BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Artificial Intelligence Advances
https://ojs.bilpublishing.com/index.php/aia

ARTICLE

Safety-critical Policy Iteration Algorithm for Control under Model Uncertainty

Navid Moshtaghi Yazdani[1*] Reihaneh Kardehi Moghaddam[1] Mohammad Hasan Olyaei[2]

1. Department of Electrical Engineering, Mashhad branch, Islamic Azad University, Mashhad, Iran
2. Department of Electrical Engineering, Sadjad University of Technology, Mashhad, Iran

ARTICLE INFO

Article history
Received: 18 January 2022
Revised: 31 March 2022
Accepted: 2 April 2022
Published: 11 April 2022

Keywords:
Safe-critical
Optimal controller
Reinforcement learning
Lyapunov
Sum-of-Square

ABSTRACT

Safety is an important aim in designing safe-critical systems. To design such systems, many policy iterative algorithms are introduced to find safe optimal controllers. Due to the fact that in most practical systems, finding accurate information from the system is rather impossible, a new online training method is presented in this paper to perform an iterative reinforcement learning based algorithm using real data instead of identifying system dynamics. Also, in this paper the impact of model uncertainty is examined on control Lyapunov functions (CLF) and control barrier functions (CBF) dynamic limitations. The Sum of Square program is used to iteratively find an optimal safe control solution. The simulation results which are applied on a quarter car model show the efficiency of the proposed method in the fields of optimality and robustness.

1. Introduction

Safety is an integral part and a central requirement for any safe-critical system such as power systems, automatic devices, industrial robots, and chemical reactors. Considering the increasing demand for safe systems in the future generation of industrial systems, and also the importance of an interaction with systems surroundings and uncertainties, there is a real need for the development of safe controllers, which can meet the already-mentioned demand. In the absence or violation of these safety conditions, the system is likely to suffer from some faults, including the system stabilization problem and its simultaneous survival in the given safety system, which lead to the rise of multiple serious challenges to designing controllers. The optimal control design, as well as the safe control design for the feedback state, is discussed separately in the literature review. Developing such safe controllers to optimize the performance of dynamic systems with uncertainties, primarily resulted from lack of safe optimal controllers with uncertainty conditions.

*Corresponding Author:
Navid Moshtaghi Yazdani,
Department of Electrical Engineering, Mashhad branch, Islamic Azad University, Mashhad, Iran;
Email: navid.moshtaghi@ut.ac.ir

DOI: https://doi.org/10.30564/aia.v4i1.4361

1.1 Related Works

The official design for the stabilization of non-linear dynamic systems is often obtained by employing the Control Lyapunov Functions (CLFs). The optimal feedback controllers necessary for general non-linear systems can be designed by solving Hamilton-Jacobi-Bellman equations (HJB), which have been done approximately by through the use of Galerkin method [1] and neural networks method [2-8]. However, due to the lack of robustness and computational infeasibility for online performance, the open-loop form of calculating these solutions seems problematic. Consequently, in this paper the optimal control of constrained systems equipped with penalty functions in the performance function [9]. However, the application of these methods is only limited to linear state constraints.

Today real-time safety in dynamic systems has gained large attention, followed by the introduction of the barrier functions, through which the risk of the system states entering the given non-safety zones can be removed [10-15]. Also, control methods using CLF and CBF have been considered as successful methods to achieve safety-stability control. Some researchers have shown that for the performance of movement tasks (manipulation and locomotion), CLF-based quadratic programs (CLF-QP) with constraints can be solved online [16,17]. They have also combined CBFs with CLF-QP to effectively for the effective management of safety constraints in real time. By the by, an itemized information on the system model is expected for every one of these CLF-based and CBF-based techniques.

Taylor et al. addressed how a minimization method for experimental risk can lead to the uncertainties in CLF and CBF constraints [18,19]. Westernbroek et al. have additionally proposed a reinforcement learning-based method to learn model uncertainty compensation for the input-output linearization control [20]. Learning-based control is also obtained in dynamic systems with high uncertainty in spite of safety constraints [21,22]. Moreover, probabilistic models such as Gaussian process can be used to learn about model uncertainties [23,24]. Using these methods, the comprehensive investigation of the learned model or policy is permitted; however, they can scale inadequately with state dimension and involving them in high-ordered systems won't be simple.

1.2 Contributions and Outline

Prajna et al. introduced a policy iteration algorithm as a way to build the safe optimal controller for a class of certain nonlinear systems [25]. However, due to the difficulty of practically obtaining accurate system information, an online training method is presented in this study to replace identifying system dynamics with an iterative algorithm featured with real data. In this paper, the effect of model uncertainty is, also, investigated on CLF and CBF dynamic constraints. For each of them, the purpose of the RL agent and the policy to be learned will be defined. The Sum-of-Square program is utilized to iteratively discover an optimal safe control solution. Finally, in order for the efficiency of the proposed method to be validated, a simulation example is employed.

The remaining part of the present paper is organized as follows: Section 2 formulates the problem and presents a new safe optimal control framework. Section 3 presents reinforcement learning for optimal safe control under uncertain dynamics, and Section 4 provides the numerical examples to validate the efficiency of the proposed method.

1.3 Notations

The term C^1 denotes the set of all continuous differential functions. Then, P denotes the set of all existing functions in C^1 that are positive, definite and proper. The polynomial $p(x)$ is Sum-of-Squares (SOS) (i.e., $p(x) \in \mathrm{P}^{SOS}$ in which P^{SOS} is a set of SOS polynomials, $p(x) = \sum_1^m p_i^2(x)$ where $p_i(x) \in \mathrm{P}\ i = 1,...,m$). Function $\mathrm{K}: \mathrm{R}^n \to \mathrm{R}^n$ is an extended class K function and $K(0) = 0$. ∇V Alludes to the gradient of the V function: $\mathrm{R}^n \to \mathrm{R}^n$.The Li derivative of function h with respect to f is defined as $L_f V(x) = \frac{\partial h}{\partial x} f(x)$. For any positive integer t1 and t2 where t2≥t1, $\bar{\mathrm{n}}_{t1,t2}(\mathrm{x})$ is the vector of all distinct monic monomial sets $\binom{m+t_2}{t_2} - \binom{m+t_1-1}{t_1-1}$ in $x \in \mathrm{R}^n$ with minimum degree of t_1 and maximum degree of t_2. Moreover, $\mathrm{R}[x]_{t_1,t_2}$ represents a set of all polynomials in $x \in \mathrm{R}^n$ with degrees less than t_2 and greater than t_1.

2. Problem Formulation and Details

In this part, we talk about safety, stability and optimization of the control systems. The initial results of each are also mentioned. Then the formulas of the optimal safe control design will be performed.

2.1 Optimal Control of Dynamical Systems

Consider the following nonlinear system:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

In which $\mathrm{x} \in \mathrm{R}^n$ is the system state vector, $\mathrm{u} \in \mathrm{R}^m$ is the control input vector, $\mathrm{f}: \mathrm{R}^n \to \mathrm{R}^n$ and $\mathrm{g}: \mathrm{R}^n \to \mathrm{R}^{n \times m}$ are both locally Lipschitz continuous with $f(0) = 0$. We expect the system as a stabilizable one.

The main goal of standard optimal control design is to

find a control policy to minimize the predefined performance index over the system trajectories (1) defined as follows:

$$J(x_0,u)=\int_0^{\infty} r(x(t),u(t))dt \tag{2}$$

In relation (2), $r(x,u)=q(x)+u^T Ru$, $q(x)$ and $R(x)$ can be considered as reward function, positive definite function and positive definite matrix, respectively. The reward function $r(x,u)$ is defined such that optimizing (2) guarantees the achievement of control objectives (e.g., minimizing the control effort to achieve the desired transient response) as well as system stability.

The presence of an optimal stabilizing solution is ensured under mild assumptions about the reward function and system dynamics [26].

Assumption 1. Considering system (1), there exists a Lyapunov function $V\in\mathcal{P}$ and a feedback control policy u which satisfies the following inequality:

$$\mathrm{L}(V,u)=-(L_fV(x)+L_gV(x)u)-r(x,u)\geq 0 \quad x\in \mathrm{R}^n \tag{3}$$

The system stability conditions are guaranteed by this assumption, implying that the cost $\forall x_0\in R^n, J(x_0,u)$ is Finite.

Theorem 1. Theorem 10.1.2 [26] considers system (1) with performance function (2), there must be a positive semi-definite function $V^*(x)\in C^1$ satisfying the Hamilton-Jacobi-Belman (HJB) equation as follows:

$\mathrm{H}(V^*)=0$

In which

$$\mathrm{H}(V)=q(x)+L_fV(x)-\frac{1}{4}L_gV(x)R^{-1}(x)(L_gV(x))^T=0,\quad V(0)=0 \tag{4}$$

Therefore, the following feedback control

$$u^*(x)=\frac{1}{2}R^{-1}(x)(L_gV^*)^T(\mathrm{x}) \tag{5}$$

Optimizes the performance index (2) and results in the achievement of asymptotic stability of the equilibrium $x=0$. Also, the optimal value function is given as follows:

$$V^*(x_0)=\min_u J(x_0,u)=J(x_0,u^*),\quad \forall x_0\in \mathrm{R}^n \tag{6}$$

Assumption 1 appears that it is vital to solve the HJB Equation (4) to find an optimal control solution.

Assumption 2: There are proper mappings $V_0:\mathrm{R}^n\to\mathrm{R}$ and $\mathrm{u}:\mathrm{R}^n\to\mathrm{R}^m$, such that $V_0\in\mathrm{R}[x]_{2,2r}\cap\mathrm{P}$ and $\mathrm{L}(V_0,u)$ are SOS.

2.2 About Control Barrier Functions and Its Relation with Safe Control of Dynamical Systems

In a safety-critical system, it is important to prevent its state starting from any initial conditions in X_0 set to enter some special unsafe regions like $X_u\in X$. To design a safe controller, control barrier functions (CBF), inspired by Control Lyapunov Function (CLF), can be employed. Now Equation (1) and the function $h:R^n\to R$ can be considered as follows:

$$\begin{aligned} &h(\mathrm{x})\geq 0,\ \forall x\in \mathrm{X}_0,\\ &h(\mathrm{x})<0,\ \forall x\in \mathrm{X}_u \end{aligned} \tag{7}$$

The following function is also defined as:

$$\mathrm{L}=\{\mathrm{x}\in \mathrm{X}\mid h(\mathrm{x})\geq 0\} \tag{8}$$

Having ZCBF $h(x)$, the admissible control space $S(x)$ is defined as follows:

$$S(x)=\{u\in U\mid L_fh(x)+L_gh(x)u+K_b(h(x))\geq 0\},x\in X \tag{9}$$

The following theorem demonstrates the way a controller is designed using the ZCBF concept to ensure that the forward invariance of the safe set and system stability.

Theorem 2. For $\mathrm{L}\subset R^n$ given in (8) and a ZCBF defined by h in (9), each controller $\mathrm{u}\in\mathrm{S}(\mathrm{x})$ for the system (1) presents a safe set L forward invariant.

The barrier functions for exponential controls are introduced. They are improved in a work by Ams et al. [27,28].

This translates to the r^{th} time-derivative of $h(x)$

$$h^{(r)}(x\,u)=L_f^r h(x)+L_gL_f^{r-1}h(x)u$$

The authors expanded the CBFs having an arbitrary relative degree $r\geq 1$ to $h(x)$ functions. To do so, we define $z=col\left(h(x),L_fh(x),L_f^2h(x),...,L_f{}^{r-1}h(x)\right)$. As well, we assume that u can be selected so that $L^r{}_fh(x)+L_gL_f{}^{r-1}h(x)u=\mu$ for $\mu\in U_\mu\subset\mathrm{R}$ which is a slack input. We have:

$$\begin{aligned} &\dot{z}(x)=f_bz(x)+g_b\mu\\ &h(x)=p_bz(x) \end{aligned}$$

Where, f_b,g_b,p_b are,

$$f_b=\begin{bmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\\0&0&0&\cdots&0\end{bmatrix},g_b=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix},p_b=\begin{bmatrix}1&0&\cdots&0\end{bmatrix}$$

If a set $\mathrm{L}\subset R^n$ is defined as the super level set for an r-times functions which are continuously differentiable, then h is considered as an exponential control barrier function (ECBF) for the control system (1). Therefore, the acceptable space SE(x) (if $K_\alpha\in\mathrm{R}^r$ exists) is defined as follows,

$$SE = \sup_{u \in U} [A + K_\alpha z(x)] \geq 0$$

Where, $A = L_f^r h(x) + L_g L_f^{r-1} h(x) u$

As Assumption 3, the admissible control space S (x) can be considered not empty.

3. Reinforcement Learning for Safe Optimal Control under Uncertain Dynamics

In this part, the potential inconformity between the model and the plant elements is examined, while there is paucity of accurate knowledge of the true plant vector fields g , f . Moreover, its effects on the dynamics of CLF and CBF will be examined.

Allow the substantial model utilized in the controller to be characterized as follows:

$$\dot{x} = \hat{f}(x) + \hat{g}(x) u \tag{10}$$

Assume that the vectors $\hat{f}: R^n \to R^n, \hat{g}: R^n \to R^n$ are Lipschitz continuous and Where,

Problem 1. (Safe Optimal Control under uncertainty dynamics): Find a controller that solves the following equation:

$$\begin{aligned} u^* &= \arg\min \int_\Omega V dx + k_\delta \delta^2 \\ \text{st.} \quad & H(V) \leq \delta \\ & \hat{A} + K_\alpha z(x) \geq 0 \end{aligned} \tag{11}$$

In relation (11), Ω is an area in which the system performance is expected to be improved, $k_\delta > 0$ is the design parameter that acts as a trade-off between the system aggressiveness toward performance and safety, and δ is the Stability relaxation factor. Note that δ can be defined as the Aspiration level for a performance that shows the level of performance sacrificed as a result of failure in satisfying safety and performance. However, this parameter is minimized to achieve the highest possible performance.

First, the relaxed optimal control problem for system (1) with performance (2) is examined as follows:

$$\begin{aligned} &\min_V \int_\Omega V(x) dx \\ &H(V) \leq 0 \\ &V \in P \end{aligned} \tag{12}$$

In which $H(V)$ is defined by Equation (4) and $\Omega \subset R^n$ is an ideal compact set containing the origin [29]. Problem 1 actually solves a relaxed version of HJB (4) in which the HJB equation is relaxed with the HJB inequality. Ames et al. have shown that the solution of problem 1 is unique and if V^* is a solution for (9), then

$$u^{opt}(x) = -\frac{1}{2} R^{-1}(x) (L_g V^*)^T(x) \tag{13}$$

The stability of the system is guaranteed and V^* plays the role of an upper bound or an overestimate for the actual cost. The superscript opt is used here to indicate that u^{opt} is a performance-oriented controller. However, with a safe control policy u^i, V^i and δ^i are determined to tackle the following optimization subject.

This control policy doesn't confirm system safety.

$$\begin{aligned} &\min_{V^i, \delta^i} \int_\Omega V^i dx + k_\delta \delta_i^2 \\ &L(V^i, u^i) = -L_f V^i - -L_g V^i u^i - r(x, u^i) \geq \delta_i \quad \forall x \in R^n \\ &V^{i-1} - V^i \geq 0 \end{aligned} \tag{14}$$

In SOS framework, this optimization problem is defined as follows:

$$\begin{aligned} &\min_{V^i, h^i} \int_\Omega V^i dx + k_\delta \delta_i^2 \\ &L(V^i, u^i) + \delta_i \text{ is SOS} \quad \forall x \in R^n \\ &V^{i-1} - V^i \text{ is SOS} \end{aligned}$$

Based on Assumption 1, there is a safe control policy u. Now we can write the control policy as $u = u^{opt} + u^{safe}$ in which $u^{opt} = \frac{1}{2} R^{-1}(x) (L_g V^*)^T(x)$ is a part of the controller that is applied to optimize performance regardless of safety and u^{safe} has been added to u^{opt} in order to guarantee safety.

3.1 Deriving u^{opt} under Uncertainty Situation

Lemma 1: Consider system (10). Suppose that u is a global safe control policy and $V_{i-1} \in P$ is also existed. Then the system (11) is feed forward.

Proof: According to the assumptions 1 and 2, $V_{i-1} \in P$. Then by sum of squares, we conclude that

$$\begin{aligned} &V_{i-1}^T \left(f + g u^{opt} + g u^{safe} \right) \leq -u^{opt^T} R u^{opt} - 2 u^{opt^T} R u^{safe} \\ &= -\left| u^{opt} + u^{safe} \right|_R^2 + \left| u^{safe} \right|_R^2 \leq \left| u^{safe} \right|_R^2 \leq \left| u^{safe} \right|_R^2 + V_{i-1} \end{aligned} \tag{15}$$

According to Result 2.11 [24], system (11) is feed forward:

There is a fixed matrix $W_i \in \mathbb{R}^{m \times n_t}$ in which $m_t = \binom{m+t}{t} - 1$ such that $u^{opt} = W_i \vec{n}_{1,t}(x)$. It is also assumed that there is a fixed vector $p \in R^{n_t}$ in which $m_t = \binom{m+t}{t} - m - 1$ so that $V = p^T \vec{n}_{2,2t}(x)$. Then, the following terms can be defined along with the solutions of the system (11):

$$\begin{aligned}\dot{V} &= (L_f V(x) + L_g V(x) u^{opt}) + L_g V(x) u^{safe} \\ &= -r\left(x, u^{opt}\right) - \mathrm{L}\left(V, u^{opt}\right) + L_g V(x) u^{safe} \\ &= -r\left(x, u^{opt}\right) - \mathrm{L}\left(V, u^{opt}\right) + \left(R^{-1} g^T \nabla V\right)^T R u^{safe}\end{aligned}$$

Note that two terms $\mathrm{L}\left(V, u^{opt}\right)$ and $R^{-1} g^T \nabla V$, depend on $\hat{f}$ and $\hat{g}$. Since there is uncertainty in these terms, we should solve them without recognizing $\hat{f}$ and $\hat{g}$.

For a similar abovementioned pair (V, u^{opt}), we can find a fixed vector $b_p \in \mathrm{R}^{n_{2t}}$, in which $m_{2t} = \binom{n+2t}{2t} - m - 1$ and $W_p \in \mathrm{R}^{m \times n_t}$ is a fixed matrix, such that

$$\mathrm{L}\left(V, u^{opt}\right) = b_p^T \vec{n}_{2,2t}(x) \tag{16}$$

$$-\frac{1}{2} R^{-1} g^T \nabla V = W_p \vec{n}_{1,t}(x) \tag{17}$$

Therefore, $\mathrm{L}\left(V, u^{opt}\right)$ and $R^{-1} g^T \nabla V$ are calculated to find b_p and W_p By substituting Equations (16) and (17) in Equation (15), we have:

$$\dot{V} = -r(x, u^{opt}) - b_p^T \vec{n}_{2,2t}(x) - 2\vec{n}_{1,t}^T(x) W_p^T \tag{18}$$

By integrating (18) into the time interval [t, t + δt]:

$$\begin{aligned}&p^T\left[\vec{n}_{2,2t}(x(t)) - \vec{n}_{2,2t}(x(t+\delta t))\right] = \\ &\int_t^{t+\delta t}\left(r(x, u_i) + b_p^T \vec{n}_{2,2t}(x) + 2\vec{n}_{1,d}^T(x) W_p^T R u^{safe}\right) dt\end{aligned} \tag{19}$$

Now, b_p and W_p can be calculated without having accurate information about $\hat{f}$ and $\hat{g}$ by using real online data.

1) Initial value:

Find the pair (V_0, u) that satisfies Assumption 1. Consider a fixed vector p_0 such that $V_0 = p_0^T \vec{m}_{2,2t}(x)$, and $i = 1$.

2) Online data collection:

First, apply $u = u^{opt} + u^{safe}$ to the system and then find an optimal solution (p_i, W_{i+1}) for the following SOS program.

$$\begin{aligned}&\min_{p, W_p} \int_\Omega \vec{n}_{2,2t}(x) dx^T p + K_\delta \delta_i^2 \\ &b_p^T \vec{n}_{2,2t}(x) \, is SOS \\ &(p_{i-1} - p_i)^T \vec{n}_{2,2t}(x) \, is SOS\end{aligned} \tag{20}$$

So, we have $V^i = p_i^T \vec{n}_{2,2t}(x)$. Then, we can derive the value of $u^{opt} = W_p \vec{n}_{1,t}(x)$ and proceed to step 2) where $i \leftarrow i + 1$.

3.2 Reinforcement Learning for CBFs

The control rule for the computed input-output linearization has the following form based on the $\hat{f}$ and $\hat{g}$:

$$\hat{u}(\mathrm{x}, \mu) = \hat{u}^*(\mathrm{x}) + \left(\mathrm{L}_{\hat{g}} \mathrm{L}_{\hat{f}} h(\mathrm{x})\right)^{-1} \mu \tag{21}$$

In which μ is also an auxiliary input.

Under the uncertainty situation, it can be written:

$$\begin{aligned}\widehat{\boldsymbol{A}} &= \boldsymbol{L}_{\hat{\boldsymbol{f}}}^{\boldsymbol{r}} \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{L}_{\hat{\boldsymbol{g}}} \boldsymbol{L}_{\hat{\boldsymbol{f}}}^{\boldsymbol{r-1}} \boldsymbol{h}(\boldsymbol{x}) \widehat{\boldsymbol{u}} \\ \boldsymbol{A} &= \widehat{\boldsymbol{A}} + \boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{\mu}\end{aligned} \tag{22}$$

Where α and β are

$$\begin{aligned}\alpha &= L_f^r h(x) - L_g L_f^{r-1} h(x)\left(L_{\hat{f}} L_{\hat{f}}^{r-1} h(x)\right)^{-1} L_{\hat{f}}^r h(x) \\ \beta &= L_g L_f^{r-1} h(x)\left(L_{\hat{g}} L_{\hat{f}}^{r-1} h(x)\right)^{-1}\end{aligned}$$

Terms obtained from the mismatch existing between model and plant. It should also be noted that if α, β are zero, we have the same equation as (22).

Using an estimator made of A that in the form $A = \widehat{A} + \alpha + \beta\mu$.

RL's goal is to learn α, β policies so that $\widehat{A}$ is close to A as much as possible. Thereby, using RL, the uncertainty terms for CBF can be estimated. Therefore, there is a need for designing the reward function to minimize policy estimation errors. Therefore, it can be defined as follows:

$$l = A - \hat{A}$$

The RL factor embraces a policy that considers the uncertainty terms in CBF, which are summed with the SOS constraints as they are extracted from the nominal model, resulting in accurate estimates. One can consider the focal RL problem with the considered reward for a given state x as the summation of the negative objective functions plus an arbitrary penalty (s) selected by the user

$$rl(x, \theta) = -\sum_{i=1}^{b} w_i l_{i,\theta} - s \tag{23}$$

Where b is the number of CBFs. One can solve RL, using common algorithms.

4. Applications

The reason of this part is to demonstrate that our proposed system can make possible the critical safe control, even in the presence of uncertain conditions. Two simulation examples are presented in this section in order to approve the efficiency of the proposed model.

Example 1:

Consider the car quarter suspension model shown in Figure 1. Its non-linear dynamic is defined as follows. However, it is worth mentioning that while the training experiences or the simulations are operating, the car quarter suspension model is assumed to be under the proper dynamics (given its uncertainties) [30].

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{1}{M_b}\left[k_a(x_1 - x_3) + k_n(x_1 - x_3)^3 + c_a(x_2 - x_4) + u\right] \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{1}{M_{us}}\left[k_a(x_1 - x_3) + k_n(x_1 - x_3)^3 + c_a(x_2 - x_4) + k_t x_3 - u\right]\end{aligned} \tag{24}$$

Where, x_1, x_2, and M_b are the car position, velocity, and its mass, respectively. x_3, x_4, and M_{us} are also the wheel position, velocity, and their total mass. K_t, K_a, K_n, and C_a shows the tire hardness, the system of linear pendency, the non-linear suspension hardness, and the damping rate of the pendency system, respectively.

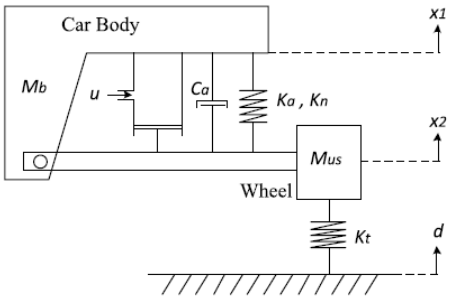

Figure 1. Quarter car model

The uncertainty for the significant model in this experiment is introduced by weighing all the components with a weighing coefficient of 2. During the training (process) of the RL agent, we only know the nominal model.

Let, $M_b \in [250,350]$, $M_{us} \in [55,65]$, $c_a \in [450,550]$, $k_a \in [7500,8500]$, $k_n \in [750,850]$, $k_t \in [90000,10000]$. Then, it can be easily observed that the system establishment has been done in a global level asymptotically, with an absence of input control. The purpose of the proposed method is to design an active suspension control system which lessens the performance index, while retains the global asymptote stability, simultaneously. As well, reducing the disorder effects in the set $\Omega = \{x \mid x \in R^4 \; and \; |x_1| \le 0.03, |x_2| \le 5, |x_3| \le 0.03, |x_4| \le 5\}$ can improve the system performance.

The reinforcement learning factor is taught using a Deep Deterministic Policy Gradient algorithm (DDPG, Silver et al. [31]). The 4 observed state variables, and the CBF component of the simulation constitute the inputs for the actor neural network. The output dimension is equal to which corresponds to 4×1 α_θ^B, and 1×1 β_θ^B.

There exist hidden layers as wide as 200 and 300 in both the actor and the critic neural networks in example 1. This agent is trained by simulation in the interval between $t=0$, and $t=80$.

A time step of Ts = 1 is employed (in this regard). The simulations have been carried out on a 6-core laptop with Intel Core™ i7-9400 (2.7 GHz) processor and 4 GB RAM.

Use SOSTOOLS to obtain an initial cost function, V0 for the simulated system having non-determined parameters [32].

Then, we apply the proposed method in which u1=0. The primary condition has been selected randomly. To do the training, we apply the noise from $t=0$ to $t=80$ till the convergence is obtained after 8 repetitions.

The obtained control policy is as follows,

$$\begin{aligned} u_8 = {} & -1.76x_1^3 - 5.33x_1^2x_2 + 7.7x_1x_3 + 3.22x_1x_3x_4 - 12.1x_1^2 + 4.43x_1x_2^2 + 0.87x_1x_2^2x_3 \\ & + 0.594x_1x_2x_4 - 4.61x_1x_2 - 6.3x_1x_3^2 - 6.19x_1x_2^2x_3 - 0.174x_1x_4^2 - 2.81\times10^8 x_1x_4 \\ & - 18.1x_1 - 0.73x_2^3 + 0.006x_2^2x_3 + 2.26x_2^2x_4 - 4.07x_2x_3^2 + 1.71x_2x_3x_4 - 4.55x_2x_3 \\ & - 1.35x_2x_4^2 - 4.94x_2x_4 - 2.8x_3^2x_4 + 4.47x_3^2 + 0.241x_3x_4^2 \\ & + 2.62\times10^8 x_3x_4 + 11.1x_3 - 11.62x_3^2 + 6.39x_3^3 + 0.33x_4^3 + 4.61x_4^2 + 10.4x_4 \end{aligned} \quad (25)$$

To test the trained controller, we choose the road disorder as a single-impact as follows,

$$\begin{cases} 0.003(2-\cos(2\pi t)) & t=60 \\ 0 & otherwise \end{cases} \quad (26)$$

In addition, as an indication of a car carrying a load, an overweight of 260 kg is applied to the vehicle assembly.

So that, the departure of position is relative to the origin. The proposed control policy performance is compared to the primary system performance without any control, as shown in Figure 2. In Figure 3, these two performances of the costs are compared by the constraint wheel position, wheel velocity when they are zero. As can be seen, V_8 has been reduced significantly compared to V_0.

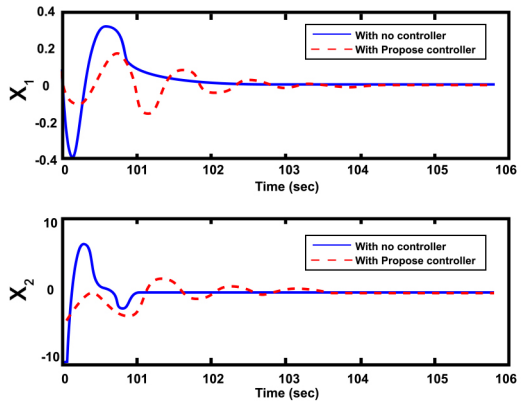

Figure 2. Comparison of performance car position and car velocity

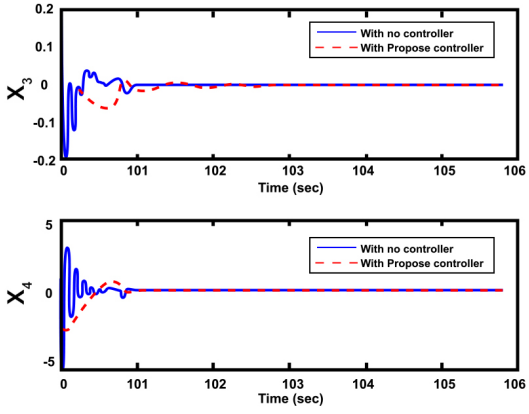

Figure 3. Comparison of performance wheel position and wheel velocity

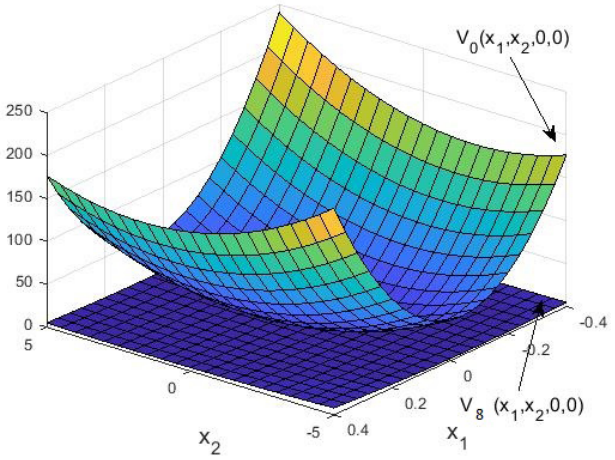

Figure 4. Comparison of learned value functions

Example 2:

Now consider the following system equations:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}x_1^2+x_1x_2^2-x_1x_2\\ 2x_1-x_2\end{bmatrix}+\begin{bmatrix}0 & \alpha_1\\ \alpha_2 & \alpha_1\end{bmatrix}u \tag{27}$$

In which $\alpha_1,\alpha_2\in[0.25,1]$ are uncertain parameters, and $x=[x_1,\ x_2]$ and u are mode and system control, respectively. The unsafe space was coded with a polynomial inequality $X_u=\{x\in R^2\,|\,bf_i(x)<0, i=1,2,3\}$

With the following details:

$$bf_1=-0.5+(x_1+1)^2+(x_2+2)^2<0$$
$$bf_2=-0.5+(x_1+1.5)^2+(x_2-1.5)^2<0$$
$$bf_3=-0.5+(x_1-1.5)^2+(x_2-1)^2<0$$

Using SOS strategies, the system (27) can stabilize, at the source level globally and asymptotically by the following robust control policy [33].

$$u_1^{opt}=\begin{bmatrix}u_1^1\\ u_2^1\end{bmatrix}=\begin{bmatrix}1.192x_1+3.568x_2\\ 1.7x_1-2.905x_2\end{bmatrix} \tag{28}$$

However, the optimality of the closed-loop system has not been fully addressed.

The primary goal of the control is to find more improved safeguard policies under uncertainty using the iterative safeguard policy algorithm. Then, with the help of solving the feasibility study and SOS-TOOLS, we will reach Equation (29) [34]:

$$\mathrm{L}\left(V,u_1^{opt}\right)\quad isSOS,\quad \forall\alpha_1,\alpha_2\in[0.25,1] \tag{29}$$

The V function is obtained as follows:

$$\mathrm{V}_1=7.6626\mathrm{x}_1^2-4.264\mathrm{x}_1\mathrm{x}_2+6.5588\mathrm{x}_2^2-0.1142\mathrm{x}_1^3+1.7303\mathrm{x}_1^2\mathrm{x}_2$$
$$-1.0845\mathrm{x}_1\mathrm{x}_2^2 3.4848\mathrm{x}_1^4-0.361\mathrm{x}_1^3\mathrm{x}_2+4.6522\mathrm{x}_1^2\mathrm{x}_2^2+1.9459\mathrm{x}_2^4$$

If we put α_1=0.5 and α_2=0.5 the initial condition is arbitrarily set to $x_1(0)$=1 and $x_2(0)$=-1.

$$u_1^6=-0.04x_1^3-0.67x_1^2x_2-0.0747x_1^2+0.0469x_1x_2-$$
$$0.986x_1-0.067x_2^3-2.698x_2$$
$$\mathrm{u}_2^6=-0.067\mathrm{x}_1^3-0.09\mathrm{x}_1^2\mathrm{x}_2-0.201\mathrm{x}_1^2+0.025\mathrm{x}_1\mathrm{x}_2^2- \tag{30}$$
$$0.187\mathrm{x}_1\mathrm{x}_2-1.436\mathrm{x}_1-0.1396\mathrm{x}_2^3-0.345\mathrm{x}_2^2-2.27\mathrm{x}_2$$

The V function is as follows:

$$V_6=1.4878x_1^2+0.8709x_1x_2+4.4963x_2^2+0.0131x_1^3+$$
$$0.2491x_1^2x_2-0.0782x_1x_2^2+0.0639x_2^3+0.0012x_1^3x_2+$$
$$0.0111x_1^2x_2^2-0.0123x_1x_2^3+0.0314x_2^4$$

The indefinite cost function and the initial cost function are compared in Figure 5.

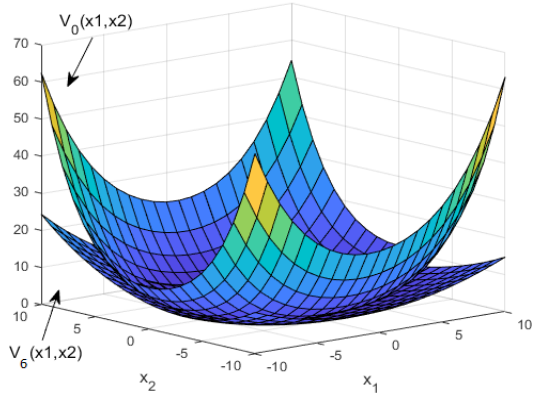

Figure 5. Comparison of learned value functions

Both operator and critical neural networks in example 2 have hidden layers with a width of 100 and 200. The training environment using the learning environment was the same as the previous example, the proposed method was learning took 2 seconds per episode. Control policy Obtained after 5 episodes.

In addition, the safe set is equal to:

$$\ell=\{x\in\mathrm{R}^2\,|\,h(x)\geq 0\}$$

In which:

$$h(x)=0.452-0.0023x_1^2-0.0382x_1-0.014x_2-0.0067x_1x_2-0.0077x_2^2 \tag{31}$$

Note that it is necessary for the safe set to be a member of the complementary set of the unsafe set, as well as being invariable in a way that it never leaves the set in the future. The safe set is obtained using CBF $h\ (x)$. Be attention that barrier certificate is bounded to a second-order polynomial. In Figure 6, the estimated safe sets for both the initial control policy and the optimal control policy are shown.

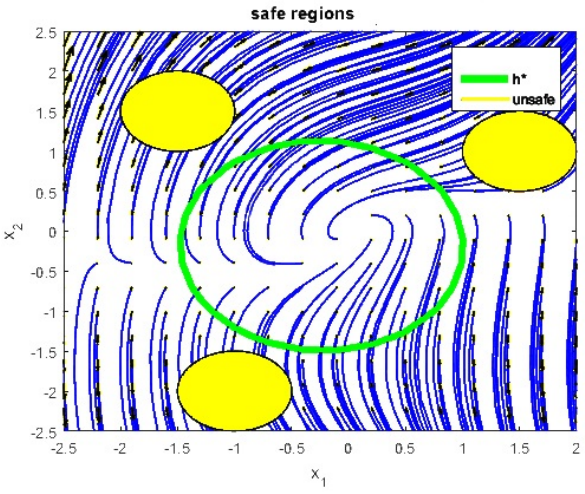

Figure 6. Safe area estimation by the proposed optimal safe controller despite the uncertainty

5. Conclusions

A safe optimization is proposed for the control of dynamics systems under model uncertainty. In order for the performance and safety to be guaranteed, a Hamilton-Jacobi-Bellman (HJB) inequality replaces the HJB equality; besides, a safe policy iteration algorithm is presented certifying the safety of the improved policy and finding a value function corresponding to it. Also, the RL factor was also presented in the proposed method to reduce model uncertainty. The effectiveness of the proposed method is illustrated through two simulation examples.

Conflict of Interest

There is no conflict of interest.

References

[1] Beard, R.W., Saridis, G.N., Wen, J.T., 1997. Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation, Automatica. 33(12), 2159-2177.
DOI: https://doi.org/10.1016/S0005-1098(97)00128-3

[2] Vamvoudakis, K.G., Lewis, F.L., 2010. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica. 46(5), 878-888.
DOI: https://doi.org/10.1109/IJCNN.2009.5178586

[3] Lewis, F.L., Vamvoudakis, K.G., 2011. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. IEEE Transactions on Systems. 41(1), 14-25. http://www.derongliu.org/adp/adp-cdrom/Vamvoudakis2011.pdf

[4] Kiumarsi, B., Lewis, F.L., 2015. Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems, IEEE Transactions on Neural Networks and Learning Systems. 26(1), 140-151.
DOI: https://doi.org/10.1109/TNNLS.2014.2358227

[5] Modares, H., Lewis, F.L., Naghibi-Sistani, M.B., 2014. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, Automatica. 50(1), 193-202.
DOI: https://doi.org/10.1016/j.automatica.2013.09.043

[6] Wang, D., Liu, D., Zhang, Y., et al., 2018. Neural network robust tracking control with adaptive critic framework for uncertain nonlinear systems, Neural Networks. 97(1), 11-18.
DOI: https://doi.org/10.1016/j.neunet.2017.09.005

[7] Bhasin, S., Kamalapurkar, R., Johnson, M., et al., 2013. A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems, Automatica. 49(1), 82-92. https://ncr.mae.ufl.edu/papers/auto13.pdf

[8] Gao, W., Jiang, Z., 2018. Learning-based adaptive optimal trackingcontrol of strict-feedback nonlinear systems. IEEE Transactions on Neural Networks and Learning Systems. 29(1), 2614-2624. https://ieeexplore.ieee.org/ielaam/5962385/8360119/8100742-aam.pdf

[9] Abu-Khalaf, M., Lewis, F.L., 2004. Nearly optimal state feedback control of constrained nonlinear systems using a neural networks hjb approach. Annual Reviews in Control. 28(2), 239-251.
DOI: http://dx.doi.org/10.1016/j.arcontrol.2004.07.002

[10] Ames, A.D., Grizzle, J.W., Tabuada, P., 2014. Control barrier function based quadratic programs with application to adaptive cruise control. 53rd IEEE Conference on Decision and Control. pp. 6271-6278.
DOI: https://doi.org/10.1109/CDC.2014.7040372

[11] Ames, A.D., Xu, X., Grizzle, J.W., et al., 2017. Control barrier function based quadratic programs for safety critical systems. IEEE Transactions on Automatic Control. 62(8), 3861-3876.
DOI: https://doi.org/10.1109/TAC.2016.2638961

[12] Nguyen, Q., Sreenath, K., 2016. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. 2016 American Control Conference (ACC). pp. 322-328.
DOI: https://doi.org/10.1109/ACC.2016.7524935

[13] Romdlony, M.Z., Jayawardhana, B., 2014. Uniting control Lyapunov and control barrier functions. 53rd IEEE Conference on Decision and Control. pp. 2293-2298.

DOI: https://doi.org/10.1109/CDC.2014.7039737

[14] Xu, X., Tabuada, P., Grizzle, J.W., et al., 2015. Robustness of control barrier functions for safety critical control. Analysis and Design of Hybrid Systems ADHS IFAC Papers Online. 48(27), 54-61. DOI: https://doi.org/10.1016/j.ifacol.2015.11.152

[15] Prajna, S., Rantzer, A., 2005. On the necessity of barrier certificates. 16thIFAC World Congress IFAC Proceedings. 38(1), 526-531. DOI: https://doi.org/10.3182/20050703-6-CZ-1902.00743

[16] Ames, A.D., Powell, M., 2013. Towards the unification of locomotion and manipulation through control lyapunov nctions and quadratic programs. In Control of Cyber Physical Systems. pp. 219-240. http://ames.caltech.edu/unify_ames_powell.pdf

[17] Galloway, K., Sreenath, K., Ames, A.D., et al., 2015. Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs. pp. 323-332. DOI: https://doi.org/10.1109/ACCESS.2015.2419630

[18] Taylor, A.J., Dorobantu, V.D., Le, H.M., et al., 2019. Episodic learning with control lyapunov functions for uncertain robotic systems. ArXiv preprint. https://arxiv.org/abs/1903

[19] Taylor, A.J., Singletary, A., Yue, Y., et al., 2019. Learning for safety-critical control with control barrier functions. ArXiv preprint. https://arxiv.org/abs/1912.10099

[20] Westenbroek, T., Fridovich-Keil, D., Mazumdar, E., et al., 2019. Feedback linearization for unknown systems via reinforcement learning. ArXiv preprint. https://arxiv.org/abs/1910.13272

[21] Hwangbo, J., Lee, J., Dosovitskiy, A., et al., 2019. Learning agile and dynamic motor skills for legged robots. Science Robotics. 4(26), 58-72. https://arxiv.org/abs/1901.08652

[22] Levine, S., Finn, C., Darrell, T., et al., 2016. End-toend training of deep visuomotor policies. Learning Research. 17(1), 1532-4435. https://arxiv.org/abs/1504.00702

[23] Bansal, S., Calandra, R., Xiao, T., et al., 2017. Goal-driven dynamics learning via Bayesian optimization. 56th Annual Conference on Decision and Control (CDC). pp. 5168-5173. DOI: https://doi.org/10.1109/CDC.2017.8264425

[24] Fisac, J.F., Akametalu, A.K., Zeilinger, M.N., et al., 2019. A general safety framework for learning-based control in uncertain robotic systems. IEEE Transactions on Automatic Control. 64(7), 2737-2752. DOI: https://doi.org/10.1109/TAC.2018.2876389

[25] Prajna, S., Jadbabaie, A., 2004. Safety verification of hybrid systems using barrier certificates. In International Workshop on Hybrid Systems: Computation and Control. Springer. 2993(1), 477-492. https://viterbi-web.usc.edu/~jdeshmuk/teaching/cs699-fm-for-cps/Papers/A5.pdf

[26] Yazdani, N.M., Moghaddam, R.K., Kiumarsi, B., et al., 2020. A Safety-Certified Policy Iteration Algorithm for Control of Constrained Nonlinear Systems. IEEE Control Systems Letters. 4(3), 686-691. DOI: https://doi.org/10.1109/LCSYS.2020.2990632

[27] Lewis, F.L., Vrabie, D., Syrmos, V.L., 2012. Optimal control, 3rd Edition. John Wiley & Sons.

[28] Wang, L., Ames, A., Egerstedt, M., 2016. Safety barrier certificates for heterogeneous multi-robot systems. 2016 American Control Conference (ACC). pp. 5213-5218. DOI: https://doi.org/10.1109/ACC.2016.7526486

[29] Ames, A.D., Coogan, S., Egerstedt, M., et al., 2019. Control barrier functions: Theory and applications. In Proc 2019 European Control Conference. https://arxiv.org/abs/1903.11199

[30] Jiang, Y., Jiang, Z., 2015. Global adaptive dynamic programming for continuous-time nonlinear systems. IEEE Transactions on Automatic Control. 60(1), 2917-2929. DOI: https://doi.org/10.1109/TAC.2015.2414811

[31] Gaspar, P., Szaszi, I., Bokor, J., 2003. Active suspension design using linear parameter varying control. International Journal of Vehicle Autonomous Systems. 1(2), 206-221. DOI: https://doi.org/10.1016/S1474-6670(17)30403-2

[32] Silver, D., Lever, G., Heess, N., et al., 2014. Deterministic policy gradient algorithms. International conference on machine learning. pp. 387-395. http://proceedings.mlr.press/v32/silver14.pdf

[33] Papachristodoulou, A., Anderson, J., Valmorbida, G., et al., 2013. SOSTOOLS: Sum of squares optimization toolbox for MATLAB. Control and Dynamical Systems, California Institute of Technology, Pasadena. http://arxiv.org/abs/1310.4716

[34] Xu, J., Xie, L., Wang, Y., 2009. Simultaneous stabilization and robust control of polynomial nonlinear systems using SOS techniques. IEEE Transactions on Automatic Control. 54(8), 1892-1897. DOI: https://doi.org/10.1109/TAC.2009.2022108