[ \chi(x) = \frac{\partial}{\partial x} \left[ c^T \frac{\partial \Phi(x)}{\partial x} f(x) + g(x)u(x) \right] \tag{13} \]
Algorithm 2 Proposed Method

Input: Initial controller $u^0$

Output: $c^N, u^N$

for $i = 0$ to $N - 1$ do

$c^i \leftarrow \arg \min \{ \mathbb{E}c + Z \} + \kappa \left\| \frac{\partial^2 \Phi}{\partial x^2} \right\|_{c=0} - P \}$

$u^{i+1} \leftarrow \frac{-1}{2} R^{-1} g(x^i) \left( c^i \frac{\partial \Phi}{\partial x} \right)^T$

end for

$P = \frac{\partial^2 V^*}{\partial x^2} \bigg|_{x=0}$ (23)

holds. In order to utilize this information, we add $\left\| \frac{\partial^2 \Phi}{\partial x^2} \right\|_{c=0} - P$ to the loss function as a penalty term. Here, the magnitude of the penalty term is tuned with the parameter $\kappa$ to obtain the intended solution. Note that a large value of $\kappa$ ensures that the obtained solution is close to the optimal solution of the linearized system nearby the origin, but this may degrade the accuracy in other regions since the solution is approximated by a finite sum of polynomial basis. The resulting constrained quadratic programming problem is as follows:

$$\begin{align*}
\text{minimize} & \quad \left\| \mathbb{E}c + Z \right\|^2 + \kappa \left\| \frac{\partial^2 \Phi}{\partial x^2} \right\|_{c=0} - P \\
\text{subject to} & \quad \Gamma c \geq 0, \\
& \quad c^T \Phi(0) = 0.
\end{align*}$$ (24)

In the proposed approach, the value estimation is performed by solving this problem, instead of the GHJB equation. Here, it is the coefficient vector $c$ that is updated in each step of the SAA. Therefore we write the coefficient vector at step $i$ as $c^i$ hereafter.

The policy improvement is performed as

$$u^{i+1} = \frac{-1}{2} R^{-1} g(x^i) \left( c^i \frac{\partial \Phi}{\partial x} \right)^T.$$

(25)

The procedure of the proposed algorithm is described in Algorithm 2.

5. Numerical Example

We now see that the proposed method is capable of obtaining stabilizing solutions of problems with strong nonlinearity by applying it to swing-up and stabilizing control of an inverted pendulum. Inverted pendulums have been one of the most popular benchmark problems in nonlinear control [27]. To design a single smooth controller that realizes both swing up and stabilization with limited control effort is especially a challenging task since the pendulum exhibits strong nonlinearity in the region apart from the target state. In this section, we apply the proposed method to the design of an optimal regulator that accomplishes this task, in order to illustrate the capability of obtaining a stabilizing controller to complex problems with strong nonlinearity.

5.1 Problem Setting

Consider an inverted pendulum system illustrated in Fig. 1, where $\theta$ is taken clockwise and $\dot{\theta} = 0$ when the mass is on the top. The state equation of this system can be written as

$$\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= \frac{g}{\ell} \sin(x_1) + \frac{1}{m\ell^2} u,
\end{align*}$$

(26)

where the state variable $x = (x_1, x_2)^T = (\theta, \dot{\theta})^T$ and the physical parameters are chosen as $m = 1, \ell = 1, g = 9.8$. The cost

![Fig. 1 Inverted pendulum.](image)

![Fig. 2 Value function and control.](image)
The functional is defined as
\[ J(x_0, u) = \int_0^\infty x^T Q x + u^T R u \, dt, \]  
(27)
where the weight matrices \( Q \) and \( R \) are determined as
\[
Q = \begin{pmatrix} 1 & 0 \\ 0 & 0.01 \end{pmatrix}, \quad R = 5.
\]  
(28)

To approximate the solution, we used 1225 polynomial basis functions which are defined by
\[
\Phi(x) = \text{vec} \left( \begin{pmatrix} \cdots \\ T_0(x_1) \\ \cdots \\ T_0(x_2) \\ \cdots \\ T_34(x_1) \\ T_34(x_2) \end{pmatrix} \right),
\]  
(29)
where \( T_l(x) \) denotes the Chebyshev polynomial of degree \( l \). The sample points for integration and constraints are 50×50 and 40×40 uniform grid points in \([-2.1\pi, 2.1\pi] \times [-12, 12] \) respectively. Both the number of points for integration and the number of points for imposing constraints are carefully chosen so that they will be large enough to approximate integral with sufficient accuracy and to obtain a solution that satisfies constraints everywhere in \([-2.1\pi, 2.1\pi] \times [-12, 12] \) but small enough to keep the computation tractable. We started the iteration from \( u_0 = 0 \), which is not an admissible control and therefore violating the assumption of the theoretical guarantee of convergence. This is because when we started from an admissible control obtained via exact linearization, the iteration is stuck in a control with a high cost. This may be the consequence of the fact that the HJB equation has many stabilizing solutions in the inverted pendulum case. Starting from \( u = 0 \), we performed value estimation and policy improvement 50 times. To solve the problem (24), we used CVX, a package for specifying and solving convex programs [28],[29] and Gurobi [30], a mathematical programming solver.

5.2 Results

The value function and control obtained by the proposed method are shown in Fig. 2. The hatched surface in Fig. 2 shows the LQ optimal control designed for the linearized system obtained by linearizing the state equation (26) at the origin. It can be seen that the value function and control are obtained as complex nonlinear functions, and the magnitude of control obtained by the proposed method is smaller than the one obtained by the LQ optimal control.

We also conducted a swing-up simulation of a pendulum with the controller in Fig. 2 from an initial state \((-\pi, 0)^T\), which corresponds to the pending position. The result is shown in Fig. 3, where the plot above and below show the control input \( u \) and the angle \( \theta \), respectively. For comparison, the simulation result of the LQ controller is also plotted with the gray dashed line. The controller obtained by the proposed method swings the pendulum five times before it reaches the top, whereas the LQ controller lifts the pendulum at a stroke. A consequence of this fact is that the controller obtained by the proposed method swings up and stabilizes the pendulum with extremely small control efforts compared to the LQ controller. As a result, when the controller designed by the proposed method is applied, the cost along this trajectory is 153, which is reduced by about 90% from the cost of the LQ controller.

In addition, simulated state trajectories in phase space starting from multiple initial states are displayed in Fig. 4. The bold spiral line is the state trajectory starting from the pending position \( x_0 = (-\pi, 0)^T \), which corresponds to the response displayed in Fig. 3. The domain surrounded by the closed curve with the
The dashed line is the region of attraction. In this figure, we can see that the controller obtained by the proposed method stabilizes all the trajectories from initial states included in a large connected domain in the state space.

Note that since in practical simulation we handle finite region of interest $\Omega \subset \mathbb{R}^n$ instead of $\mathbb{R}^n$ itself, we cannot guarantee that the trajectory starting from $\Omega$ converges to the origin without reaching $\mathbb{R}^n \setminus \Omega$. When the trajectory reaches $\mathbb{R}^n \setminus \Omega$, where the numerical solution is not obtained, the trajectory cannot be expected to reach the origin. Therefore $\Omega$ needs to be large enough to obtain a controller with sufficient region of attraction.

Finally, we show the result of the case where the conventional Galerkin method is applied to the identical problem. In Fig. 5, the value function obtained by the Galerkin method is displayed. We used the same basis functions and performed 50 times iteration. The integrals in equations of weighted residual are computed numerically by trapezoidal integration with $150 \times 150$ grids. The value function in Fig. 5 is not positive definite and does not correspond to admissible control. Since the Galerkin method solves linear equations without constraints at each step, it could not prevent the numerical error due to the lack of the number of basis functions from making the solution inadmissible, which led to the failure of the numerical scheme in this example. Therefore in this numerical example, we conclude that a stabilizing controller is obtained by reformulating the linear equation of weighted residuals (12) into the constrained quadratic programming problem (24).

6. Conclusion

This paper proposed a numerical method for solving HJB equations based on the SAA. In particular, we performed value estimation by solving quadratic programming problems under the constraints that reflect an attribute of the control problem, which enabled this approach to deal with complex problems with strong nonlinearity. This was illustrated by the numerical example of swing up and stabilizing control of an inverted pendulum. Finally, a possible future work will be analyzing the behavior of solutions of GHJB equations when started from inadmissible control such as $u = 0$.

References

This work was supported by JSPS KAKENHI Grant Number JP16H06093.

[1] A.E. Bryson and Y.C. Ho: Applied Optimal Control, Blaisdell, New York, 1969.
[2] D.P. Bertsekas: Dynamic Programming and Optimal Control, Athena Scientific, 2005.
[3] X. Wang and K. Shimizu: Approximate solution of Hamilton-Jacobi-Bellman equation by using neural networks and matrix calculus techniques, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E84-A, No. 6, pp. 1549–1556, 2001.
[4] C.J. Goh: On the nonlinear optimal regulator problem, Automatica, Vol. 29, No. 3, pp. 751–756, 1993.
[5] R. Munos, L.C. Baird, and A.W. Moore: Gradient descent approaches to neural-net-based solutions of the Hamilton-Jacobi-Bellman equation, Proceedings of the International Joint Conference on Neural Networks, pp. 2152–2157, 1999.
[6] N. Sakamoto and A.J. van der Schaft: Analytical approximation methods for the stabilizing solution of the Hamilton-Jacobi equation, IEEE Transactions on Automatic Control, Vol. 53, No. 10, pp. 2335–2350, 2008.
[7] R. Fujimoto and N. Sakamoto: The stable manifold approach for optimal swing up and stabilization of an inverted pendulum with input saturation, Proceedings of the 18th IFAC World Congress, pp. 8046–8051, 2011.
[8] Y. Oishi and N. Sakamoto: Numerical computational improvement of the stable-manifold method for nonlinear optimal control, Proceedings of the 20th IFAC World Congress, pp. 5103–5108, 2017.
[9] T. Horibe and N. Sakamoto: Optimal swing up and stabilization control for inverted pendulum via stable manifold method, IEEE Transactions on Control Systems Technology, Vol. 26, No. 2, pp. 708–715, 2018.
[10] R.J. Leake and R.-W. Liu: Construction of suboptimal control sequences, SIAM Journal on Control, Vol. 5, No. 1, pp. 54–63, 1967.
[11] G.N. Saridis and C.G. Lee: An approximation theory of optimal control for trainable manipulators, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 3, pp. 152–159, 1979.
[12] G.N. Saridis and J. Balaram: Design of suboptimal controllers for nonlinear systems, Proceedings of the 9th IFAC World Congress, pp. 293–298, 1984.
[13] R.W. Beard, G.N. Saridis, and J.T. Wen: Improving the performance of stabilizing controls for nonlinear systems, IEEE Control Systems Magazine, Vol. 16, No. 5, pp. 27–35, 1996.
[14] R.W. Beard, G.N. Saridis, and J.T. Wen: Approximate solutions to the time invariant Hamilton Jacobi Bellman equation, Journal of Optimization Theory and Applications, Vol. 96, No. 3, pp. 589–626, 1998.
[15] R.W. Beard and T.W. McLain: Successive Galerkin approximation algorithms for nonlinear optimal and robust control, International Journal of Control, Vol. 71, No. 5, pp. 717–743, 1998.
[16] J. Lawton and R.W. Beard: Numerically efficient approximations to the Hamilton-Jacobi-Bellman equation, Proceedings of the 1998 American Control Conference, pp. 195–199, 1998.
[17] J.W. Curtis and R.W. Beard: Successive collocation: An approximation to optimal nonlinear control, Proceedings of the 2001 American Control Conference, pp. 3481–3485, 2001.
[18] M.B. Horowitz, A. Damle, and J.W. Burdick: Linear Hamilton Jacobi Bellman equations in high dimensions, Proceedings of the 53rd IEEE Conference on Decision and Control, pp. 5880–5887, 2014.
[19] D. Kalise and K. Kunisch: Polynomial approximation of high-
dimensional Hamilton-Jacobi-Bellman equations and applications to feedback control of semilinear parabolic PDEs, *SIAM Journal on Scientific Computing*, Vol. 40, No. 2, pp. A629–A652, 2018.

[20] M. Abu-Khalaf and F.L. Lewis: Nearly optimal HJB solution for constrained input systems using a neural network least-squares approach, *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 943–948, 2002.

[21] F.L. Lewis and M. Abu-Khalaf: A Hamilton-Jacobi setup for constrained neural network control, *Proceedings of the 2003 IEEE International Symposium on Intelligent Control*, pp. 1–15, 2003.

[22] Z. Chen and S. Jagannathan: Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete-time systems, *IEEE Transactions on Neural Networks*, Vol. 19, No. 1, pp. 90–106, 2008.

[23] A. Wernrud and A. Rantzer: On approximate policy iteration for continuous-time systems, *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1453–1458, 2005.

[24] F.Y. Wang, H. Zhang, and D. Liu: Adaptive dynamic programming: An introduction, *IEEE Computational Intelligence Magazine*, Vol. 4, No. 2, pp. 39–47, 2009.

[25] F.L. Lewis and D. Vrabie: Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits and Systems Magazine*, Vol. 9, No. 3, pp. 32–50, 2009.

[26] A. Al-Tamimi, F.L. Lewis, and M. Abu-Khalaf: Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 38, No. 4, pp. 943–949, 2008.

[27] O. Boubaker: The inverted pendulum benchmark in nonlinear control theory: A survey, *International Journal of Advanced Robotic Systems*, Vol. 10, No. 5, p. 233, 2013.

[28] M. Grant and S. Boyd: CVX: Matlab software for disciplined convex programming, version 2.1, March 2014.

[29] M. Grant and S. Boyd: Graph implementations for nonsmooth convex programs, V. Blondel, S. Boyd, and H. Kimura, eds., *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag, 2008.

[30] Gurobi Optimization, LLC: Gurobi optimizer reference manual, 2018.

Ichiro MARUTA (Member)

He received the Bachelor of Engineering, Master of Informatics, and Doctor of Informatics degrees from Kyoto University, Kyoto, Japan, in 2006, 2008, and 2011, respectively. He was a research fellow of the Japan Society for the Promotion of Science from 2008 to 2011. From 2012 to 2017, he was an Assistant Professor at the Graduate School of Informatics, Kyoto University. In 2017, he joined the Graduate School of Engineering, Kyoto University, as a Lecturer of the Department of Aeronautics and Astronautics, and since 2019, he is an Associate Professor.

Shuhei NISHIDA

He received his B.S. degree in Engineering from Kyoto University, Japan, in 2019. He is currently an M.S. student at Kyoto University. His research interests include optimal control theory.

Kenji FUJIMOTO (Member)

He received his B.S. and M.S. degrees in Engineering and Ph.D. degree in Informatics from Kyoto University, Japan, in 1994, 1996, and 2001, respectively. He is currently a professor of Graduate School of Engineering, Kyoto University, Japan. From 1997 to 2004, he was a research associate of Graduate School of Engineering and Graduate School of Informatics, Kyoto University, Japan. From 2004 to 2012, he was an associate professor of Graduate School of Engineering, Nagoya University, Japan. His research interests include nonlinear control and stochastic systems theory.