Proceedings Paper:
Kim, J (2022) Improved Robustness Analysis of Reinforcement Learning Embedded Control Systems. In: Lecture Notes in Networks and Systems. RiTA 2021: International Conference on Robot Intelligence Technology and Applications, 16-17 Dec 2022, Daejeon, Korea. Springer , pp. 104-115. ISBN 9783030976712

https://doi.org/10.1007/978-3-030-97672-9_10

Improved Robustness Analysis of Reinforcement Learning Embedded Control Systems

Jongrae Kim[1]

University of Leeds, Leeds LS2 9JT, UK,
menjkim@leeds.ac.uk,
WWW home page: http://robustlab.org

Abstract. Reinforcement learning emerges as an efficient tool to design control algorithms for nonlinear systems. There are, however, few results available on how the robustness of the closed-loop dynamics with reinforcement learning is performed. While μ-analysis is well established as the robustness analysis tool for linear systems, there is also a limitation caused by ignoring the equilibrium shift by the uncertain parameters. An improved linearisation method for μ-analysis is presented and the method is applied to the inverted-pendulum system with the reinforcement learning control. The resulting robustness analysis provides a significantly less conservative upper bound to the smallest worst-case perturbation.

Keywords: Robustness Analysis, Reinforcement Learning, Inverted-Pendulum

1 Introduction

Consider the following nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathrm{OL}}(\mathbf{x},\mathbf{p}) + \mathbf{g}(\mathbf{x},\mathbf{p})\mathbf{u} \tag{1}$$

where $\dot{\mathbf{x}} = d\mathbf{x}/dt$, $\mathbf{x}$ is the state in $\Re^n$, $\mathbf{p}$ is the parameters to characterise the nonlinear system in $\Re^p$, $\mathbf{u}$ is the control input in $\Re^m$, $\mathbf{f}_{\mathrm{OL}}(\cdot,\cdot)$ and $\mathbf{g}(\cdot,\cdot)$ are nonlinear functions, which satisfy the conditions for the existence and the uniqueness of the solution of the nonlinear differential equation, $\Re$ is the real number set, $d()/dt$ is the time derivative, and n, p and m are the positive integers with appropriate values. Once the state feedback control is designed so that $\mathbf{u} = \mathbf{u}(\mathbf{x})$, the closed-loop system dynamics becomes

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{p}) \tag{2}$$

where

$$\mathbf{f}(\mathbf{x},\mathbf{p}) = \mathbf{f}_{\mathrm{OL}}(\mathbf{x},\mathbf{p}) + \mathbf{g}(\mathbf{x},\mathbf{p})\mathbf{u}(\mathbf{x}) \tag{3}$$

and $\mathbf{f}(\mathbf{x},\mathbf{p})$ satisfies the conditions for the existence and the uniqueness of the solution of the differential equation. Note that $\mathbf{u}(\mathbf{x})$ would be a complex nonlinear function. In this paper, it is a control policy trained by the reinforcement learning with neural network functions, particularly, the Deep Deterministic Policy Gradient (DDPG) [1] is used to design $\mathbf{u}(\mathbf{x})$.

The equilibrium point, $\mathbf{x}_{\text{eq}}$, is obtained by solving the following algebraic equation:

$$\mathbf{f}(\mathbf{x}_{\text{eq}}, \mathbf{p}) = 0 \tag{4}$$

Introduce a small perturbation, $\delta\mathbf{x}$, around the equilibrium point as

$$\mathbf{x} = \mathbf{x}_{\text{eq}} + \delta\mathbf{x} \tag{5}$$

Take the time derivative and obtain

$$\delta\dot{\mathbf{x}} = \left.\frac{\partial \mathbf{f}}{\partial x}\right|_{(\mathbf{x}_{\text{eq}}, \mathbf{p})} \delta\mathbf{x} = F_x(\mathbf{x}_{\text{eq}}, \mathbf{p})\, \delta\mathbf{x} \tag{6}$$

The robustness analysis is performed by introducing perturbations in the parameters for the linearised system, (6), as follows:

$$\mathbf{p} = \bar{\mathbf{p}} + \delta\mathbf{p} \tag{7}$$

and checking the stability of $F_x(\mathbf{x}_{\text{eq}}, \mathbf{p})$ for some ranges of the perturbation magnitude, i.e., $\|\delta\mathbf{p}\| < (1/\mu)$, where $\bar{\mathbf{p}}$ is the nominal parameter values, $\|\cdot\|$ is typically the ∞-norm and μ is a positive real number. Finding the maximum μ, where $F_x(\mathbf{x}_{\text{eq}}, \mathbf{p})$ is destabilised for some $\delta\mathbf{p}$, whose norm is less than or equal to $1/\mu$, is the μ-analysis problem [2].

Although this approach has been widely used, its validity only for the region, in which the equilibrium point remains *sufficiently close* to the original equilibrium point for all possible parametric perturbations, has not been fully considered to the best of the author's knowledge. Only two exceptions are the study presented by [3] and [4], where the equilibrium shift for polynomial uncertain parameters is considered for a polynomial system typically occurred in chemical reaction networks. If the validity of the linear approximation is violated, there is the risk that all analyses based on this model provide incorrect robustness results. Due to *the completely ignored equilibrium point shifting* when the perturbation is introduced, it would provide inadequate robustness results for the linearised systems derived from nonlinear systems.

The rest of the paper is organised as follows: firstly, the robustness analysis including equilibrium perturbation is formulated; secondly, the approach is applied to the inverted-pendulum with the DDPG control; finally, the conclusions and future works are presented.

2 Robustness Analysis

This section introduces perturbations in equilibrium points. The linearisation including the perturbations is derived. And, the robustness analysis method is proposed.

2.1 Equilibrium Point Perturbation

Suppose $\mathbf{p}$ is perturbed as (7), then the equilibrium point is perturbed by the parameter changes as follows:

$$\mathbf{x}_{\text{eq}}^{+} = \mathbf{x}_{\text{eq}} + \delta\mathbf{x}_{\text{eq}} \tag{8}$$

where $\mathbf{x}_{\text{eq}}^{+}$ indicates the equilibrium point with the perturbation, $\delta\mathbf{p}$. $\delta\mathbf{x}_{\text{eq}}$ is not an independent perturbation but depending on $\delta\mathbf{p}$. The relationship between these two perturbations are obtained by solving the following algebraic equation:

$$\mathbf{f}(\mathbf{x}_{\text{eq}}^{+}, \mathbf{p}) = \mathbf{f}(\mathbf{x}_{\text{eq}} + \delta\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}} + \delta\mathbf{p}) = 0 \tag{9}$$

The Taylor series expansion up to the first-order provides

$$\mathbf{f}(\mathbf{x}_{\text{eq}}^{+}, \mathbf{p}) \approx \mathbf{f}(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) + F_x(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}})\delta\mathbf{x}_{\text{eq}} + F_p(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}})\delta\mathbf{p} = 0$$

where

$$F_x(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) = \left.\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right|_{(\mathbf{x}_{\text{eq}},\bar{\mathbf{p}})}, \qquad F_p(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) = \left.\frac{\partial\mathbf{f}}{\partial\mathbf{p}}\right|_{(\mathbf{x}_{\text{eq}},\bar{\mathbf{p}})} \tag{10}$$

Hence,

$$\delta\mathbf{x}_{\text{eq}}(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) = -F_x^{-1}(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}})\, F_p(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}})\delta\mathbf{p} \tag{11}$$

where the inversion of F_x always exists with the assumption that the linearised system at the unperturbed equilibrium point, $(\mathbf{x}_{\text{eq}},\ \bar{\mathbf{p}})$, is Hurwitz stable, i.e., all real parts of the eigenvalues are strictly negative. This is the prerequisite for any robustness analysis. Equation (11) provides the way to calculate how much the equilibrium would be perturbed from the original equilibrium by the parameter perturbations.

It is worth pointing out that $F_x(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}})$ is expressed in terms of the open-loop dynamics and the control input function as follows:

$$\begin{aligned} F_x(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) = & \left.\frac{\partial\mathbf{f}_{\text{OL}}}{\partial\mathbf{x}}\right|_{(\mathbf{x}_{\text{eq}},\bar{\mathbf{p}})} \\ & + \sum_{i=1}^{m} \left.\frac{\partial\mathbf{g}}{\partial x_i}\right|_{(\mathbf{x}_{\text{eq}},\bar{\mathbf{p}})} \mathbf{u}(\mathbf{x}) + \mathbf{g}(\mathbf{x}_{\text{eq}}, \bar{\mathbf{p}}) \sum_{i=1}^{m} \left.\frac{\mathbf{u}(\mathbf{x})}{\partial x_i}\right|_{\mathbf{x}=\mathbf{x}_{\text{eq}}} \end{aligned} \tag{12}$$

2.2 Linearized System at $\mathbf{x}_{\text{eq}}^{+}$

Introduce a small perturbation $\delta\mathbf{x}$ around $\mathbf{x}_{\text{eq}}^{+}$, and the dynamics of the perturbation is approximated by

$$\delta\dot{\mathbf{x}} = F_x(\mathbf{x}_{\text{eq}}^{+}, \mathbf{p})\, \delta\mathbf{x} \tag{13}$$

Equation (13) includes the equilibrium point perturbation caused by the parametric perturbation. Calculating $\mathbf{x}_{\mathrm{eq}}^{+}$ accurately and direct usage of (13) for the robustness analysis with respect to the parametric uncertainty, $\delta\mathbf{p}$, requires solving the following nonlinear algebraic equation:

$$\mathbf{f}(\mathbf{x}_{\mathrm{eq}}^{+}, \mathbf{p}) = \mathbf{f}_{\mathrm{OL}}(\mathbf{x}_{\mathrm{eq}}^{+}, \mathbf{p}) + \mathbf{g}(\mathbf{x}_{\mathrm{eq}}^{+}, \mathbf{p})\mathbf{u}(\mathbf{x}_{\mathrm{eq}}^{+}) = \mathbf{0} \tag{14}$$

and obtaining the jacobian of the control input at the perturbed equilibrium point for every $\delta\mathbf{p}$ as follows:

$$\left.\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}^{+}} \tag{15}$$

This would require additional computations and could be cumbersome, if not impossible, for some complex functions or mapping based controllers such as reinforcement learning or the neural network based control algorithm. As there might not be an explicit analytical expression for the control algorithm, calculating the derivative of every perturbation would increase further the computational cost for the robustness analysis.

To avoid the jacobian calculation of the control input for every perturbation, approximate the right-hand side of (13) using the Taylor series expansion up to the first-order terms as follows:

$$\delta\dot{\mathbf{x}} \approx \left[F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p}) + \sum_{i=1}^{n} \left.\frac{\partial F_x}{\partial x_{\mathrm{eq}}^{(i)}}\right|_{(\mathbf{x}_{\mathrm{eq}}, \mathbf{p})} \delta x_{\mathrm{eq}}^{(i)} \right] \delta\mathbf{x} \tag{16}$$

where $\partial(\cdot)/\partial x_{\mathrm{eq}}^{(i)}$ is the partial derivative with respect to the i-th component of $\mathbf{x}_{\mathrm{eq}}$, and $\delta x_{\mathrm{eq}}^{(i)}$ is the i-th component of $\delta\mathbf{x}_{\mathrm{eq}}$. Substitute (11) into (16)

$$\delta\dot{\mathbf{x}} = \left[F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p}) + \mathit{\Delta} F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p})\right] \delta\mathbf{x} = A(\delta\mathbf{p})\delta\mathbf{x} \tag{17}$$

where

$$\mathit{\Delta} F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p}) = \sum_{i=1}^{n} \left.\frac{\partial F_x}{\partial x_{\mathrm{eq}}^{(i)}}\right|_{(\mathbf{x}_{\mathrm{eq}}, \mathbf{p})} \delta x_{\mathrm{eq}}^{(i)}(\mathbf{x}_{\mathrm{eq}}, \bar{\mathbf{p}}) \tag{18a}$$

$$A(\delta\mathbf{p}) = F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p}) + \mathit{\Delta} F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p}) \tag{18b}$$

$\mathit{\Delta} F_x(\mathbf{x}_{\mathrm{eq}}, \mathbf{p})$ is caused by the effect of the parameter perturbations on the equilibrium point shift.

2.3 Robustness Analysis at $\mathbf{x}_{\mathbf{eq}}^{+}$

Define the difference matrix by the perturbation of $A(\delta\mathbf{p})$ as follows [5]:

$$A_{\mathit{\Delta}}(\delta\mathbf{p}) = A(\delta\mathbf{p}) - A(\mathbf{0}) \tag{19}$$

and the linearised system is given by

$$\delta\dot{\mathbf{x}} = A(0)\delta\mathbf{x} + A_{\Delta}(\delta\mathbf{p})\delta\mathbf{x} \tag{20}$$

The stability of the perturbed system is determined by the following transfer function:

$$G(s, \delta\mathbf{p}) = \left[I - M(s)A_{\Delta}(\delta\mathbf{p})\right]^{-1} M(s) \tag{21}$$

where s is the complex frequency variable in the Laplace transform,

$$M(s) = \left[sI - A(0)\right]^{-1} \tag{22}$$

and I is the identify matrix with the appropriate dimension.

The robustness analysis problem is seeking the minimum magnitude of the perturbation, $\|\delta\mathbf{p}\|$, among the following singularity is satisfied:

$$\|\delta\mathbf{p}^{*}\| = \operatorname*{argmin}_{\|\delta\mathbf{p}\|} |I - M(j\omega)A_{\Delta}(\delta\mathbf{p})| = 0 \tag{23}$$

for $\omega \in [0, \infty)$, where $j = \sqrt{-1}$. This is the μ-analysis problem, where μ is equal to the inverse of $\|\delta\mathbf{p}^{*}\|$. In the standard μ-analysis problem, the uncertainty is pulled out from $A(\delta\mathbf{p})$ and the inversion in (21) is given as $(I - M\Delta)^{-1}$, where Δ is a diagonal matrix for the real-valued parameter perturbation problem as this. This is only possible if the uncertainty appears in the polynomial form.

The uncertain parameters in the robustness analysis problem given in (21), however, cannot be separated from $A(\delta\mathbf{p})$ in general as they are not necessarily given in polynomial equations. For the non-polynomial form uncertainty structures, the sampling-based μ-analysis algorithm in [5], which is an improved algorithm originally presented in [6] and [7], is to solve the robustness analysis problem given in (23). The algorithm finds the intersection of the two hypersurfaces in the uncertain space using random samples defined by the following equation:

$$\Re\,|I - M(j\omega)A_{\Delta}(\delta\mathbf{p})| = 0 \tag{24a}$$

$$\Im\,|I - M(j\omega)A_{\Delta}(\delta\mathbf{p})| = 0 \tag{24b}$$

at ω in $[0, \infty)$, where $\Re(\cdot)$ and $\Im(\cdot)$ are the real and the imaginary part of the argument, respectively. The algorithm is to find the minimum $\|\delta\mathbf{p}\|$ that the singular conditions are met using a random sampling-based method.

3 Example: Inverted-Pendulum Stabilisation

The pendulum stabilisation is one of the benchmark problems for many control algorithms. The reinforcement learning is applied to the inverted-pendulum stabilisation problem shown in Figure 1 and the robustness analysis is performed. In the following, we use *exact bound* for the robustness bound of linearised systems and *true bound* for the true robustness bound of nonlinear systems.

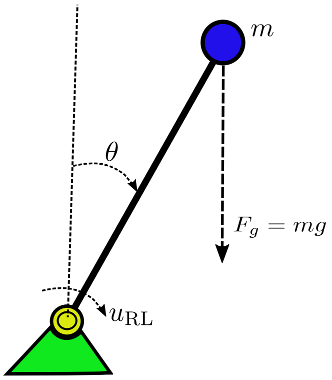


Fig. 1. A simple pendulum control problem

3.1 Dynamics & DDPG

OpenAI Gym provides a pendulum environment [8]. The discretized pendulum dynamics in the python source program corresponds to the following differential equation:

$$\ddot{\theta} = \frac{3g}{2\hat{\ell}} \sin\theta + \frac{3}{\hat{m}\hat{\ell}^2} u_{\mathrm{RL}} \tag{25}$$

where $\ddot{\theta} = d^2\theta/dt^2$, g is the gravitational acceleration equal to $9.81\mathrm{m/s}^2$, $\hat{m} = 1\mathrm{kg}$, $\hat{\ell} = 1\mathrm{m}$ and u_{RL} is the control input to be designed using the reinforcement learning approach and its magnitude is in the range of $[-2, 2]$ Nm.

The reinforcement learning for u_{RL} is trained using the python code in [9], where the DDPG in [10] is implemented in the code and the arguments to the control are as follows:

$$u_{\mathrm{RL}}(s_1, s_2, s_3) = u_{\mathrm{RL}}(\cos\theta, \sin\theta, \dot{\theta}) \tag{26}$$

where $s_1 = \cos\theta$, $s_2 = \sin\theta$, $s_3 = \dot{\theta}$ and $\dot{\theta} = d\theta/dt$.

Based on the freebody-diagram shown in Figure 1, the equation of motion must be, in fact, given by

$$\ddot{\theta} = \frac{g}{\ell} \sin\theta + \frac{1}{m\ell^2} u_{\mathrm{RL}} \tag{27}$$

To keep the reinforcement learning model for training u_{RL} the same as the original dynamics, redefine the nominal mass and length as follows: $\bar{m} = (3\hat{m})/4 = 0.75\mathrm{kg}$ and $\bar{\ell} = (2\hat{\ell})/3 = 0.667\mathrm{m}$. The true mass and length are defined with uncertainties as follows:

$$m = \bar{m} + \delta m \ [\mathrm{kg}], \ \ \ell = \bar{\ell} + \delta\ell \ [\mathrm{m}] \tag{28}$$

and $\delta\mathbf{p} = [\delta m, \ \delta\ell]^T$.

The instantaneous reward of the reinforcement learning is given by

$$R = -J = -\left(\theta^2 + \frac{\dot{\theta}^2}{10} + \frac{u_{\mathrm{RL}}^2}{1000}\right) \tag{29}$$

where R is the reward, J is the cost, and θ is in $[-\pi, \pi]$. It is also worth pointing out that the resulting control might not achieve the maximum reward, $R = 0$, at $\theta = 0$, $\dot{\theta} = 0$ and $u_{\mathrm{RL}} = 0$. The equilibrium achieved by u_{RL} satisfies

$$u_{\mathrm{RL}}(\cos\theta_{\mathrm{eq}}, \sin\theta_{\mathrm{eq}}, \dot{\theta}_{\mathrm{eq}} = 0) = -mg\ell \sin\theta_{\mathrm{eq}} \tag{30}$$

3.2 Linearization

The stability analysis is to check the eigenvalues of the Jacobian, $F_x(\mathbf{x}, m, \ell)$, at the equilibrium point as follows:

$$F_x(\mathbf{x}_{\mathrm{eq}}, m, \ell) = \left.\frac{d\mathbf{f}(\mathbf{x}, \mathbf{m}, \ell)}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{31}$$

where

$$\mathbf{f}(\mathbf{x}, m, \ell) = \begin{bmatrix} \dot{\theta} \\ \frac{g}{\ell}\sin\theta + \frac{1}{m\ell^2}u_{\mathrm{RL}} \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \ \mathbf{x}_{\mathrm{eq}} = \begin{bmatrix} \theta_{\mathrm{eq}} \\ 0 \end{bmatrix} \tag{32a}$$

The Jacobian is given by

$$F_x(\mathbf{x}_{\mathrm{eq}}, m, \ell) = \begin{bmatrix} 0 & 1 \\ F_{21} & F_{22} \end{bmatrix} \tag{33}$$

where

$$F_{21} = \left.\frac{df_2}{d\theta}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} = \frac{g}{\ell}\cos\theta_{\mathrm{eq}} + \frac{1}{m\ell^2}\left.\frac{du_{\mathrm{RL}}}{d\theta}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{34a}$$

$$F_{22} = \left.\frac{df_2}{d\dot{\theta}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} = \frac{1}{m\ell^2}\left.\frac{du_{\mathrm{RL}}}{d\dot{\theta}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{34b}$$

and

$$\frac{du_{\mathrm{RL}}}{d\theta} = -\frac{du_{\mathrm{RL}}}{ds_1}\sin\theta + \frac{du_{\mathrm{RL}}}{ds_2}\cos\theta \tag{35a}$$

$$\frac{du_{\mathrm{RL}}}{d\dot{\theta}} = \frac{du_{\mathrm{RL}}}{ds_3} \tag{35b}$$

The control input derivative with respect to s_i at the equilibrium point is obtained numerically. TensorFlow, for example, has a function to calculate the derivatives with respect to the input variables. The jacobian with respect to the uncertain parameters is given by

$$F_p(\mathbf{x}_{\mathrm{eq}}, \bar{\mathbf{p}}) = \begin{bmatrix} 0 & 0 \\ -\dfrac{1}{\bar{m}^2\bar{\ell}^2}\bar{u}_{\mathrm{RL}} & -\dfrac{g}{\bar{\ell}^2}\sin\theta_{\mathrm{eq}} - \dfrac{2}{\bar{m}\bar{\ell}^3}\bar{u}_{\mathrm{eq}} \end{bmatrix} \tag{36}$$

where $\bar{u}_{\mathrm{RL}}$ is the control input for the equilibrium point with the nominal values of the uncertain parameters, i.e., the solution of (30) with $m = \bar{m}$ and $\ell = \bar{\ell}$.

The second-derivatives are obtained as

$$\frac{\partial F_{21}}{\partial \theta_{\mathrm{eq}}} = -\frac{g}{\ell}\sin\theta_{\mathrm{eq}} + \frac{1}{m\ell^2}\left.\frac{\partial^2 u_{\mathrm{RL}}}{\partial \theta^2}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{37a}$$

$$\frac{\partial F_{22}}{\partial \theta_{\mathrm{eq}}} = \frac{\partial F_{21}}{\partial \dot{\theta}_{\mathrm{eq}}} = \frac{1}{m\ell^2}\left.\frac{\partial^2 u_{\mathrm{RL}}}{\partial \theta \partial \dot{\theta}}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{37b}$$

$$\frac{\partial F_{22}}{\partial \dot{\theta}_{\mathrm{eq}}} = \frac{1}{m\ell^2}\left.\frac{\partial^2 u_{\mathrm{RL}}}{\partial \dot{\theta}^2}\right|_{\mathbf{x}=\mathbf{x}_{\mathrm{eq}}} \tag{37c}$$

The second-derivative of the control input is obtained as

$$\frac{\partial^2 u_{\mathrm{RL}}}{\partial \theta^2} = \frac{\partial^2 u_{\mathrm{RL}}}{\partial s_1^2}(\sin\theta)^2 + \frac{\partial^2 u_{\mathrm{RL}}}{\partial s_2^2}(\cos\theta)^2 - \frac{\partial u_{\mathrm{RL}}}{\partial s_1}\cos\theta - \frac{\partial u_{\mathrm{RL}}}{\partial s_2}\sin\theta \tag{38a}$$

$$\frac{\partial^2 u_{\mathrm{RL}}}{\partial \dot{\theta}^2} = \frac{\partial^2 u_{\mathrm{RL}}}{\partial s_3^2} \tag{38b}$$

$$\frac{\partial^2 u_{\mathrm{RL}}}{\partial \theta \partial \dot{\theta}} = -\frac{\partial^2 u_{\mathrm{RL}}}{\partial s_1 \partial s_3}\sin\theta + \frac{\partial^2 u_{\mathrm{RL}}}{\partial s_2 \partial s_3}\cos\theta \tag{38c}$$

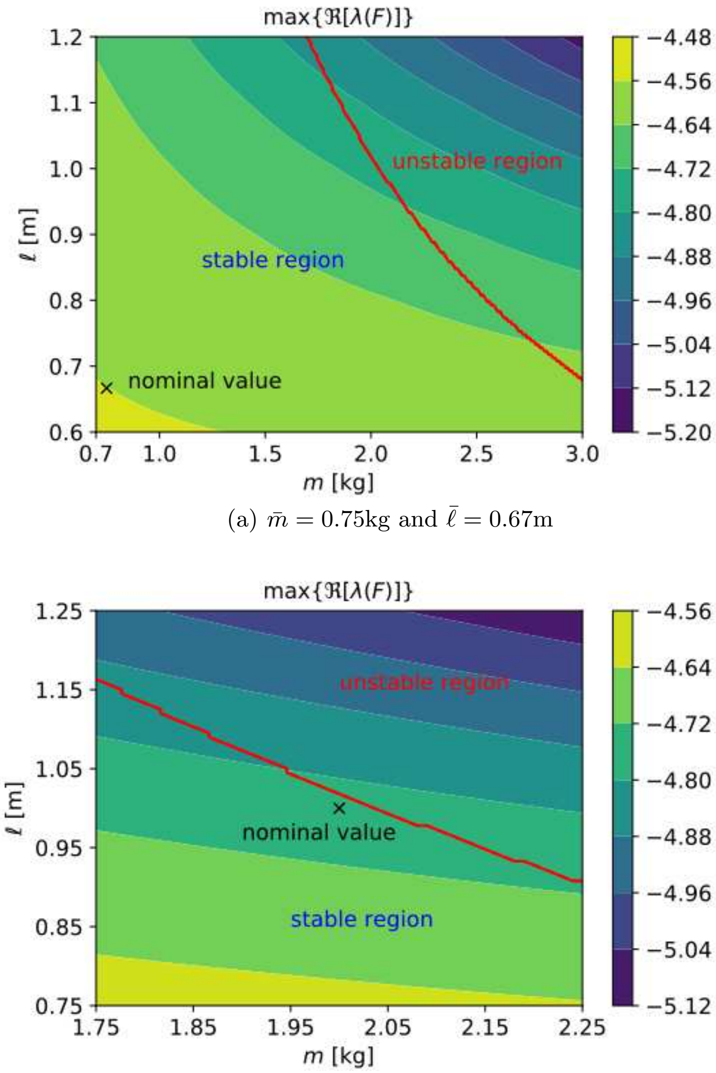
Similar to the first-derivative of u_{RL}, the Tensorflow function calculates the second-derivatives of u_{RL} with respect to s_1, s_2 and s_3. Now, we have all necessary derivatives to obtain (11) and (17), and we are ready to perform the robustness analysis for the pendulum system with the reinforcement learning.

3.3 Worst Perturbation with Usual Linearization (6)

The exact worst perturbation for the pendulum system with the usual linearisation approach, (6), is to be obtained. The nominal values for the mass and the length of the pendulum are the ones used to train the reinforcement learning algorithm, i.e., $\bar{m} = 0.75$kg and $\bar{\ell} = 0.67$m. The true mass and the true length are perturbed as follows:

$$m = 0.75 + \delta m \text{ [kg]}, \;\; \ell = 0.67 + \delta\ell \text{ [m]} \tag{39}$$

The stability of the perturbed system from the nominal value shown in Figure 2(a) is of the ranges in 0.7kg $\leq m \leq$ 3kg and 0.75m $\leq \ell \leq$ 1.25kg. The true stable and the unstable regions are divided by the red solid line, which is obtained by the linearised system at the corresponding perturbed m and ℓ, hence, including



(a) $\bar{m} = 0.75$kg and $\bar{\ell} = 0.67$m

(b) $\bar{m} = 2$kg and $\bar{\ell} = 1$m.

Fig. 2. The usual linearization (6) is used for the stability check. This figure shows the whole regions for both of the nominal value cases are stable, i.e., the largest real part of the eigenvalues is negative. The contours show the largest real part of the eigenvalues of the perturbed system given by (6) for each nominal value case. (b) is zoomed-in to show the proximity of the nominal value to the unstable boundary. The red line shows how the true stable and unstable regions are divided.

the true effect of equilibrium shift. The largest real part of the eigenvalues of the perturbed linearised system for each perturbation is calculated and the contour plot is shown in Figure 2(a). The exact worst perturbations for the linearised system is calculated, but it provides the conclusion that the system is robustly stable for all perturbation in the region, while the significant area in the uncertain space is, in fact, unstable.

The incorrect robust stability analysis cannot be fixed by adjusting the nominal values. Set the nominal values very close to the unstable region, for example, $\bar{m} = 2$kg and $\bar{\ell} = 1$m. The robustness analysis shows that the linearised system is stable in the same whole region of perturbation and Figure 2(b) shows the zoom-in contour around the new nominal value.

3.4 Robustness Analysis with Improved Linearization (17)

The improved linearisation of the pendulum closed-loop system with the reinforcement learning is given by

$$A(\delta\mathbf{p}) = F_x(\mathbf{x}_{\text{eq}}, m, \ell) + \frac{\partial F_x(\mathbf{x}_{\text{eq}}, m, \ell)}{\partial \theta_{\text{eq}}}\delta\theta_{\text{eq}} + \frac{\partial F_x(\mathbf{x}_{\text{eq}}, m, \ell)}{\partial \dot{\theta}_{\text{eq}}}\delta\dot{\theta}_{\text{eq}} \tag{40}$$

where

$$\begin{bmatrix} \delta\theta_{\text{eq}} \\ \delta\dot{\theta}_{\text{eq}} \end{bmatrix} = -F_x^{-1}([\theta_{\text{eq}}, \dot{\theta}_{\text{eq}}]^T, [\bar{m}, \bar{\ell}]^T) F_p([\theta_{\text{eq}}, \dot{\theta}_{\text{eq}}]^T, [\bar{m}, \bar{\ell}]^T) \begin{bmatrix} \delta m \\ \delta p \end{bmatrix} \tag{41}$$

Note that $\dot{\delta}_{\text{eq}}$ is always equal to zero by the definition of the equilibrium point.

The robustness of the pendulum system is performed using (24). For $\omega = 0$, the boundaries, where the real part sign change occurs, are the singular lines as shown in Figure 3. In this case, the imaginary part is zero over the whole perturbation space. The exact singular point, which is the closest point from the nominal value to the singular line in the ∞-norm sense, is indicated by the filled circles. The worst-case perturbation is the size of the smallest square box centred at $m = 2$ kg and $\ell = 1$ m contacted the singular lines or at least one of the singular points. The square box is elongated in the vertical axis because of the non-equal scale used. As shown in Figure 3, the exact robustness bound, $\delta\mathbf{p}^*$ for the improved linearised system with the nominal values, $\bar{m} = 2$ kg and $\bar{\ell} = 1$ m, is about 1.7.

The robustness analysis algorithm in [5] with the improved linearised system at $\omega = 0$ gives the bound around 1.9, which is relatively tight to the exact value. For the other frequencies, the bounds are a lot bigger than the one found at $\omega = 0$.

4 Conclusions

The destabilising uncertainty bounds with the linearisation approaches are frequently optimistic. This is the limitation of linearised approach itself. Therefore,

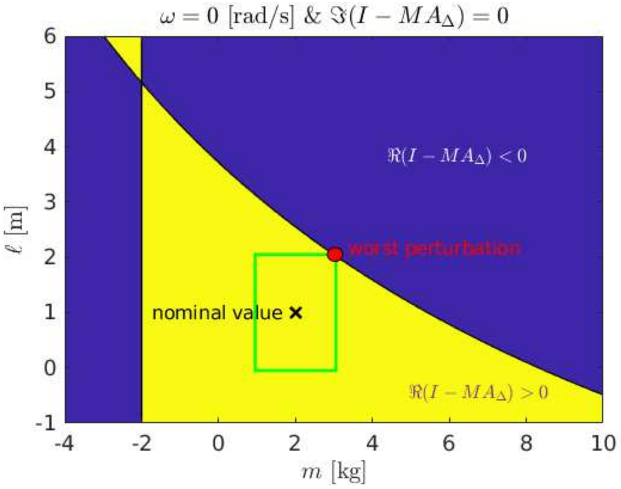


Fig. 3. Singular lines and the exact worst perturbation for $\omega = 0$

the worst-case perturbation found must be used as the upper bound on the minimum magnitude worst-case perturbation. The new method provides a lot closer bound to the true value than the usual linearisation method.

The smallest bound, 1.9, is still far from the true bound for the nonlinear system, which is about 0.2, the closest distance between the nominal value and the red line in Figure 2(b). This is not the limitation of the robustness analysis algorithm but the limitation of the linearised model itself. The discrepancy between the estimated and the true is mainly caused by the large slope of the control function u_{RL} with respect to the states. The implemented reinforcement learning has abrupt changes in the input causing the large equilibrium shifts. Restricting these abrupt changes would improve the robustness of the system. Nevertheless, the proposed model significantly improves the original bound by including the equilibrium shift.

Acknowledgement

This research was supported by Unmanned Vehicles Core Technology Research and Development Program (No. 2020M3C1C1A0108316111) through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center(UVARC) funded by the Ministry of Science and ICT, Republic of Korea.

References

1. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)

2. Young, P.M., Newlin, M.P., Doyle, J.C.: μ analysis with real parametric uncertainty. (1991)
3. Andersson, L., Rantzer, A.: Robustness of equilibria in nonlinear systems. IFAC Proceedings Volumes **32**(2) (1999) 2256–2261
4. Paulino, N.M., Foo, M., Kim, J., Bates, D.G.: Robustness analysis of a nucleic acid controller for a dynamic biomolecular process using the structured singular value. Journal of Process Control **78** (2019) 34–44
5. Darlington, A.P., Kim, J., Bates, D.G.: Robustness analysis of a synthetic translational resource allocation controller. IEEE control systems letters **3**(2) (2018) 266–271
6. Zhao, Y.B., Kim, J., Bates, D.G.: Lft-free μ-analysis of lti/lptv systems. In: 2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD), IEEE (2011) 638–643
7. Kim, J., Bates, D.G., Postlethwaite, I.: A geometrical formulation of the μ-lower bound problem. IET control theory & applications **3**(4) (2009) 465–472
8. OpenAI gym, pendulum-v0 (2020)
9. Deep deterministic policy gradient (DDPG) (2020)
10. Cheng, R., Verma, A., Orosz, G., Chaudhuri, S., Yue, Y., Burdick, J.: Control regularization for reduced variance reinforcement learning. In: International Conference on Machine Learning, PMLR (2019) 1141–1150