Goal-Directed Reward Generation

Alymbek Sadybakasov
Autonomous Systems
TU Darmstadt
alymbek.sadybakasov@stud.tu-darmstadt.de

Boris Belousov
IAS
TU Darmstadt
boris@robot-learning.de

Abstract

Optimal control framework provides a way to generate a controller by specifying an immediate reward function and subsequently minimizing the sum of rewards along a trajectory. To every controller, there is an associated value function—derived from dynamics and the reward function—that can be viewed as a measure of goodness of the controller. Thus, what one really cares about is to synthesize a controller that produces desired trajectories, and different controllers are ranked using their value functions. In this paper, we consider the feasibility of constructing the value function directly, for example as a Lyapunov function of a given nonlinear system, and subsequently generating a controller from it. Using the method of ideal Lyapunov functions combined with the method of formal linearization, we are able to generate a nonlinear controller for the mountain car problem without torque limits by formally applying linear-quadratic control techniques. Experiments confirm superior performance of the resulting nonlinear controller, in particular of the regularized and the damped nonlinear controller. The positive results suggest that the current literature approach to controller synthesis based on the Lyapunov theory can be potentially rather useful.

1 Introduction

Most robotic as well as biological systems are nonlinear and have nonlinear dynamics, and their analysis is usually done using Lyapunov functions. The Lyapunov functions (Lyapunov [1992]) are widely used in the stability analysis of nonlinear systems. A Lyapunov function can also be seen as an optimal value function that correspond to a certain reward function. This value function is then used to construct the corresponding control law. The relationship of such controllers to value functions was shown by Todorov [2018]. This observation motivates the following: instead of generating a controller from a reward function, we explore the possibility of generating a nonlinear controller from a Lyapunov function using the desired goal state. Possible success results produced by the resulting nonlinear controller will give a hint to an existing connection between value and Lyapunov functions. Since the Lyapunov analysis of a nonlinear system is not a trivial task, a corresponding linearized system offers itself to be used. Itschner [1977] proposed a simplified procedure to construct an "ideal Lyapunov function" (IL-function). Subsequently, Sieber [1989] used these functions to model a nonlinear controller by aligning the produced dynamics with the dynamics produced by a reference linear controller using linearized system dynamics. The nonlinearity of the resulting controller can potentially encode a very expressive nonlinear behaviour of control inputs while still having similar trajectories of the states as those produced by the reference linear controller. This consideration motivates us to consider designing a nonlinear controller for a dynamic system with perfectly known dynamics and to learn later a controller assuming no knowledge of the system dynamics.

In this paper, we first provide the related work on nonlinear controllers and then show how a nonlinear system can be transformed into a linear system and then controlled using a Linear Quadratic Regulator (LQR). Afterwards, we provide the necessary knowledge on the Lyapunov analysis, i.e. how to construct the Lyapunov functions for linear systems and their usefulness for analyzing the stability of the system. We then describe how to construct a nonlinear controller by aligning its dynamics with the dynamics of a reference linear controller and discuss our evaluations and further work.

2 Related Work

Most work on nonlinear control is done by analyzing the stability of the systems and deriving the appropriate nonlinear controllers which are based on the linearization of the state feedback. However, such methods become unfeasible as long as the number of degrees of freedom (DoF) of the system increases which gives an intuition to use rather numerical techniques. For example, Sieber [1994] used an observer to construct the nonlinear controller from a linear observer. The method is especially useful when the state variables are not completely observable. Bemporad et al. [2002] derived an LQR for constrained systems, e.g. with limited torques. Similarly, Johansen et al. [2000] derived a suboptimal LQR constrained systems. Both controllers can be viewed as nonlinear due to their piece-wise linearization.

Sieber [1989] used a full-state feedback (FSF) controller (Ackermann [1977]) to construct the reference linear controller by using placement of the predefined poles. Instead of using the FSF-controller, we evaluate the usefulness of LQR as a reference linear controller. In particular, LQR allows us to define different cost functions which will affect the resulting trajectory of both reference linear and nonlinear controller. In addition, we evaluate the usage of a regularized pseudo-inverse to compute the gain matrix of the controller which helps us to avoid numerical instabilities. Moreover, we improve the regular nonlinear controller by damping the control gain matrix.

3 Constructing a Reference Linear Controller

As mentioned before, we first need to linearize the nonlinear system in order to construct a linear controller. The following subsection illustrates the process of linearization of a nonlinear system. Subsequently, a concept of LQR is introduced which will be used later as a reference linear controller.

3.1 Linearization of Nonlinear Systems

Consider the following system

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{B}\mathbf{u},$$

where $f(\mathbf{x})$ is a nonlinear function that depends on $\mathbf{x}$ and $\mathbf{B}$ is a constant matrix. This system is underactuated due to the possibility of changing its behavior by applying some torque $\mathbf{u}$. An example of such system is the electrical generator that produces direct current using a commutator, also known as dynamo (Keller [1986]).

In order to control such system with a linear controller, one linearizes the nonlinear parts of the system around the equilibrium point.

We rewrite the nonlinear function as

$$f(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x},$$

and, thus, have

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{1}$$

Although the resulting equation is formally linear, the matrix $\mathbf{A}(\mathbf{x})$ still depends on $\mathbf{x}$. Thus, we rewrite the controller as

$$\mathbf{u} = -\mathbf{K}_{\mathbf{lin}}\mathbf{x}$$

and substitute it into (1), obtaining

$$\dot{\mathbf{x}} = [\mathbf{A}(\mathbf{0}) - \mathbf{B}\mathbf{K}_{\mathbf{lin}}]\mathbf{x} = [\mathbf{A}_{\mathbf{lin}} - \mathbf{B}\mathbf{K}_{\mathbf{lin}}]\mathbf{x}. \tag{2}$$

The last equation is obtained by linearizing around the equilibrium point using Taylor expansion and neglecting higher order terms. The resulting system is fully linear and can be controlled using standard linear controllers.

3.2 LQR

One way to control linearized dynamical systems is to use the cost function described by a quadratic function. Several variations on defining the quadratic cost exist. We restrict us to the following definition of the quadratic cost:

$$\mathbf{J} = \int_{\mathbf{0}}^{\infty} (\mathbf{x^T Q x} + \mathbf{u^T R u}) dt,$$

where $\mathbf{Q} = \mathbf{Q^T} \geq \mathbf{0}$ denotes the state cost and $\mathbf{R} = \mathbf{R^T} \geq \mathbf{0}$ denotes the control cost. The design of the matrices $\mathbf{Q}$ and $\mathbf{R}$ is an essential part of constructing LQR. The higher is $\mathbf{Q}$, the more dynamical the system becomes, while the higher is $\mathbf{R}$, the more smooth the control signal has to be while reaching the target point.
To obtain the gain matrix $\mathbf{K_{lin}}$ from (2), we minimize $\mathbf{J}$ by solving numerically the following algebraic Riccati equation over $\mathbf{P}$:

$$\begin{aligned} &\mathbf{A_{lin}^T P} + \mathbf{P A_{lin}} - \mathbf{P B R^{-1} B^T P} + \mathbf{Q} = 0 \\ &\mathbf{K_{lin}} = \mathbf{R^{-1} B^T P}. \end{aligned}$$

Once the matrix $\mathbf{K_{lin}}$ is found, the control input signal is computed online using $\mathbf{u} = -\mathbf{K_{lin} x}$.

4 Lyapunov Theory

Lyapunov functions are widely used in the analysis of dynamical systems. Having a suitable Lyapunov function allows us to give statements about the global asymptotic stability of the system. In the following subsections, we introduce the Lyapunov functions and one possible way to construct them.

4.1 Lyapunov Functions

Consider the following dynamical system

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \tilde{\mathbf{A}}\mathbf{x},$$

which has an equilibrium point at $\mathbf{x} = 0$. A Lyapunov function is then any positive definite function $\mathbf{V}(\mathbf{x})$ around the equilibrium point of the system whose negative derivatives are positive definite as well. Given these properties, one can make statements on the asymptotic stability of the equilibrium points. This method is also called a *Lyapunov's direct method.* The term direct method is used since the stability problem is investigated directly instead of solving the state differential equations.

4.2 Constructing Lyapunov Functions for Linear Systems

We reconsider the system $\dot{\mathbf{x}} = (\mathbf{A_{lin}} - \mathbf{B K_{lin}})\mathbf{x}$, where $\mathbf{K_{lin}}$ is the gain matrix computed by LQR. In order to apply Lyapunov's direct method, we consider a quadratic Lyapunov function

$$\mathbf{V} = \mathbf{x^T P x}, \tag{3}$$

with its time derivative

$$\dot{\mathbf{V}} = \mathbf{x^T P \dot{x}} + \mathbf{\dot{x}^T P x}.$$

Setting the derivative of the system into the above equation yields

$$\dot{\mathbf{V}} = \mathbf{x^T P A_{lin} x} + \mathbf{x^T A_{lin}^T P x},$$

and, thus,

$$\begin{aligned} \dot{\mathbf{V}} &= \mathbf{x^T}(-\mathbf{Z_{lin}})\mathbf{x}, \\ -\mathbf{Z_{lin}} &= \mathbf{P A_{lin}} + \mathbf{A_{lin}^T P}, \end{aligned} \tag{4}$$

where $\mathbf{Z_{lin}}$ is a symmetric and positive definite matrix. The second equation from (4) is a so called Lyapunov equation which is usually used to prove the asymptotic stability around the equilibrium point $\mathbf{x_E}$.

Theorem 4.1 *Let Z_{lin} be a symmetric and positive definite matrix, then there exists a clearly determined symmetric and positive solution for P from (4), if the eigenvalues of the matrix A lie at the left side of the j-Axis of the complex plane.*

A proof of 4.1 was derived by LaSalle and Lefschetz [1967]. Given the conditions of 4.1 and that $\mathbf{Z_{lin}}$ can be any symmetric and positive definite matrix, a number of numerical techniques were established to compute the matrix $\mathbf{P}$. However, we are interested in finding rather the matrix $\mathbf{Z_{lin}}$ which we will use to construct the nonlinear controller. According to Itschner [1977], we find $\mathbf{Z_{lin}}$ using a predefined matrix $\mathbf{P}$. We define the matrix $\mathbf{P}$ from (3) as a constant, symmetric and positive definite matrix (e.g. a diagonal matrix with positive elements). $\mathbf{Z_{lin}}$ can be then found as

$$\mathbf{Z_{lin}} = -(\mathbf{P}\mathbf{A_{lin}} + \mathbf{A_{lin}^T}\mathbf{P}),$$

having in mind that $\dot{\mathbf{V}} = \mathbf{x^T}(-\mathbf{Z_{lin}})\mathbf{x}$ stays negative definite. The crucial part of this method is that $\mathbf{P}$ depends on the system matrix $\mathbf{A_{lin}}$ in the following way:

$$\mathbf{P} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\tilde{P}_{ii}}\mathbf{\bar{w}_i}\mathbf{w_i}, \tag{5}$$

where $\mathbf{\tilde{P}}$ is any positive diagonal matrix, $\mathbf{w_i}$ are the left eigenvectors of $\mathbf{A_{lin}}$ and $\mathbf{\bar{w}_i}$ are their corresponding complex conjugate vectors. $\mathbf{P}$ is, thus, positive definite, has real values and can be now used to construct the nonlinear controller.

5 Nonlinear Controller

We consider now the nonlinear system:

$$\dot{\mathbf{x}} = [\mathbf{A}(\mathbf{x}) - \mathbf{B}\mathbf{K}(\mathbf{x})]\mathbf{x}.$$

The system has a formally linear form, in which the system matrix $\mathbf{A}(\mathbf{x})$ depends directly on $\mathbf{x}$, whereas the regularization matrix $\mathbf{K}(\mathbf{x})$ depends on $\mathbf{x}$. The equilibrium point of the system is given by $\mathbf{x_E} = \mathbf{0}$.
Suppose now that we are interested in finding such a trajectory that "transports" the system's state variable $\mathbf{x}(\tau)$ from one equilibrium point to another. In Section 3.2, we have already seen how to achieve this goal by linearizing the system around the target point. We will use the same linearized system to construct a nonlinear controller by aligning its dynamics with the dynamics of the reference linear controller and using the Lyapunov stability theory.
We first construct a quadratic Lyapunov function $\mathbf{V} = \mathbf{x^T}\mathbf{P}\mathbf{x}$ as given in (3) for the linearized system. Its derivative is given by

$$\dot{\mathbf{V}}_{\mathbf{lin}} = -\mathbf{x^T}\mathbf{Z_{lin}}\mathbf{x} = \mathbf{x^T}(-\mathbf{Z_{lin}})\mathbf{x},$$

where

$$\mathbf{Z_{lin}} = [\mathbf{K_{lin}^T}\mathbf{B^T} - \mathbf{A_{lin}^T}]\mathbf{P} + \mathbf{P}[\mathbf{B}\mathbf{K_{lin}} - \mathbf{A_{lin}}]. \tag{6}$$

It should be noted that $\mathbf{Z_{lin}}$ was written explicitly as $-\mathbf{Z_{lin}}$. Thus, by definition of Lyapunov functions, the following rule has to be considered: if $\dot{\mathbf{V}}$ is negative definite, $\mathbf{Z_{lin}}$ should stay positive definite.
We construct now the Lyapunov function for the nonlinear systems by choosing the same quadratic function as given by (3). For a nonlinear system, the time derivative of its Lyapunov function is now different:

$$\dot{\mathbf{V}}(\mathbf{x}) = -\mathbf{x^T}\mathbf{Z}(\mathbf{x})\mathbf{x} = \mathbf{x^T}[-\mathbf{Z}(\mathbf{x})]\mathbf{x},$$

where the matrix $\mathbf{Z}(\mathbf{x})$ depends now on $\mathbf{x}$:

$$\mathbf{Z}(\mathbf{x}) = [\mathbf{K}(\mathbf{x})^\mathbf{T}\mathbf{B^T} - \mathbf{A}(\mathbf{x})^\mathbf{T}]\mathbf{P} + \mathbf{P}[\mathbf{B}\mathbf{K}(\mathbf{x}) - \mathbf{A}(\mathbf{x})]. \tag{7}$$

If we assume that the linear controller of the linear function is optimal, we can also say that a nonlinear controller that aligns its dynamics with the resulting dynamics of the linear controller is optimal as well.
It remains to show how the actual aligning works exactly. Due to optimality of $\mathbf{K_{lin}}$, $\mathbf{V_{lin}}$ strictly decreases, $\dot{\mathbf{V}}_{\mathbf{lin}}$ becomes strictly negative and the system, thus, converges to the equilibrium point

rapidly. To obtain a nonlinear controller with similar dynamics, we intuitively align $\dot{\mathbf{V}}(\mathbf{x})$ with $\dot{\mathbf{V}}_{\mathbf{lin}}$. For this purpose, the difference

$$\mathbf{x}^{\mathbf{T}}\mathbf{Z}_{\mathbf{lin}}\mathbf{x} - \mathbf{x}^{\mathbf{T}}\mathbf{Z}(\mathbf{x})\mathbf{x} = \mathbf{x}^{\mathbf{T}}[\mathbf{Z}_{\mathbf{lin}} - \mathbf{Z}(\mathbf{x})]\mathbf{x}$$

is minimized, leading to the minimization of the following norm:

$$N \equiv ||\mathbf{Z}_{\mathbf{lin}} - \mathbf{Z}(\mathbf{x})||. \tag{8}$$

The minimization is performed by choosing a suitable gain matrix $\mathbf{K}(\mathbf{x})$ so that $N = 0$, inducing $\mathbf{Z}_{\mathbf{lin}} = \mathbf{Z}(\mathbf{x})$. After mathematical reformulations, we obtain

$$\mathbf{M} \cdot \mathrm{vec}(\mathbf{K}(\mathbf{x})) = \mathrm{vec}(\mathbf{Z}_{\mathbf{lin}}) + \mathbf{J} \cdot \mathrm{vec}(\mathbf{A}(\mathbf{x})), \tag{9}$$

where vec($\mathbf{M}$) denotes a vectorization operator over the matrix $\mathbf{M}$ and

$$\mathbf{J} = (\mathbf{I} \otimes \mathbf{P}) + (\mathbf{P} \otimes \mathbf{I})\mathbf{U}_{n\times n},$$
$$\mathbf{M} = \mathbf{I} \otimes (\mathbf{PB}) + [(\mathbf{PB}) \otimes I]\mathbf{U}_{p\times n},$$

where $\mathbf{U}_{a\times b}$ denotes a permutation matrix.
Finally, solving (9) for the vectorized gain matrix vec($\mathbf{K}(\mathbf{x})$) yields

$$\mathrm{vec}(\mathbf{K}(\mathbf{x})) = \mathbf{M}^{+}[\mathrm{vec}(\mathbf{Z}_{\mathbf{lin}}) + \mathbf{J} \cdot \mathrm{vec}(\mathbf{A}(\mathbf{x})], \tag{10}$$

where $\mathbf{M}^{+}$ stands for Moore–Penrose inverse. The resulting controller $\mathbf{u} = \mathbf{K}(\mathbf{x})\mathbf{x}$ aligns the dynamics of the nonlinear controller with the dynamics of a reference linear controller.

6 Evaluations

We evaluate the described nonlinear controller on a classical nonlinear system mountain car ((Sutton and Barto [1998])) with several modifications. We provide a derivation of the gain matrices for both LQR and nonlinear controller as well as simulations of the resulting controllers. In addition, we improve the performance of the regular nonlinear controller by using a regularized pseudo-inverse and by damping the gain matrix $\mathbf{K}(\mathbf{x})$.

6.1 Environment

The problem of mountain car is described as follows:

$$\dot{x_1} = x_2,$$
$$\dot{x_2} = -0.0025\cos(3x_1) + 0.001u,$$

where x_1 and x_2 denote the position and the velocity respectively. The constraints of the system are given by $u \in [-1, 1]$ and $x_2 \in [-0.07, 0.07]$. To analyze this problem, we consider it as a first-order dynamic system and rewrite in the matrix form:

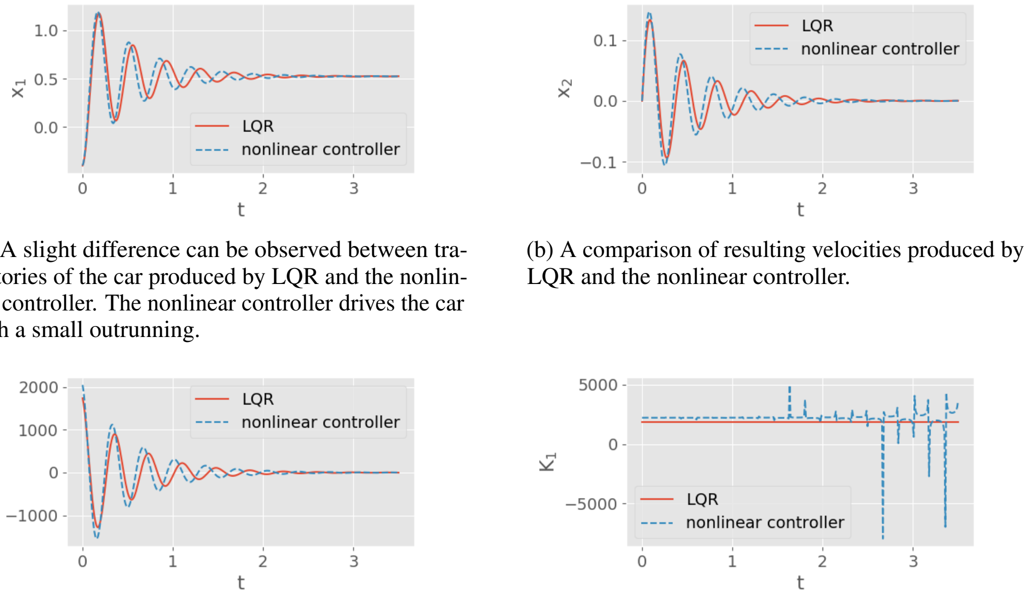
$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ \frac{-0.0025\cdot\cos(3\cdot x_1)}{x_1} & 0 \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 0 \\ 0.001 \end{bmatrix}, \tag{11}$$
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

bringing it to the formally linear form.
We relax the constraints of the system by not clipping the state variables, i.e. $u \in [-\infty, +\infty]$, $x_1 \in [-\infty, +\infty]$ and $x_2 \in [-\infty, +\infty]$. This assumption lets us to use LQR and the nonlinear controller without any restrictions.
To drive the car to a desired position, we set

$$\mathbf{\Delta x} = \mathbf{x} - \mathbf{x_E} \implies \mathbf{x} = \mathbf{\Delta x} + \mathbf{x_E},$$
$$\delta u = u - u_E \implies u = \delta u + u_E.$$

We substitute the above terms into (11) and obtain a system with the following dynamics:

$$\mathbf{A}(\mathbf{\Delta x}) = \begin{bmatrix} 0 & 1 \\ -\frac{0.0025\cdot\cos(3\cdot\delta x_1 + 3\cdot x_{1E})}{x_1} & 0 \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 0 \\ 0.001 \end{bmatrix}.$$



(a) A slight difference can be observed between trajectories of the car produced by LQR and the nonlinear controller. The nonlinear controller drives the car with a small outrunning.

(b) A comparison of resulting velocities produced by LQR and the nonlinear controller.

(c) A comparison of resulting torques produced by LQR and the nonlinear controller. A slightly higher initial gain of the nonlinear controller leads the car slightly faster to the goal position.

(d) The trend of the first element of the gain matrices. We can observe that the gain matrix of the regular nonlinear controller tends to the gain matrix of the nonlinear controller.

Figure 1: A comparison between the regular nonlinear controller and LQR. The initial conditions are given for both controllers as $\mathbf{x_0} = [-0.4, 0]^T$ and $\mathbf{x_E} = [\frac{\pi}{6}, 0]^T$.

6.2 Nonlinear Controller

To construct a nonlinear controller, we need a reference linear controller, in particular LQR. We choose $\mathbf{Q} = \mathbf{I} \cdot 35600$ and $R = 0.01$ to drive the car to the goal position very rapidly. The gain matrix has been computed by solving Riccati equations, yielding $\mathbf{K_{lin}} = [1881.80285139, 2463.54834913]$. To compute an example trajectory, we set $\mathbf{x_E} = [\frac{\pi}{6}, 0]^T$ and $\mathbf{x_0} = [-0.4, 0]^T$. The resulting trajectory is illustrated in Figure 1. Figure 1a shows that the trend of positions is smooth and reaches the desired equilibrium point within a few seconds. Having such dynamics was a requirement for constructing a nonlinear controller, as mentioned in Section 5.

To construct a nonlinear controller, we first construct a Lyapunov function. For this purpose, we set $\tilde{\mathbf{P}}$ as an identity matrix of size 2 and use (5) to gather the matrix $\mathbf{P}$. $\mathbf{Z_{lin}}, \mathbf{J}$ and $\mathbf{M}$ are computed in the next place and the controller's gain matrix is computed according to (10). Subsequently, the controller is computed using $u = -\mathbf{K}(\boldsymbol{\Delta}\mathbf{x})\boldsymbol{\Delta}\mathbf{x}$.

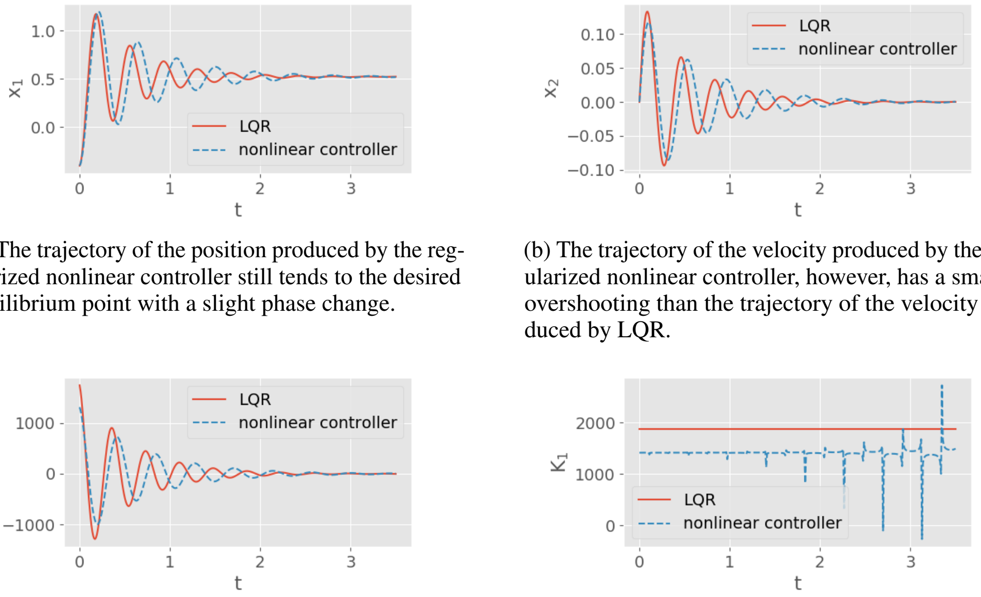
The controller's gain matrix $\mathbf{K}(\boldsymbol{\Delta}\mathbf{x})$ can be actually viewed as a linear combination of the elements of the matrix $\mathbf{A}(\boldsymbol{\Delta}\mathbf{x})$ as well, i.e.:

$$\begin{aligned}\mathbf{K}(\boldsymbol{\Delta}\mathbf{x}) = &\begin{bmatrix} a_{12}\mathbf{A}(\boldsymbol{\Delta}\mathbf{x})_{21} + a_{13} + \alpha_1 \\ \alpha_{22}\mathbf{A}(\boldsymbol{\Delta}\mathbf{x})_{21} + \alpha_{23} + \alpha_2 \end{bmatrix}^T, \\ &\mathbf{a} = \mathbf{M}^{+}\mathbf{J}, \\ &\alpha = \mathbf{M}^{+}\mathbf{Z_{lin}}.\end{aligned}$$

A nonlinear controller is, thus, given by

$$u = -\mathbf{K}(\boldsymbol{\Delta}\mathbf{x})\boldsymbol{\Delta}\mathbf{x} = -a_{12}\mathbf{A}(\boldsymbol{\Delta}\mathbf{x})_{21}\delta x_1 - (a_{13} + \alpha_1)\delta x_1 - \alpha_{22}\mathbf{A}(\boldsymbol{\Delta}\mathbf{x})_{21}\delta x_2 - (\alpha_{23} + \alpha_2)\delta x_2.$$

Finally, after obtaining the closed form solution for u, we set again $\mathbf{x_E} = [\frac{\pi}{6}, 0]^T$ and $\mathbf{x_0} = [-0.4, 0]^T$. Figure 1 shows a comparison between trajectories produced by the LQR and the regular nonlinear controller. As can be seen, the nonlinear controller tends to copy the state variables of the reference linear controller which was also expected. Although the nonlinear controller performs



(a) The trajectory of the position produced by the regularized nonlinear controller still tends to the desired equilibrium point with a slight phase change.

(b) The trajectory of the velocity produced by the regularized nonlinear controller, however, has a smaller overshooting than the trajectory of the velocity produced by LQR.

(c) Just like the velocity trajectory, the torque produced by the regularized nonlinear controller tends to zero in a slightly more damped manner with a slight phase change.

(d) The trend of the first element of the gain matrices. The one from regularized nonlinear controller tends to the one from LQR towards the end of the simulation.

Figure 2: A comparison between the regularized nonlinear controller and LQR. The initial conditions are given for both controllers as $\mathbf{x_0} = [-0.4, 0]^T$ and $\mathbf{x_E} = [\frac{\pi}{6}, 0]^T$.

slightly better, i.e. reaches the steady state a few moments earlier, the difference is rather small to be called significant.
Analyzing it further, we found that the matrix $\mathbf{Z_{lin}}$ is not positive because its first element is equal to zero. This observation can be explained if we look at the matrices $\mathbf{A_{lin}}$ and $\mathbf{B}$ which are used to compute $\mathbf{Z_{lin}}$. It is obvious that if we apply (6) to compute $\mathbf{Z_{lin}}$, the result of matrix multiplications will always produce zero for the first element of the resulting matrix. However, having a non-positive definite matrix does not seem to affect the dynamics of the nonlinear controller significantly. Still, it is possible to have a positive definite $\mathbf{Z_{lin}}$ if we rewrite $\dot{x}_1$ using an integral of $\dot{x}_2$. This improvement is left for a future work.

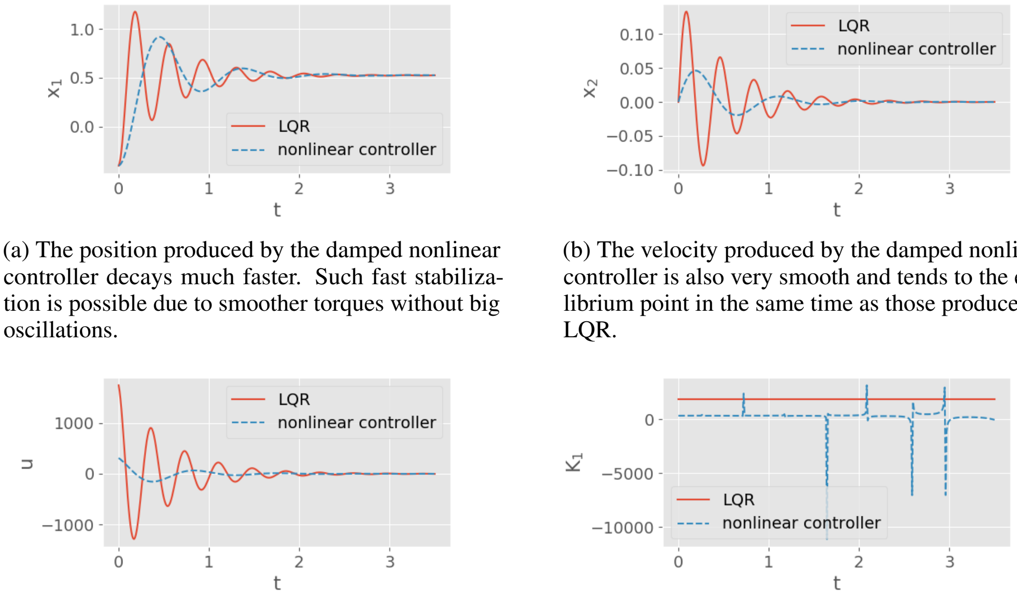
6.2.1 Regularized Nonlinear Controller

Figure 1c shows that the torque becomes too high in the beginning of the simulation. The reason of it is that the computation of the pseudo-inverse in (10) results in numerical instabilities. To solve this problem, we use a regularized pseudo-inverse $\mathbf{M} = (\mathbf{M^T M} + \lambda \mathbf{I})\mathbf{M^T}$, where λ is the regularization parameter. The main advantage of this approach is that by tuning λ, we can achieve different behaviors of the system: decreasing it results in lower torques but also in slower stabilization of the system and vice versa. Figure 2 shows an example trajectory with $\lambda = 10^{-7}$. As can be seen, the velocity of the system produced by the regularized nonlinear controller (Figure 2b) has smoother oscillations compared to LQR while still reaching the desired position at the same time (Figure 2a).

6.2.2 Damped Nonlinear Controller

Since we align the dynamics of the nonlinear controller with the dynamics of the reference controller, we suggest to use the gain matrix $\mathbf{K_{lin}}$ to damp the appropriate gain matrix $\mathbf{K}(\mathbf{\Delta x})$. Since the second element in both gain matrices are equal, it makes sense to damp only the first element:

$$\mathbf{K}(\mathbf{\Delta x})_{\mathbf{1}} - = \mathbf{K}_{\mathbf{lin,1}}.$$



(a) The position produced by the damped nonlinear controller decays much faster. Such fast stabilization is possible due to smoother torques without big oscillations.

(b) The velocity produced by the damped nonlinear controller is also very smooth and tends to the equilibrium point in the same time as those produced by LQR.

(c) Smooth torques produced by the damped nonlinear controller are the main reason of a nice behavior of the system. They are damped much calmer than those produced by LQR.

(d) On one side, the first element of the gain matrix $\mathbf{K}(\mathbf{x})$ does not tend to the gain matrix of LQR. On the other side, it produces smoother torques as well as position and velocity trajectories.

Figure 3: A comparison between the damped nonlinear controller and LQR. The initial conditions are given for both controllers as $\mathbf{x_0} = [-0.4, 0]^T$ and $\mathbf{x_E} = [\frac{\pi}{6}, 0]^T$.

Surprisingly, this method produces the best behavior of the system as can be seen in Figure 3. Both position and velocity get damped with much less oscillations while the torque is initialized with a much lower value and has almost no oscillation compared to the torque produced by LQR.

7 Conclusion

We introduced LQR for nonlinear systems and showed how to construct a nonlinear controller whose dynamics is aligned with the dynamics of LQR. A mountain car problem with no torque limits was extensively studied, including linearization, applying LQR and deriving the appropriate nonlinear controller. It turned out that the regular nonlinear controller did not provide a significant improvement compared to LQR. The main problem is in the computation of the inverse of $\mathbf{M}$. Since the equation in (9) is overdetermined, the computation rely on the Moore–Penrose inverse, possibly causing numerical instabilities. Using a regularized pseudo-inverse yields a better performance in one of the states leaving the other state and the torque the same as in regular nonlinear controller. The second improvement was made because the initial torque was always higher than the torque of LQR. We tried to alleviate this problem by additionally damping the gain matrix $\mathbf{K}(\mathbf{\Delta x})$ using $\mathbf{K_{lin}}$. Such method showed a significant improvement mainly due to much smoother torques.

The success of both improvements as well as a limited success of the regular nonlinear controller allow us to conclude our work by confirming superior performance of the resulting nonlinear controller. Thus, we suggest that the current literature approach to synthesis of controllers based on the Lyapunov theory can be potentially rather useful for generating appropriate reward functions and plan to investigate further this approach.

References

J. Ackermann. Entwurf durch polvorgabe. *at - Automatisierungstechnik*, 25(6):173–179, 1977.

Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002. ISSN 0005-1098. doi: 10.1016/S0005-1098(01)00174-1.

B. Itschner. Einführung idealer ljapunov-funktionen / introduction of ideal lyapunov-functions. *Regelungstechnik*, 25:251–257, 1977.

T. A. Johansen, I. Petersen, and O. Slupphaug. On explicit suboptimal lqr with state and input constraints. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, volume 1, pages 662–667 vol.1, 2000. doi: 10.1109/CDC.2000.912842.

H. Keller. *Entwurf nichtlinearer Beobachter mittels Normal formen.* PhD thesis, Universität Karlsruhe, 1986.

J.P. LaSalle and S. Lefschetz. *Die Stabilitätstheorie von Ljapunow: die direkte Methode mit Anwendungen*. BI-Hochschultaschenbücher. Bibliographisches Institut, 1967.

A.M. Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55 (3):531–534, 1992.

H. Neudecker. A note on kronecker matrix products and matrix equation systems. *SIAM Journal on Applied Mathematics*, 17(3):603–606, 1969. ISSN 00361399.

U. Sieber. Ljapunow-synthese nichtlinearer regelungen über die approximation einer linearen regelung. *Automatisierungstechnik*, 37:455–461, 1989.

U. Sieber. Nichtlinearer ausgangsreglerentwurf durch gütemaßangleichung. *Automatisierungstechnik*, 42(1-12):149–154, 1994.

Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

E. Todorov. Goal directed dynamics. In *IEEE International Conference on Robotics and Automation*, 2018.

Appendix

A Mathematical Operations

The following subsections provide necessary mathematical operations needed for the norm derivation.

A.1 Kronecker Product

The Kronecker product is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & & & \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix}. \quad (12)$$

Just like matrix multiplication, the Kronecker product is associative and distributive and not commutative.

A.2 Vectorization operator

According to Neudecker [1969], the vectorization operator vec($\mathbf{M}$) transforms the matrix $\mathbf{M}$ into a vector in the following way:

$$\text{vec}(\mathbf{M}) = \text{vec}([m_1 \dots m_n]) = \begin{bmatrix} m_1 \\ \cdot \\ \cdot \\ m_n \end{bmatrix},$$

where $m_1, \dots, m_n$ denote the matrix columns. Subsequently, it follows that vec is linear and it holds

$$\text{vec}(\mathbf{ABC}) = \mathbf{C^T} \otimes \mathbf{A}\text{vec}(\mathbf{B}). \quad (13)$$

A.3 Permutation Matrix

A permutation matrix is given by

$$\mathbf{U}_{n \times m} = \begin{bmatrix} E_{11} & E_{21} \dots E_{m1} \\ E_{12} & E_{22} \dots E_{m2} \\ \vdots & \\ E_{1n} & E_{2n} \dots E_{nm} \end{bmatrix},$$

where $\mathbf{E_{ik}}$ is a (m, n)-matrix with 1 at the position (ik) and zeros at all other positions. For the mountain car problem, the permutation matrices are given by

$$\mathbf{U_{n \times n}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{U_{p \times n}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

It holds that

$$\mathbf{U}_{m \times n} \cdot \mathbf{U}_{n \times m} = \mathbf{I_{mn}},$$

where $\mathbf{I_{mn}}$ is an identity matrix with the size mn.

B Norm minimization

To minimize (8) according to Sieber [1989], we consider the following equation

$$\mathbf{Z_{lin}} - \mathbf{Z}(\mathbf{x}) = \mathbf{Z_{lin}} + \mathbf{PA}(\mathbf{x}) + \mathbf{A}(\mathbf{x})^\mathbf{T}\mathbf{P} - \mathbf{PBR} - \mathbf{R^T B^T P}.$$

After applying (A.2), we get

$$\begin{aligned}\text{vec}(\mathbf{Z_{lin}} - \mathbf{Z}(\mathbf{x})) = \text{vec}(\mathbf{Z_{lin}}) + \text{vec}(\mathbf{PA}(\mathbf{x})\mathbf{I}) + \text{vec}(\mathbf{IA}(\mathbf{x})^{\mathbf{T}}\mathbf{P}) - \\ - \text{vec}(\mathbf{PBRI}) - \text{vec}(\mathbf{IR}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{P}).\end{aligned}$$

The identity matrix was introduced in order to split the parts of summands. For that, we apply (13) and obtain

$$\begin{aligned}\text{vec}(\mathbf{Z_{lin}} - \mathbf{Z}(\mathbf{x})) = \text{vec}(\mathbf{Z_{lin}})\mathbf{I} \otimes \mathbf{P}\text{vec}(\mathbf{A}(\mathbf{x}) + \mathbf{P} \otimes \mathbf{I}U_{n\times n}\text{vec}(\mathbf{A}(\mathbf{x})) - \\ -(\mathbf{I} \otimes (\mathbf{PB}))\text{vec}(\mathbf{R}) - ((\mathbf{PB} \otimes \mathbf{I})\mathbf{U_{p\times n}}\text{vec}(\mathbf{R}))\end{aligned}$$

By grouping the summands of the last equation, we obtain the matrices $\mathbf{M}$ and $\mathbf{J}$ as defined in (9).

C Linearization of the Mountain Car Problem

Since $\mathbf{A}(\mathbf{\Delta x})$ is obviously nonlinear because of the function

$$f(x) = \frac{-0.0025 \cdot \cos(3 \cdot x)}{x}$$

we first need to linearize it at $\mathbf{\Delta x} = 0$ in order to construct a linear controller.
We use the first order Taylor expansion to approximate the nonlinear term:

$$\begin{aligned}f'(x) &\approx f(x_{1E}) + f'(x_{1E})(x - x_{1E}) \\ &= \frac{-0.0025 \cdot \cos(0)}{0} + f'(0)x.\end{aligned}$$

This function is obviously indeterminate because of the term $\frac{-0.0025 \cdot \cos(0)}{0}$. By applying l'Hopital's rule, we obtain for $f(x_E)$

$$\begin{aligned}f(0) &= \lim_{x\to 0} \frac{(-0.0025 \cdot \sin(3x))'}{x'} \\ &= \lim_{x\to 0} \frac{(-3 \cdot 0.0025 \cdot \cos(3x))}{1} \\ &= -3 \cdot 0.0025 \cdot \cos(3 \cdot 0) \\ &= -0.075.\end{aligned}$$

For function's derivative, we apply l'Hopital's rule again and obtain $f'(0) = 0$. Substituting the resulting functions in the first order Taylor expansion, we obtain a numerical solution to the system matrix:

$$\mathbf{A_{lin}} = \begin{bmatrix} 0 & 1 \\ -0.075 & 0 \end{bmatrix}.$$

This form of the system matrix allows us to apply LQR in order to find $\mathbf{K_{lin}}$ as described in Section 3.2.