Numerical Solutions of Hamilton-Jacobi Inequalities by Constrained Gaussian Process Regression

Kenji FUIJIMOTO*, Hirofumi BEPPU*, and Yuji TAKAKI*

Abstract: This paper proposes numerical solutions of Hamilton-Jacobi inequalities based on constrained Gaussian process regression. While Gaussian process regression is a tool to estimate an unknown function from its input and output data conventionally, the proposed method applies it to solving a known partial differential inequality. This is done by generating sample data pairs of states and corresponding values of the unknown function satisfying the inequality. A formal algorithm to execute such a procedure to obtain the probability of a solution to the Hamilton-Jacobi inequality is proposed. In addition, a nonstationary covariance function is introduced to increase the accuracy of the solutions and to reduce the computational cost. Furthermore, its hyper parameters are optimized using an empirical gradient method.

Key Words: nonlinear control, optimal control, Hamilton-Jacobi inequalities, Gaussian processes.

1. Introduction

Hamilton-Jacobi inequalities are fundamental to optimal control theory [1]–[4], $H_\infty$ control theory [5]–[8], dissipative systems theory [9], and so on. For linear systems, the inequalities reduce to the quadratic inequalities called Riccati inequalities for which there are efficient established algorithms including linear matrix inequalities (LMI) [10] implemented in some softwares, e.g., MATLAB, to solve them numerically. When the systems are nonlinear, however, Hamilton-Jacobi inequalities lead to nonlinear partial differential inequalities which are generally difficult to be solved analytically. In order to solve such problems, several algorithms to obtain approximate solutions have been proposed, for instance, approximate solutions based on series expansion [11], a method using nonlinear matrix inequalities (NLMl) [12], a method based on neural networks [13], an improved method of existing techniques by convex combination [14], and so on. However, a practically efficient algorithm to solve the inequalities accurately has not been found, and therefore application examples of nonlinear systems are few compared with those of linear systems.

On the other hand, Gaussian process regression [15],[16], which is a method to estimate an unknown function from given input-output data, has gathered much attention in machine learning society. This method has been applied to some control problems recently, e.g., a controller design to stabilize Gaussian process dynamical models for unknown nonlinear systems [17] and an output feedback controller design for Gaussian process state space models [18]. Gaussian process regression has the following properties. Firstly, in Gaussian process regression, it is not assumed that the unknown function has a parametric form. This enables us to estimate more varieties of functions than the conventional methods based on series expansion. Secondly, Gaussian process regression gives an analytic function of the input variable as the estimation of the given input-output map which is helpful to analyze and to improve the method theoretically. Thirdly, Gaussian process regression can easily incorporate the information on the derivative of the unknown function into estimation processes [19]. Fourthly, in Gaussian process regression, we can impose inequality constraints on the unknown function [20]. As a result, a function satisfying the inequality constraints at arbitrarily assigned discrete points can be derived. Due to those properties, Gaussian process regression is expected to give a new approach to solve Hamilton-Jacobi inequalities.

In this paper, new numerical solutions to Hamilton-Jacobi inequalities by using Gaussian process regression are proposed. While Gaussian process regression is a tool to estimate an unknown function from its input and output data conventionally, the proposed method applies it to solving a known partial differential inequality, the Hamilton-Jacobi inequality. This is accomplished by selecting sample data pairs of states and the corresponding values of the unknown function satisfying the inequality. How to construct a formal algorithm to generate such samples to ensure certain control criteria statistically is the main issue of the present paper. Gaussian process regression is specified by two quantities, a mean function and a covariance function, which represent a rough sketch of the unknown function and its uncertainty, respectively. Instead of using a conventional quadratic covariance function, a new nonstationary covariance function is proposed to deal with its highly nonlinear dependence on the state variable which the solutions of Hamilton-Jacobi inequalities generally have. Furthermore, a systematic design procedure to optimize adjustable parameters contained in those covariance functions which affect the estimation results is given by maximizing a cost function based on the marginal likelihood.

This paper is organized in the following manner. Section 2 introduces Hamilton-Jacobi inequalities which the proposed method deals with. Section 3 proposes numerical solutions via Gaussian process regression with equality and inequality constraints. Section 4 verifies the effectiveness of the proposed...
method by applying the method for problems of nonlinear $H_{\infty}$ control. The last section concludes the paper.

2. Hamilton-Jacobi Inequalities

In this section, we introduce Hamilton-Jacobi inequalities which the proposed method deals with [5]–[8].

2.1 Introduction of Hamilton-Jacobi Inequalities

This subsection gives a class of nonlinear partial differential inequalities used in nonlinear optimal control such as nonlinear $H_{\infty}$ control [5]–[8]. Let us consider the following generalized plant

$$\begin{align*}
\dot{x} &= f(x) + g(x)u + g_d(x)w,
\dot{z} &= h(x) + j(x)u,
\end{align*}$$

(1)

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, $u_d \in \mathbb{R}^q$ is the disturbance, and $z \in \mathbb{R}^p$ is the performance output. The functions $f(x), g(x), g_d(x), h(x),$ and $j(x)$ are all given. It is assumed that $f(0) = 0$ and $h(0) = 0$. The purpose of nonlinear $H_{\infty}$ control is to find a control input $u = u_*(x)$ such that the cost function with a positive constant $\gamma > 0$

$$J(u_*, z) = \int_0^\infty \|z(t)\|^2 - \gamma^2\|u_0(t)\|^2 dt, \quad x(0) = 0$$

(2)

is negative for any $u_0(t)$, that is, the map $u_0 \mapsto z$ has an $L_2$ gain no greater than $\gamma$. In order to obtain $u = u_*(x)$, one needs to solve the Hamilton-Jacobi inequality expressed as

$$\frac{\partial v(x)}{\partial x} A(x) \frac{\partial v(x)^T}{\partial x} + \frac{\partial v(x)}{\partial x} b(x) + c(x) < 0,$$

(3)

$$v(0) = 0, \quad \left. \frac{\partial v(x)}{\partial x} \right|_{x=0} = 0,$$

(4)

where an unknown positive definite function $v(x)$ called a value function is to be found. Here $A(x) \in \mathbb{R}^{n \times n}$ is a symmetric matrix valued function, $b(x) \in \mathbb{R}^n$ is a vector valued function, and $c(x) \in \mathbb{R}$ is a scalar valued function where the functions $A(x), b(x),$ and $c(x)$ are known functions calculated using the functions $f(x), g(x), g_d(x), h(x),$ and $j(x)$ in (1). For simplicity, suppose that equations $h(x)^Tj(x) = 0$ and $j(x)^Tj(x) = I$ hold where the matrix $I$ is the identity matrix. Then the functions $A(x), b(x),$ and $c(x)$ are given by

$$A(x) = \frac{1}{4} \left( \frac{1}{\gamma} g_d(x)^Tg_d(x) - g(x)^Tg(x) \right),$$

(5)

$$b(x) = f(x),$$

(6)

$$c(x) = h(x)^Tj(x).$$

(7)

A solution $u_*$ is given by $u_*(x) = -(1/2)g_d(x)^T(\partial v(x)/\partial x)^T$ using a value function $v(x)$ satisfying (3) and (4). This is a typical inequality appearing in nonlinear control problems.

2.2 Hamilton-Jacobi Inequalities of a Linearized System

This subsection shows the relationship between Hamilton-Jacobi inequalities and algebraic Riccati inequalities.

When the generalized plant (1) is linearized around the origin, the inequality (3) reduces to an algebraic Riccati inequality by selecting $v(x) = x^T P x$

$$P \left. \frac{\partial b(x)}{\partial x} \right|_{x=0} + \left. b(x)^T \right|_{x=0} P + 4PA(0)P + Q < 0,$$

(8)

where a matrix $Q$ has an appropriate dimension. Thereby a solution to the Hamilton-Jacobi inequalities for a nonlinear system near the origin can be approximated by $v(x) = x^T P x$ using a solution $P$ to the corresponding algebraic Riccati inequality (8).

3. Main Result

This section gives the main result, an approximate solution to Hamilton-Jacobi inequalities based on Gaussian process regression [15],[16]. Section 3.1 introduces the conventional method of Gaussian process regression. Section 3.2 explains how to obtain an approximate solution to Hamilton-Jacobi inequalities by using (4) for Gaussian process regression. Section 3.3 gives an approximate solution by adding a constraint derived from the inequality (3) to the result of Section 3.2.

3.1 Introduction of Gaussian Process Regression

The aim of Gaussian process regression is to estimate an unknown function $v(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ using the input-output data

$$v = v(x),$$

(9)

where $x = [x_1, \ldots, x_d]^T \in \mathbb{R}^n$ is a state variable (or an input variable of the function $v(\cdot)$) and $v \in \mathbb{R}$ in the left hand side of (9) is an output variable of the function. A Gaussian process is used to describe a distribution over functions. This fact is expressed formally in Definition 1 as follows [16].

Definition 1 A set of random variables is said to be a Gaussian process if the joint probability distribution of any finite elements among them is Gaussian.

It is assumed that the unknown function $v(x)$ in (9) follows a Gaussian process. That is, a joint probability of values of the function $v(\cdot)$ corresponding to an arbitrary set of state variables $x$ follows a multivariate Gaussian distribution. Thus the function $v(x)$ is distributed according to an unknown probability $p(v(x))$, namely,

$$v(x) \sim p(v(x)).$$

(10)

The Gaussian process is determined only by the mean function $\mu(x) \in \mathbb{R}$ of $v(x)$ and the covariance function $k(x, x') \in \mathbb{R}$ of the output variables $v(x)$ and $v(x')$ corresponding to arbitrary state variables $x$ and $x'$. They are expressed as

$$\mu(x) := \mathbb{E}[v(x)],$$

(11)

$$k(x, x') := \mathbb{E}[(v(x) - \mu(x))(v(x') - \mu(x'))],$$

(12)

where the expectation operator $\mathbb{E}[\cdot]$ is defined as

$$\mathbb{E}[v(x)] := \int v(x)p(v(x)) \, dv,$$

(13)

$$\mathbb{E}[(v(x) - \mu(x))(v(x') - \mu(x'))] := \int v(x)v(x')p(v(x), v(x')) \, dv \, dv' - \mu(x)\mu(x')$$

(14)

by using the symbols in (11) and (12) where $v = v(x)$ and $v' = v(x')$. Since the probability $p(v(x))$ is unknown, the mean function $\mu(x)$ in (11) is determined based on prior knowledge. If there is no prior knowledge, zero mean function $\mu(x) \equiv 0$ is
usually used [16]. The way of determining the covariance function \( k(x, x') \) in (12) is given in Section 3.4. Using the functions (11) and (12), the Gaussian process of \( v(x) \) in (9) is described as
\[
v(x) \sim \mathcal{GP}(v_0(x), k(x, x')).
\] (15)

A simple example of the Gaussian process (15) is the joint probability of two outputs of the unknown function \( v(\cdot) \) given by a multivariate Gaussian distribution according to Definition 1 as
\[
p(v(x), v(x')) := N\left( \begin{bmatrix} v(x) \\ v(x') \end{bmatrix} \bigg| \begin{bmatrix} v_0(x) \\ v_0(x') \end{bmatrix}, K(x, x'), \right) \exp \left\{ -\frac{1}{2} \begin{bmatrix} v(x) - v_0(x) \\ v(x') - v_0(x') \end{bmatrix}^T K^{-1} \begin{bmatrix} v(x) - v_0(x) \\ v(x') - v_0(x') \end{bmatrix} \right\}.
\] (16)

Here \( x \) and \( x' \) are arbitrary state variables, \( \mathcal{N} \) denotes the Gaussian distribution, \( v_0(\cdot) \) is a known mean function, and \( K(x, x') \in \mathbb{R}^{2 \times 2} \) is a known matrix of covariance functions defined as
\[
K(x, x') = \begin{bmatrix} k(x, x) & k(x, x') \\ k(x', x) & k(x', x') \end{bmatrix}.
\] (17)

In a typical problem of Gaussian process regression, a training data set \( \mathcal{D}^0 = \{ (x_i, v_i^0), i = 1, \ldots, N \} \) is given [15],[16]. Its elements satisfy the model (9) as
\[
v_i^0 = v(x_i).
\] (18)

The superscript \( (\cdot)^0 \) is used for the training data. Following the same procedure as the example in (16), the joint probability of a new output \( v(x) \) corresponding to an arbitrary state variable \( x \) and the output data \( v_1^0, \ldots, v_N^0 \) in the training data set \( \mathcal{D}^0 \) is given by a multivariate Gaussian distribution as
\[
p(v_1^0, \ldots, v_N^0, v(x)) := N\left( \begin{bmatrix} v_1^0 \\ \vdots \\ v_N^0 \\ v(x) \end{bmatrix} \bigg| \begin{bmatrix} v_0(1) \\ \vdots \\ v_0(N) \\ v_0(1) \end{bmatrix}, \Sigma^0(x) \right) \exp \left\{ -\frac{1}{2} \begin{bmatrix} v_1^0 - v_0(1) \\ \vdots \\ v_N^0 - v_0(N) \\ v(x) - v_0(x) \end{bmatrix}^T \Sigma^0(x)^{-1} \begin{bmatrix} v_1^0 - v_0(1) \\ \vdots \\ v_N^0 - v_0(N) \\ v(x) - v_0(x) \end{bmatrix} \right\}.
\] (19)

Here,
\[
v^0 = [v_1^0, \ldots, v_N^0] \in \mathbb{R}^N,
\] (20)
\[
v_0^0 = [v_0(1), \ldots, v_0(N)] \in \mathbb{R}^N,
\] (21)
\[
\Sigma^0(x) = \begin{bmatrix} K^0(x) \\ k^0(x, x') \end{bmatrix} \in \mathbb{R}^{(N+1) \times (N+1)},
\] (22)
\[
K^0(x) = \begin{bmatrix} k(x_1^0, x_1^0) & \ldots & k(x_1^0, x_N^0) \\ \vdots & \ddots & \vdots \\ k(x_N^0, x_1^0) & \ldots & k(x_N^0, x_N^0) \end{bmatrix} \in \mathbb{R}^{N \times N},
\] (23)
\[
k^0(x, x') = \begin{bmatrix} k(x_1^0, x) \\ \vdots \\ k(x_N^0, x) \end{bmatrix} \in \mathbb{R}^N,
\] (24)

where \( v_0(\cdot) \) and \( k(\cdot, \cdot) \) are known functions (design parameters). Then completing the square of (19) for \( v(x) \) leads to the probability distribution \( p(v(x)|v_1^0, \ldots, v_N^0) \) as
\[
p(v(x)|v_1^0, \ldots, v_N^0) = \mathcal{N}(v(x)|\hat{v}(x), k(x, x')).
\] (25)

Here,
\[
\hat{v}(x) = v_0(x) + k^0(x, x')^T K^{-1}(v^0 - v_0^0),
\] (26)
\[
k(x, x') = k(x, x') - k_0(x, x')^T K^{-1} k(x, x'),
\] (27)

Thus the estimation of the unknown function \( v(x) \) in (9) based on Gaussian process regression is given by the output of the Gaussian distribution (25) with (26) and (27). Practically, the mean function \( \hat{v}(x) \) in (26) is used as the representative of the estimation of \( v(x) \).

### 3.2 Gaussian Process Regression with Equality Constraints

This section aims to give an approach to cope with the equality constraints (4) by using Gaussian process regression according to [19],[20]. In Gaussian process regression with equality constraints with respect to its derivative, the input-output models
\[
v = v(x),
\] (28)
\[
w = w(x) = \frac{\partial v(x)}{\partial x}
\] (29)

are used to estimate an unknown function \( v(x) \) and its derivative \( w(x) \) in (28) and (29), where \( x = [x_1, \ldots, x_n] \in \mathbb{R}^n \) is a state variable. The signals \( v \in \mathbb{R} \) and \( w \in \mathbb{R}^n \) in the left hand sides of (28) and (29) are the output variables of those functions. It is known that if the unknown function \( v(x) \) in (28) and (29) follows a Gaussian process (15), then its derivative \( \frac{\partial v(x)}{\partial x} \) also follows a Gaussian process [20]. The derivative \( \frac{\partial v(x)}{\partial x} \) is distributed according to an unknown probability \( p(\frac{\partial v(x)}{\partial x}) \), namely,
\[
\frac{\partial v(x)}{\partial x} \sim \mathcal{N}(\frac{\partial v_0(x)}{\partial x}) \in \mathbb{R}^n.
\] (30)

The mean function of the derivative \( \frac{\partial v(x)}{\partial x} \), the covariance function of the derivative \( \frac{\partial v(x)}{\partial x} \) and \( v(x') \), and the covariance function of the derivatives \( \frac{\partial v(x)}{\partial x} \) and \( \frac{\partial v(x')}{\partial x'} \) satisfy
\[
E \left[ \frac{\partial v(x)}{\partial x} \right] = \frac{\partial v_0(x)}{\partial x} \in \mathbb{R}^n,
\] (31)
\[
E \left[ \left( \frac{\partial v(x)}{\partial x} - \frac{\partial v_0(x)}{\partial x} \right)^T \left( \frac{\partial v(x')}{\partial x'} - \frac{\partial v_0(x')}{\partial x'} \right) \right] = \frac{\partial^2 v(x, x')}{\partial x \partial x'} \in \mathbb{R}^{n \times n},
\] (32)
\[
E \left[ \frac{\partial v(x)}{\partial x} - \frac{\partial v_0(x)}{\partial x} \right]^T \frac{\partial v(x')}{\partial x'} = \frac{\partial^2 v(x, x')}{\partial x \partial x'} \in \mathbb{R}^{n \times n},
\] (33)

due to the definitions (13) and (14). Using the functions (31) and (33), the Gaussian process of \( \frac{\partial v(x)}{\partial x} \) is described as
\[
\frac{\partial v(x)}{\partial x} \sim \mathcal{GP}\left( \frac{\partial v_0(x)}{\partial x}, \frac{\partial^2 v(x, x')}{\partial x \partial x'} \right).
\] (34)

In the following, the function
\[
v_0(x) = x^TPx
\] (35)
is used as the mean function \( \mu(x) \) where the matrix \( P \) satisfies (8). In the proposed method, a training data set \( \mathcal{D}^{(ec)} = \{ (x_1^{(ec)}, w_1^{(ec)}), (x_2^{(ec)}, v_2^{(ec)}) \} \) satisfying the model (28) and (29) as
\[
\begin{align*}
  v^{(ec)} & = v(x^{(ec)}), \\
  w^{(ec)} & = w(x^{(ec)}) = \frac{\partial \mu(x)}{\partial x} |_{x=x^{(ec)}}
\end{align*}
\]  
(36) 
(37)

is determined by
\[
\begin{align*}
x_1^{(ec)} & = 0, \quad w_1^{(ec)} = w(x_1^{(ec)}) = \frac{\partial \mu(x)}{\partial x} |_{x=x_1^{(ec)}} = 0, \\
x_2^{(ec)} & = 0, \quad v_2^{(ec)} = v(x_2^{(ec)}) = 0,
\end{align*}
\]  
(38) 
(39)

which are equivalent to the condition (4). Here the superscript \((\cdot)^{(ec)}\) stands for equality constraints. The symbol \( \mathcal{D}^{(ec)} \) is introduced for simplicity of notation. Assuming the Gaussian processes (15) and (34), the joint probability of a new output \( \nu(x) \) corresponding to an arbitrary state variable and the output data \( w_1^{(ec)}, v_2^{(ec)} \) in (38) and (39) is given by a multivariate Gaussian distribution according to Definition 1 by
\[
p = N\left( \begin{bmatrix} w_1^{(ec)} \\ v_2^{(ec)} \end{bmatrix} | \begin{bmatrix} w_0(x_1^{(ec)}) \\ w_0(x_2^{(ec)}) \end{bmatrix}, \Sigma^{(ec)}(x) \right)
\]  
(40)

Here,
\[
\begin{align*}
  w_0(x) & = \frac{\partial \mu(x)}{\partial x} |_{x=x_1^{(ec)}} \in \mathbb{R}^n, \\
  \Sigma^{(ec)}(x) & = \begin{bmatrix} \Sigma^{(ec)}(x) \\ \Sigma^{(ec)}(x) \end{bmatrix} = \mathcal{P}(x^{(ec)}), \\
  K^{(ec)}(x) & = \begin{bmatrix} k^{(ec)}(x, x_1^{(ec)}) \\ k^{(ec)}(x, x_2^{(ec)}) \end{bmatrix} \in \mathbb{R}^{(n+2)(n+2)}, \\
  k^{(ec)}(x) & = \begin{bmatrix} \frac{\partial k(x, x_1^{(ec)})}{\partial x} |_{x=x_1^{(ec)}} \\ \frac{\partial k(x, x_2^{(ec)})}{\partial x} |_{x=x_2^{(ec)}} \end{bmatrix} k(x_2^{(ec)}, x_2^{(ec)}) \in \mathbb{R}^{(n+1)(n+1)}
\end{align*}
\]  
(41) 
(42) 
(43) 
(44)

where \( \mu(x) \) and \( k(\cdot, \cdot) \) are known functions. Completing the square of (40) for \( \nu(x) \) leads to the probability distribution of \( \nu(x) | w_1^{(ec)}, v_2^{(ec)} \) as
\[
p = N(\nu(x) | \mu^{(ec)}(x), k^{(ec)}(x)).
\]  
(45)

Here, \( \mu^{(ec)}(x) \) and \( k^{(ec)}(x) \) are calculated according to Eqs. (26) and (27) as
\[
\begin{bmatrix} \mu^{(ec)}(x) \\ k^{(ec)}(x) \end{bmatrix} = \begin{bmatrix} \nu(x) + k^{(ec)}(x) \end{bmatrix}^{-1} \begin{bmatrix} w^{(ec)}(x) \nu(x) v_0^{(ec)}(x) \end{bmatrix}
\]  
(46) 
(47)

Hence the expectation of the unknown function \( \nu(x) \) based on Gaussian process regression with equality constraints is \( \nu^{(ec)}(x) \) in (46). This result is obtained by using Eq. (4) as equality constraints for Gaussian process regression. The estimation (46) satisfies the equality constraints. In Section 3.3, the functions in (46) and (47) are further used for Gaussian process regression with inequality constraints.

3.3 Gaussian Process Regression with Inequality Constraints

The purpose of this section is to provide a way to incorporate the inequality constraint (3) into Gaussian process regression based on [19],[20]. Gaussian process regression with inequality constraints estimates the unknown function \( \nu(x) \) in (28) by requiring the output variable \( w \) of the model (29) to satisfy the inequality (3) for \( x \), namely,
\[
w^T A(x) w + w^T b(x) + c(x) < 0.
\]  
(48)

For a given \( x \), a region \( C(x) \) of \( w \) satisfying the above constraint is determined, whereas a fixed \( w^{(ec)}(x) \) is determined for a given \( x^{(ec)} \) in the equality constraint. It is not so easy to cope with a pair of a point \( x \) and the corresponding set \( C(x) \) in Gaussian process regression, sampled points \( (x_i^{(ic)}), (w_i^{(ic)}) \) \( (i = 1, \ldots, L) \) are selected to represent the sets \( (x_i^{(ic)}, C(x_i^{(ic)})) \)'s. Here the superscript \((\cdot)^{(ic)}\) stands for the inequality constraints. The output variable \( w_i^{(ic)} \) to represent the set \( C(x_i^{(ic)}) \) will be determined later. The training data thus chosen is expressed as \( \mathcal{D}^{(ic)} = \{ (x_i^{(ic)}, w_i^{(ic)}) \} | C(x_i^{(ic)}) \), \( i = 1, \ldots, L \). It satisfies
\[
\nu_i^{(ic)}(x) = \nu_i^{(ic)}(x) = \frac{\partial \nu(x)}{\partial x} |_{x=x_i^{(ic)}} \in \mathbb{R}^n, \\
C(x_i^{(ic)}) = \begin{bmatrix} \nu_i^{(ic)} \\ \nu_i^{(ic)} \end{bmatrix} F(\nu_i^{(ic)}) < 0, \quad i = 1, \ldots, L, \\
F(\nu_i^{(ic)}) = \begin{bmatrix} \frac{\partial \nu(x)}{\partial x} |_{x=x_i^{(ic)}} A(x_i^{(ic)}) \frac{\partial \nu(x)}{\partial x} |_{x=x_i^{(ic)}} \\ + \frac{\partial \nu(x)}{\partial x} |_{x=x_i^{(ic)}} b(x_i^{(ic)}) + c(x_i^{(ic)}) \end{bmatrix} \\
= w_i^{(ic)} A(x_i^{(ic)}) + w_i^{(ic)} b(x_i^{(ic)}) + c(x_i^{(ic)}).
\]  
(49) 
(50) 
(51)

Here it is assumed that \( w_i^{(ic)} = \nu_i^{(ic)}(x_i) \) is uniquely determined for \( x_i^{(ic)} \). In the following, \( \nu^{(ic)}(x) \) in (46) and \( k^{(ic)}(x) \) in (47) are used as the mean function and the covariance one, respectively, for a new Gaussian process to cope with the additional inequality constraints in (48). Let us assume that the function \( \nu(x) \) follows a Gaussian process with the mean function and covariance one expressed as
\[
\nu(x) \sim GP(\nu^{(ic)}(x), k^{(ic)}(x, x'))
\]  
(52)
\[
\frac{\partial^2 k^{(ic)}(x, x')}{\partial x \partial x'} = \frac{\partial k^{(ic)}(x, x')}{\partial x} \cdot \frac{\partial k^{(ic)}(x, x')}{\partial x'} \cdot \frac{\partial^2 k^{(ic)}(x, x')}{\partial x \partial x'}. \quad (53)
\]

Where the joint probability \( p(x) \) is used to obtain the numerical solution of the Hamilton-Jacobi inequality (3) and the training data set \( \tilde{v}^{(ic)}(x) = x^T P x \), where \( k^{(ic)}(x, x') \) is the covariance function of the training data set \( (ic) \) as

\[
k^{(ic)}(x, x') = k(x, x') - k^{(ic)}(x)^T K^{(ic)-1} k^{(ic)}(x'). \quad (55)
\]

According to Definition 1, the joint probability of \( v(x) \) and \( w^{(ic)} \) of the training data set \( D^{(ic)} \) is defined as

\[
\begin{align*}
\mathbb{E} \left[ \tilde{w}^{(ic)} \right] & = \frac{1}{S} \sum_{s=1}^{S} \tilde{w}^{(ic)}_s, \\
\tilde{w}^{(ic)}_s & = \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}(x). \\
\tilde{w}^{(ic)}(x) & = \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}(x). \\
\end{align*}
\]

(67)

where \( S \) is a number of samples and \( \tilde{w}^{(ic)}(x) \) denotes a \( s \)-th sample from the distribution as

\[
\tilde{w}^{(ic)}(x) = \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_s. \quad (68)
\]

By using (67), the function \( \tilde{v}(x) \) is regarded as the numerical solution of the proposed method. The feedback controller is given as \( u(x) = - (1/2) K(x) \partial v(x) K(x) \partial x \), where \( K(x) \) is a hyper-parameter which the designers can determine arbitrarily. If no prior knowledge on the covariance function \( k^{(ic)}(x, x') \) is available, the variance of the predictive distribution \( \gamma_0^2 \) is selected to be a constant. This selection means that the mean function has the same level of confidence at any point \( x \).

The mean function of the proposed method, however, is obtained by using the linearized system at the origin, so the accuracy of the approximation depends on the magnitude \( |x| \).

This is the reason why it is better to avoid using the stationary covariance function (72) in the proposed method.

On the other hand, when a function \( k^{(ic)}(x, x') \) can be used as a covariance function, it is proved that the following function can also be used as a covariance [15].

\[
k^{(ic)}(x, x') = \gamma \exp \left\{ - \frac{1}{2} (x - x')^T \Gamma^{-1} (x - x') \right\} \quad (72)
\]

which is a stationary covariance function in the sense that the coefficients \( \gamma_0^2 \) and \( \Gamma \) is constant. Here, \( \Gamma = \text{diag}(\gamma_1^2, \ldots, \gamma_n^2) \), and \( \gamma_0 \) is hyper-parameters which the designers can determine arbitrarily. If no prior knowledge on the covariance function is available, the variance of the predictive distribution \( \gamma_0^2 \) is selected to be a constant. This selection means that the mean function has the same level of confidence at any point \( x \). The mean function of the proposed method, however, is obtained by using the linearized system at the origin, so the accuracy of the approximation depends on the magnitude \( |x| \).

This is the reason why it is better to avoid using the stationary covariance function (72) in the proposed method.
\[ k(x, x') = f(x)k_1(x, x')f(x') \]  

(73)

Here \( f(\cdot) \) is an arbitrary function. Now in our research, we define \( f(x) \) and \( k_1(x, x') \) as \( (x^T x + \kappa^2)^d \) and \( (x - x')^d \), respectively. Consequently the following nonstationary covariance function is adopted:

\[
 k(x, x') = (x^T x + \kappa^2)^d (x^T x' + \kappa^2)^d 
\times \gamma^2 \exp \left( -\frac{1}{2} (x - x')^T \Gamma^{-1} (x - x') \right),
\]  

(74)

where \( \kappa \) and \( J \geq 0 \) are constant hyper-parameters which the designers determine. The covariance function in (74) describes that the confidence of the mean function depends on the state variables \( x \) and \( x' \). Thus the covariance function in (74) is suitable for the problems in the proposed method.

### 3.5 Optimization of Hyper-Parameters

Since the choice of the hyper-parameters has a certain influence on the predictive distributions of the Gaussian processes, it is desirable to find a guideline to decide them automatically. A popular method used for determining hyper-parameters is to maximize the logarithm of a likelihood function \( p(D|\theta) \) [15]. Here, \( D \) is a training data set and \( \theta \) is a set of hyper-parameters of a Gaussian process model. However, the integration term including inequality constraints (63) prevents us from using conventional methods to determine hyper-parameters. Now, we maximize the following cost function \( J(\theta) \) given by

\[
 J(\theta) := \int_{\mathbb{R}^{n+1} \in \bar{C}} N \left( \tilde{w}^{(ic)}, K^{(ic)}(\theta) \right) dw^{(ic)} 
= \alpha(\theta).
\]  

(75)

Here, \( \alpha(\theta) \) is the same as in (63). The effect of maximizing the above function is to enlarge the area where \( \alpha(\theta) \) satisfies the Hamilton-Jacobi inequality (3). The gradient of (75) is given by

\[
 \frac{\partial J}{\partial \theta_j}(\theta) = -\frac{\alpha(\theta)}{2} \left( \gamma \left[ K^{(ic)} - \frac{\partial K^{(ic)}}{\partial \theta_j} \right] \right) 
\]  

\[
 - \mathbb{E} \left[ \left( \tilde{w}^{(ic)} - \tilde{w}^{(ic)} \right)^T K^{(ic)} - \frac{\partial K^{(ic)}}{\partial \theta_j} K^{(ic)} - \frac{\partial K^{(ic)}}{\partial \theta_j} \left( \tilde{w}^{(ic)} - \tilde{w}^{(ic)} \right) \right].
\]  

(76)

Because it is not possible to compute \( \alpha(\theta) \) analytically, we use

\[
 \alpha(\theta) \approx \frac{S}{S_a}.
\]  

(77)

Here \( S \) is the same as in (67) and \( S_a \) is a total number of samplings. The sampling in (68) is performed by rejection sampling where a sampled \( \tilde{w}^{(ic)} \) is rejected if it is not in \( \bar{C} \). The sampling procedure is repeated until \( S \) accepted samples of \( \tilde{w}^{(ic)} \) are obtained. The validity of the approximation (77) is discussed in Appendix B. Then we employ the steepest descent method expressed as

\[
 \theta_{r+1} = \theta_r + \eta \frac{\partial J}{\partial \theta_j}(\theta_j),
\]  

(78)

where \( r \) is the number of steps and \( \eta \) is the step size. The procedure of the proposed method is summarized in Algorithm 1. The computational cost of this algorithm is mainly due to Step 5 in Algorithm 1. It is \( O(n^4 L^4 S_a) \), composed of calculating the inverse matrix of \( K^{(ic)} \in \mathbb{R}^{n \times n} \) in (60) and the total number of samplings \( S_a \) in (77).

### 4. Numerical Example

In this section, we verify the effectiveness of the proposed method by using a numerical example.

#### 4.1 Problem Setting

Consider a nonlinear \( H_\infty \) control problem of the nonlinear system (1) with \( x = [x_1, x_2]^T \). The desired \( L_2 \) gain of this system is \( \gamma = 10 \). Here, \( f(x) = [x_2, -(30x_1 + 0.01x_1^2 + 0.1x_2)]^T \), \( g = g_0 = [0, 1]^T \), \( h(x) = [0, x_1]^T \), and \( j = [1, 0]^T \) are given. Then the Hamilton-Jacobi inequality of this system is expressed as

\[
 F(v(x)) \equiv \frac{1}{4} \frac{\partial v(x)}{\partial x} \left( \gamma \frac{1}{2} K^{(ic)} - g^T g \right) \frac{\partial v(x)}{\partial x}^T 
+ \frac{\partial v(x)}{\partial x} f(x) + h(x)^T h(x) < 0
\]  

(79)

using Eqs. (3)-(7).

#### 4.2 Results of Calculations

Firstly, a solution \( v_0(x) = x^T Px \) to the Hamilton-Jacobi inequality (79) of the linearized system and the corresponding function \( F(v_0(x)) \) are shown in Figs. 1 and 2. This solution is calculated by performing Step 1 in Algorithm 1. Figure 1 represents \( v_0(x) \). Figure 2 expresses the value of the function \( F(v_0(x)) \) in (79) obtained by using \( v_0(x) \). The shaded flat surface is the plane \( F(v(x)) = 0 \). This figure implies that when the value of \( F(v(x)) \) is negative (i.e., \( F(v(x)) \) is under the shaded surface in the figure), \( v(x) \) satisfies the Hamilton-Jacobi inequality.

![Fig. 1](image-url)  

**Fig. 1** Shape of the function \( v_0(x) \).
is because the approximate solution $v_0(x)$ is a solution to the linearized system. It is valid only in a small neighborhood of the origin.

Secondly, a solution $v_1(x)$ to (79) obtained by performing Steps 1-5, 10 and 11 of Algorithm 1 and the corresponding $F(v_1(x))$ are depicted in Figs. 3 and 4. The initial hyper-parameters in $\theta_0$ are selected as

$$\gamma_0 = 10, \gamma_1 = 20, \gamma_2 = 20, \kappa = 1.$$  

In Eq. (74), $\lambda = 1/4$ is determined by trial and error in order to avoid complexity of calculation in this problem. The number of samplings $S$ for acquiring the expectation $\mathbb{E}[W^{(i)}]$ in (67) is set as $S = 100$. Both the black dots plotted in the Fig. 3 and the white dots plotted in Fig. 4 describe the sampled vectors $x^{(i)}$. Compared with Fig. 2, the region satisfying the Hamilton-Jacobi inequality (79) around the origin in Fig. 4 is enlarged drastically.

Lastly, a solution $v_2(x)$ to (79) obtained by performing all the steps of Algorithm 1, where $\eta = 0.01$, and the corresponding function $F(v_2(x))$ are given in Figs. 5 and 6. The function $v_2(x)$ is expressed as in (69). Here Fig. 7 expresses the history of the cost function $J(\theta)$ in (75) with respect to the step number $r$ in the process of optimization. The optimized hyper-parameters are calculated in Steps 1-9 in Algorithm 1 as follows:

$$\gamma_0 = 4.6296, \gamma_1 = 20.148, \gamma_2 = 22.317, \kappa = 0.71414.$$  

In Fig. 7, the history of the cost function $J(\theta)$ is oscillatory instead of increasing monotonously. This is because it is approximated by the sampled data in (77). Figure 6 shows that the areas satisfying the Hamilton-Jacobi inequality (79) becomes much wider than those in Figs. 2 and 4.

5. Conclusion

This paper proposes numerical solutions to Hamilton-Jacobi inequalities by using Gaussian processes with the equality and inequality constraints. They are obtained by sampling the valued functions corresponding to inputs which satisfy the inequalities. Then, a nonstationary covariance function is used to cope with state dependent uncertainty of the estimation. Furthermore, the adjustable hyper-parameters are optimized to increase the accuracy of the estimation. Numerical examples by applying it to an $H_\infty$ control problem of a nonlinear system verify the effectiveness of the proposed method. It is remarkable
that the proposed method can be applied to other control problems, e.g., a Lyapunov inequality to ensure asymptotic stability of the system. It is in fact applicable to several control objectives with Lyapunov inequality constraints.

References


Appendix A Proof of Theorem 1

Proof Firstly, we prove the equation $\dot{v}(0) = 0$ without supposing that the constraint set $\mathcal{C}$ is convex. Substituting $x = 0$ for (69), we have

$$\dot{v}(0) = \varphi^{(ec)}(0) + k^{(ec)}(0)^T K^{(ec)} \left( \frac{1}{T} \sum_{i=1}^{T} w^{(ic)}_{k - \tilde{w}^{(ic)}} \right). \quad (A. 1)$$

The equation

$$\varphi^{(ec)}(0) = 0 \quad (A. 2)$$

follows from (46) obviously. Using (61), the equation

$$k^{(ec)}(0) = \begin{bmatrix} \frac{\partial k^{(ec)}(0, x')}{\partial x'}_{x' = x'^{ec}_1}, \ldots, \frac{\partial k^{(ec)}(0, x')}{\partial x'}_{x' = x'^{ec}_n} \end{bmatrix} \quad (A. 3)$$

follows. Substituting $x = 0$ for (55) leads to

$$k^{(ec)}(0, x') = k(0, x') - k^{(ec)}(0)^T K^{(ec)^{-1}} k^{(ec)}(x'). \quad (A. 4)$$

We put $k^{(ec)}(0)^T K^{(ec)^{-1}} = [a_1^T, a_2^T]$, where $a_1$ is a vector value and $a_2$ is a scalar value. Then multiplying the matrix $K^{(ec)}$ from the right hand side, we have

$$k^{(ec)}(0)^T = [a_1^T, a_2^T] K^{(ec)}. \quad (A. 5)$$

Using (43) and (44), we see that (A. 5) becomes

$$\begin{align*}
\frac{\partial k(0, x')}{\partial x'}_{x' = x'^{ec}_1} & = a_1^T \frac{\partial^2 k(x, x')}{\partial x \partial x'}_{x' = x'^{ec}_1} + a_2 \frac{\partial k(0, x')}{\partial x'}_{x' = x'^{ec}_1}, \\
\frac{\partial k(0, x')}{\partial x'}_{x' = x'^{ec}_n} & = a_n \frac{\partial k(0, x')}{\partial x'}_{x' = x'^{ec}_n} + a_2 k(0, 0),
\end{align*} \quad (A. 6)$$

since $x'^{ec}_n = 0$. Comparing both sides of equations in (A.6), we can assert that the solution to (A.6) is $[a_1^T, a_2^T] = [0^T, 1]$. Then substituting $k^{(ec)}(0)^T K^{(ec)^{-1}} = [0^T, 1]$ for (A.4) leads to

$$k^{(ec)}(0, x') = k(0, x') - [0^T, 1]^T k^{(ec)}(x'),$$

$$= k(0, x') - [0^T, 1] \frac{\partial k(x', x)}{\partial x}_{x' = x'^{ec}_1}, k(x', 0)^T$$

$$= k(0, x') - k(x', 0) = 0 \quad (A. 7)$$

since $k(x, x') = k(x', x)$. Substituting (A.7) for (A.3), we have

$$k^{(ec)}(0) = [0^T, \ldots, 0^T]^T. \quad (A. 8)$$

Hence

$$\dot{v}(0) = 0 \quad (A. 9)$$
follows from (A.1), (A.2) and (A.8). As for the equation \( \frac{\partial \tilde{v}(x)}{\partial x} \bigg|_{x=\alpha} = 0 \), it is possible to prove it by the similar procedure, and we omit the proof.

Next we prove the inequality (70) supposing that the constraint set \( \tilde{C} \) is convex. Using (69),

\[
\frac{\partial \tilde{v}(x)}{\partial x} \bigg|_{x=\alpha} = \frac{\partial \tilde{v}(\alpha)}{\partial x} + \frac{\partial k^{(ic)}(x)}{\partial x} \bigg|_{x=\alpha} K^{(ic)^{-1}} \left( \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i - \tilde{w}^{(ic)} \right)
\]

(A.10)

follows. Now we put \( \frac{\partial k^{(ic)}(x)}{\partial x} \bigg|_{x=\alpha} = [B_1, \ldots, B_L] K^{(ic)} \) where \( B_1, \ldots, B_L \in \mathbb{R}^{n \times n} \) are valued matrices. Then multiplying the matrix \( K^{(ic)^{-1}} \) from the right hand side, we have

\[
\frac{\partial k^{(ic)}(x)}{\partial x} \bigg|_{x=\alpha} = [B_1, \ldots, B_L] K^{(ic)}.
\]

(A.11)

Using (60) and (61), we can assert that \( l \)-th element of (A.11) becomes

\[
\frac{\partial^2 k^{(ic)}(x, x')}{\partial x \partial x'} \bigg|_{(x, x')=(\tilde{x}^{(ic)}_i, \tilde{x}^{(ic)}_j)} = \sum_{j=1}^{L} B_j \frac{\partial^2 k^{(ic)}(x, x')}{\partial x \partial x'} \bigg|_{(x, x')=(\tilde{x}^{(ic)}_i, \tilde{x}^{(ic)}_j)}.
\]

(A.12)

Comparing both sides of (A.12), we deduce that the solution to (A.12) is

\[
[B_1, \ldots, B_{l-1}, B_l, B_{l+1}, \ldots, B_L] = [O, O, I, O, \ldots, O].
\]

(A.13)

Here \( O \) describes a zero matrix, and \( I_l \) expresses the \( l \)-th unit matrix. Then substituting \( \frac{\partial k^{(ic)}(x)}{\partial x} \bigg|_{x=\alpha} = [O, O, I, O, \ldots, O] \) for (66) leads to

\[
\frac{\partial \tilde{v}(x)}{\partial x} \bigg|_{x=\alpha} = \frac{\partial \tilde{v}(\alpha)}{\partial x} + [O, O, I, O, \ldots, O] \left( \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i - \tilde{w}^{(ic)} \right)
\]

\[
= \frac{\partial \tilde{v}(\alpha)}{\partial x} + \left( \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i \right) - \frac{\partial \tilde{v}(\alpha)}{\partial x} \bigg|_{x=\alpha} K^{(ic)^{-1}} \left( \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i - \tilde{w}^{(ic)} \right)
\]

\[
= \left( \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i \right) \cdot \tilde{C}.
\]

(A.14)

Hence

\[
\left[ \frac{\partial \tilde{v}(x)}{\partial x} \bigg|_{x=\alpha} \right]^{T} = \frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i \cdot \tilde{C}.
\]

(A.15)

follows from (A.14). Since the constraint set \( \tilde{C} \) is convex,

\[
\frac{1}{S} \sum_{i=1}^{S} \tilde{w}^{(ic)}_i \in \tilde{C}
\]

(A.16)

follows. Therefore, the inequality (70) follows from (A.15), (A.16), and the definition of \( \tilde{C} \) in (62).

\[
\text{Appendix B The Validity of the Approximation in (77)}
\]

Consider the situation of sampling \( \tilde{w}^{(ic)}_s \) given by

\[
\tilde{w}^{(ic)}_s \sim \mathcal{N} \left( \tilde{w}^{(ic)}_s \left| \tilde{w}^{(ic)}_s, K^{(ic)} \right. \right), \quad s = 1, \ldots, S_a,
\]

(B.1)

\[
\begin{cases}
\tilde{w}^{(ic)}_s \in \tilde{C}, \quad s = 1, \ldots, S,
\tilde{w}^{(ic)}_s \not\in \tilde{C}, \quad s = S + 1, \ldots, S_a.
\end{cases}
\]

(B.2)

Here, \( \alpha \) is given in (63), namely,

\[
\alpha := \int_{w^{(ic)} \in \tilde{C}} \mathcal{N} \left( w^{(ic)} \left| \tilde{w}^{(ic)}, K^{(ic)} \right. \right) \, dw^{(ic)}.
\]

(B.3)

Then, from (B.3), the probability \( p \left( w^{(ic)} \in \tilde{C} \right) \) and \( p \left( w^{(ic)} \not\in \tilde{C} \right) \) are given by

\[
p \left( w^{(ic)} \in \tilde{C} \right) = \int_{w^{(ic)} \in \tilde{C}} \mathcal{N} \left( w^{(ic)} \left| \tilde{w}^{(ic)}, K^{(ic)} \right. \right) \, dw^{(ic)} = \alpha,
\]

(B.4)

\[
p \left( w^{(ic)} \not\in \tilde{C} \right) = \int_{w^{(ic)} \not\in \tilde{C}} \mathcal{N} \left( w^{(ic)} \left| \tilde{w}^{(ic)}, K^{(ic)} \right. \right) \, dw^{(ic)} = 1 - \alpha.
\]

(B.5)

Assuming that \( \tilde{w}^{(ic)}_s \) is sampled independently by (B.1), we calculate the joint probability of \( \tilde{w}^{(ic)}_s \) as

\[
p \left( \tilde{w}^{(ic)}_1, \ldots, \tilde{w}^{(ic)}_s \in \tilde{C}, \tilde{w}^{(ic)}_{s+1}, \ldots, \tilde{w}^{(ic)}_S \not\in \tilde{C} \right) = \prod_{i=1}^{S_a} p \left( \tilde{w}^{(ic)}_i \in \tilde{C} \right) \prod_{i=S+1}^{S_a} p \left( \tilde{w}^{(ic)}_i \not\in \tilde{C} \right)
\]

\[
= \left( \prod_{i=1}^{S_a} \alpha \right) \left( \prod_{i=S+1}^{S_a} (1 - \alpha) \right) = \alpha^S (1 - \alpha)^{S_a - S}
\]

(B.6)

by using (B.4) and (B.5). In order to maximize the probability in (B.6), by taking the logarithm of (B.6),

\[
\ln p \left( \tilde{w}^{(ic)}_1, \ldots, \tilde{w}^{(ic)}_S \in \tilde{C}, \tilde{w}^{(ic)}_{S+1}, \ldots, \tilde{w}^{(ic)}_S \not\in \tilde{C} \right) = S \ln \alpha + (S_a - S) \ln(1 - \alpha)
\]

(B.7)

holds. Differentiating (B.7) for \( \alpha \), we see that

\[
\frac{dp}{d\alpha} \left( \tilde{w}^{(ic)}_1, \ldots, \tilde{w}^{(ic)}_S \in \tilde{C}, \tilde{w}^{(ic)}_{S+1}, \ldots, \tilde{w}^{(ic)}_S \not\in \tilde{C} \right) = \frac{S}{\alpha} - \frac{S_a - S}{1 - \alpha} = \frac{S_a - S}{\alpha(1 - \alpha)} = 0.
\]

(B.8)

Solving (B.8) for \( \alpha \), we obtain

\[
\alpha = \frac{S_a}{S_a - S}.
\]

(B.9)

This value of \( \alpha \) in (B.9) maximizes the probability that \( S \) samples are accepted and \( (S_a - S) \) samples are rejected, namely, the situation in (B.1) and (B.2) is most likely caused with the value in (B.9). This implies that the approximation \( \alpha(\theta) \) in (77) is valid.
Kenji Fujimoto (Member)

He received the B.Sc. and M.Sc. 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.

Hirofumi Beppu

He received his B.S. degree in Engineering from Kyoto University, Japan, in 2017. He is currently an M.S. student of Graduate School of Engineering, Kyoto University, Japan. His research interests include control theory and machine learning.

Yuji Takaki

He received his B.S. and M.S. degrees in Engineering from Kyoto University, Japan, in 2015 and 2017 respectively. He is currently an engineer at Mitsubishi Heavy Industries, Ltd. His research interests include stochastic control theory.