Technical Notes

Successive Linearization of Nonlinear Systems

Institute of Flight Mechanics and Controls

Elias Niepötter

September 24, 2024

1 Setup

Let $f(x, u)$ be a non-autonomus nonlinear system which maps $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^g$. The state space and input space are denoted by $\chi \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ respectively. The domain of f is defined as $\Omega := \chi \times \mathcal{U}$. The system f is assumed to be Lipschitz continuous in the state space and continuous in time, guaranteeing the existence and uniqueness of the solution.

In the following notes, the forced Van der Pol oscillator will be used as an example of the described dynamics

$$\mathbf{f}(x, u) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \mu(1 - x_1^2)x_2 - x_1 + u \end{bmatrix}. \tag{1}$$

The system (1) has a limit cycle close to the origin. The parameter μ defines the stability of the limit cycle and the equilibrium at the origin. For $\mu < 0$ the limit cycle is unstable and the origin is asymptotically stable inside the limit cycle. For $\mu > 0$ the limit cycle becomes attractive while the equilibrium becomes unstable. Figure 1 shows exemplary the vector field of the Van der Pol oscillator for $\mu = 0.5$ and two trajectories.

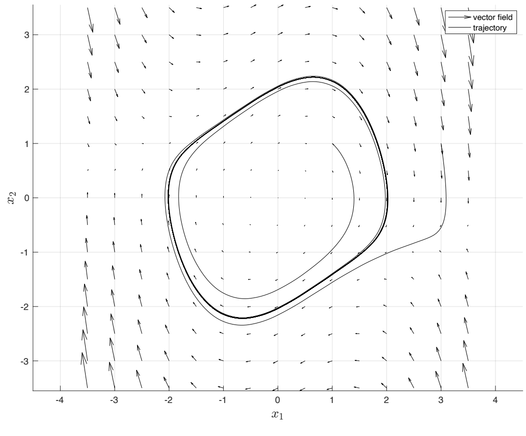

Figure 1: Van der Pol Oscillaotr for $\mu = 0.5$

2 Linearization error dynamics

Let $\overline{f}(x,u)$ denote the linearized dynamics of $f(x,u)$. The linearization is based on a Taylor series expansion

$$\begin{aligned} \overline{f}(x,u) &= \left.\frac{\partial f}{\partial x}\right|_{\overline{x},\overline{u}} \cdot x + \left.\frac{\partial f}{\partial u}\right|_{\overline{x},\overline{u}} \cdot u + f(\overline{x},\overline{u}) && (2) \\ &= Ax + Bu + Z. && (3) \end{aligned}$$

The linearization error $e(x,u)$ is defined as

$$e(x,u) = f(x,u) - \overline{f}(x,u). \tag{4}$$

The gradient of the linearization error w.r.t the state is denoted by $\nabla_x e$ and the gradient of the linearization error w.r.t the input is denoted by $\nabla_u e$. The magnitude of the linearization error is computed by the 2-norm

$$||e(x,u)||_2 = ||f(x,u) - \overline{f}(x,u)||_2. \tag{5}$$

By assumption, the system $f(x,u)$ is Lipschitz continuous in the state space, therefore the Lipschitz constant $\mathcal{L}_S$ w.r.t the set $S = \Omega$ exists. The Lipschitz constant can be computed by

$$\mathcal{L}_S = \max_{(x,u)\in S} ||\nabla_x f(x,u)||_2. \tag{6}$$

The operator ∇_x indicates the gradient w.r.t x, so the Jacobian of f. The norm in (6) represents the spectral norm of a matrix which is defined as the largest singular value of the matrix to which it is applied. The operator norm can be computed via the following result: [1]

$$||\nabla_x f(x,u)||_2 = \sqrt{\lambda_{\max}\left[\nabla_x f(x,u) \nabla_x^T f(x,u)\right]} \tag{7}$$

Equations (6) and (7) are also applied to $e(x,u)$. [1]

3 Control Approach

The nonlinear control approach presented here is motivated by

1. a lightweight computational footprint,
2. optimality to some extend
3. and the lack of a precomputed trajectory.

The algorithms presented later are based on a setpoint control scheme. At each instance in time, a setpoint dictates where the state should be driven to. This setpoint is updated every time it is reached. This method relates to the lack of a precomputed trajectory, which would normally define all setpoints a priori. To tackle the optimality and the lightweight computational footprint, Linear Quadratic Regulators (LQR) are computed successively to stabilize the system around the active setpoint. Obviously, the optimality criteria can only be guaranteed to some extent in a local set around the active setpoint. The computational demand is mainly driven by the solver performance for the Ricatti equation and the setpoint computation. However, once a local controller for the active setpoint is computed, the computational demand to determine the control input becomes neglectable. That is the advantage of the linear controller.

[1] not sure if the Lipschitz property is actually inherited by $e(x,u)$

The main challenge is to compute the setpoints. When a new setpoint is selected, one has to make sure that the linearization holds in order to make the LQR work. By intuition, if the new setpoint is too far away from the current state, the linearization is not valid anymore, thus the LQR will fail to stabilize the system around the setpoint. The idea here now is to include knowledge about the linearization error in the control law respectively the setpoint computation. The next sections present some results regarding the described control approach.

4 Numerical Experiments

The following briefly explains the implemented algorithms. Algorithm 1 gives an overview of the computation of the control input u. Not shown in algorithm 1 is the actual update of the dynamics. To solve the system (1) a simple Runge-Kutta-4 integration scheme is used.

Algorithm 1 Successive LQR

```
Require: $x, s, f$                    ▷ current state $x$, setpoint $s$ and dynamics $f$
Ensure: $u$                           ▷ control input $u$
 1: $s \leftarrow \textbf{setpointGeneration}(x, s, f)$     ▷ update setpoint
    linerization scheme:
 2: $\overline{x} \leftarrow s$
 3: $\overline{u} \leftarrow 0$
 4: $\mathbf{A} \leftarrow \frac{\partial f}{\partial x}|_{\overline{x},\overline{u}}$
 5: $\mathbf{B} \leftarrow \frac{\partial f}{\partial u}|_{\overline{x},\overline{u}}$
 6: $\mathbf{Z} \leftarrow f(\overline{x}, \overline{u})$
    controller synthesis:
 7: assign scaling matrices $\mathbf{Q}, \mathbf{R}$
 8: $\mathbf{K} \leftarrow \mathbf{LQR}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R})$     ▷ neglecting affine term $\mathbf{Z}$
    compute control input:
 9: $x^* \leftarrow x - s$            ▷ change of coordinates to track the setpoint
10: $u \leftarrow -\mathbf{K}x^*$
```

Note that because the algorithm tracks a defined setpoint, the tracking error dynamics are solved, instead of (1). This is implemented by a change of coordinates indicated at line 9. The main simplification performed to compute the control input is the neglection of the affine term $\mathbf{Z}$. The setpoint generation method is the key part of algorithm 1. Different approaches are presented in the next sections.

4.1 Setpoint Generation - Sampling Approach

The first and simplest approach is to search for a new setpoint by sampling random candidates around the current setpoint and evaluating the linearization error. An overview of this method is given in algorithm 2. The algorithm takes the current state, the dynamics and two user-defined parameters ϵ_1 and ϵ_2 as input arguments and outputs a new setpoint.

After generating a set of samples $\mathcal{S}$, the search for a new setpoint is the core of algorithm 2. First the function $e(x, u)$ describing the linearization error dynamics is constructed. Afterwards, the linearization error is evaluated via the 2-norm. At this point, it is very important to note what is evaluated in line 9. The samples represent candidates for new setpoints. Later, an LQR is designed to regulate the system towards this setpoint. So in line 8, the dynamics f are linearized around a sample resp. a possible setpoint. The equation $e(x, u)$ describes the linearization error when moving away from this possible setpoint. That is why $e(x, u)$ is constructed uniquely for every sample. [2] The linearization error is then evaluated at the same point in the state space every time, namely at the current state x. When all setpoint candidates are evaluated, all setpoints with a larger error than the user-defined threshold Δ are neglected. Only the valid setpoints remain in the set $\mathcal{S}_v$.

Algorithm 2 setpointGeneration - sampling approach

```
Require: $x, f, \epsilon_1, \epsilon_2$
Ensure: $s^*$                                                    ▷ new setpoint $s^*$
  sample generation
  $\mu \leftarrow x$
  $\sigma \leftarrow \epsilon_1$                                  ▷ parameter $\epsilon_1$ selected by user
  $\mathcal{S} \leftarrow \text{normrnd}(\mu, \epsilon_1, n)$      ▷ size of sample set $\mathcal{S}$ is $n$
  search for valid setpoints
  $\Delta \leftarrow \epsilon_2$                                  ▷ parameter $\epsilon_2$ selected by user
  for $i \leq n$ do
      $\overline{x} \leftarrow \mathcal{S}(i)$
      $\overline{u} \leftarrow 0$
      $e(x,u) \leftarrow f(x,u) - \frac{\partial f}{\partial x}|_{\overline{x},\overline{u}} \cdot x - \frac{\partial f}{\partial u}|_{\overline{x},\overline{u}} \cdot u - f(\overline{x},\overline{u})$   ▷ error dynamics
      $\mathcal{E}(i) \leftarrow ||e(x - \overline{x}, 0)||_2$    ▷ evaluating the linearization error at $x$
  end for
  $\mathcal{S}_v \leftarrow \{\mathcal{S}|\mathcal{E}_i \leq \Delta \quad \text{for } i = [1, n]\}$   ▷ defining the valid set of setpoints
  search for best setpoint
  $m \leftarrow \text{length}(\mathcal{S}_v)$
  $d \leftarrow 10^6$
  for $i \leq m$ do
      $d^* \leftarrow ||\mathcal{S}_v(i)||_2$                     ▷ searching for setpoint closest to the origin
      if $d^* < d$ then
          $d \leftarrow d^*$
          $s^* \leftarrow \mathcal{S}_v(i)$
      end if
  end for
```

Lastly, it is searched for the best setpoint candidate simply by evaluating the distance from each candidate to the origin. The closest valid setpoint candidate becomes the new setpoint. The figures 2 and 3 show exemplary results for the combined algorithms 1 and 2. The number of samples at each setpoint search is $n = 500$. The variance for the sampling is selected to $\sigma = 0.5$ for both states. The threshold to distinct between valid and non-valid candidate setpoints is $\Delta = \overline{\mathcal{E}} - \text{var}(\mathcal{E})$, so all setpoints below the 1σ interval of the distribution of the linearization error. The LQR weighting matrices are selected to $\mathbf{Q} = \mathbb{I} \cdot 100$ and $\mathbf{R} = 1$.

[2] the assumption here is that the linearization error evaluated at a point x_1 when linearizing around x_2 is not equal to the error when evaluating at x_2 and linearizing around x_1

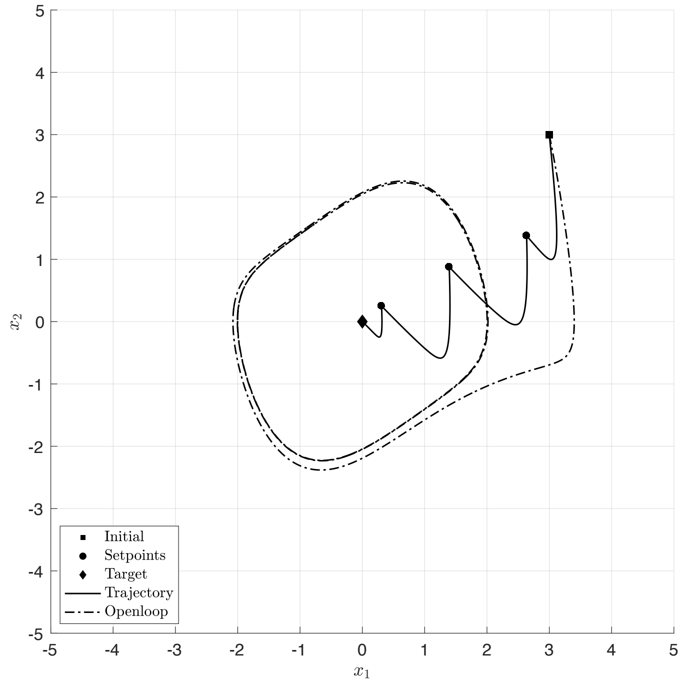

Figure 2: State space trajectory of the Van der Pol Oscillator for $\mu = 0.5$ when controlled with alg. 1 and 2. Additionally, the open-loop trajectory without any control input is included to illustrate the limit cycle.

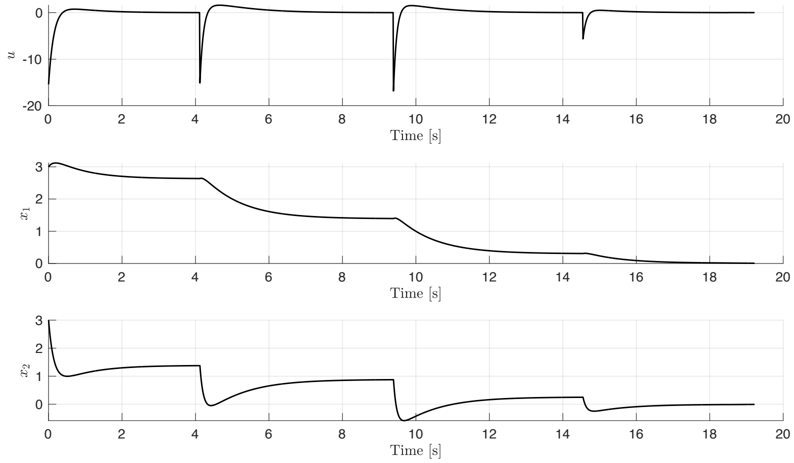

Figure 3: State and input signals of the Van der Pol Oscillator for $\mu = 0.5$ when controlled with alg. 1 and 2.

4.2 Setpoint Generation - Optimization Approach

In contrast to the sampling approach, the approach described in this section tries to utilize the knowledge about the linearization error dynamics, as it is assumed that the model dynamics are known. Further, a nonlinear optimization problem or nonlinear program (NLP) is formulated in such a way that the optimal solution resembles the optimal new setpoint.

Recalling the definition of the linearization error dynamics from (4):

$$\begin{aligned} e(x,u) &= f(x,u) - \overline{f}(x,u) \\ &= f(x,u) - \frac{\partial f}{\partial x}|_{\overline{x},\overline{u}} \cdot x - \frac{\partial f}{\partial u}|_{\overline{x},\overline{u}} \cdot u - f(\overline{x},\overline{u}) \end{aligned}$$

In most cases, the variables x and u represent the independent variables of any dynamics. The parameters $\overline{x}$ and $\overline{u}$ are in general fixed parameters for the linearization. However, in this case, the linearization point is a priori unknown. The parameters $\overline{x}$ and $\overline{u}$ are now to be seen as independent variables, while the variables x and u resemble the fixed evaluation point of the linearization error. Therefore, the linearization error dynamics are redefined to

$$\begin{aligned} e(\overline{x},\overline{u}) &= f(\hat{x},\hat{u}) - \overline{f}(\overline{x},\overline{u}) && (8) \\ &= f(\hat{x},\hat{u}) - \frac{\partial f}{\partial x}|_{\overline{x},\overline{u}} \cdot \hat{x} - \frac{\partial f}{\partial u}|_{\overline{x},\overline{u}} \cdot \hat{u} - f(\overline{x},\overline{u}). && (9) \end{aligned}$$

To better distinct the variables, the parameters $\hat{x}$ and $\hat{u}$ denote the evaluation point of the linearization error. With the reformulation of the linearization error dynamics, the NLP can be formulated to

$$\begin{aligned} \min_{\overline{x},\overline{u}} \quad & ||e(\overline{x},\overline{u})||_2 \quad \text{at } \hat{x},\hat{u} \\ \text{s.t.} \quad & \overline{x} \in \mathcal{X}, \quad \overline{u} \in \mathcal{U} \end{aligned} \qquad (10)$$

The algorithm to determine the new setpoint with this optimization approach is summarized in alg. 3.

Algorithm 3 setpointGeneration - optimization approach

```
Require: $\hat{x}, \hat{u}, f, \mathcal{X}, \mathcal{U}$
Ensure: $s^*$
1: $e(\overline{x},\overline{u}) \leftarrow f(\hat{x},\hat{u}) - \frac{\partial f}{\partial x}|_{\overline{x},\overline{u}} \cdot \hat{x} - \frac{\partial f}{\partial u}|_{\overline{x},\overline{u}} \cdot \hat{u} - f(\overline{x},\overline{u})$.
2: $J \leftarrow ||e(\overline{x},\overline{u})||_2$
3: NLP $\leftarrow \min_{\overline{x},\overline{u}} J$ s.t. $\overline{x} \in \mathcal{X}, \overline{u} \in \mathcal{U}$
4: $s^* \leftarrow$ sol(NLP)
```

The figures 4 and 5 show exemplary results for the Van der Pol Oscillator controlled with alg. 3. The optimization algorithm is MATLAB's implementation of an interior-point method interfaced through `fmincon`. The input at the linearization point $\overline{u}$ is set to zero to simplify the problem. Only the linearization state is unknown. The evaluation control input $\hat{u}$ is as well always set to zero. The sets where the optimizer looks for the minimum are set to $\mathcal{X} = \{\overline{x} | \hat{x} - 1.5 < \overline{x} < \hat{x} + 1.5\}$ for the states and $\mathcal{U} = \phi$ for the input denoting the empty set.

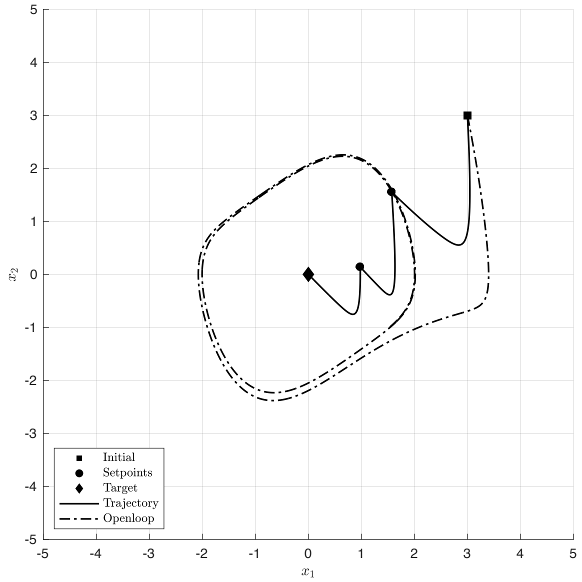

Figure 4: State space trajectory of the Van der Pol Oscillator for $\mu = 0.5$ when controlled with alg. 1 and 3.

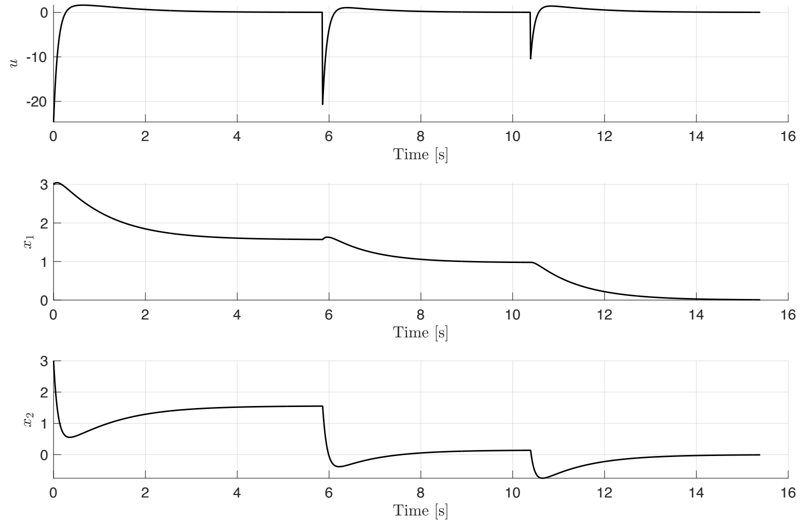

Figure 5: State and input signals of the Van der Pol Oscillator for $\mu = 0.5$ when controlled with alg. 1 and 3.

5 Outlook

The following topics represent possible next steps and further considerations for the algorithms presented:

- properly including the affine dynamics into the synthesis of the LQR (alg. 1)
- parameter studies for alg. 2 and 3 on the
  - threshold Δ
  - variances σ and potentially cross-correlations
  - parameters of the optimizer
  - state constraints/set $\mathcal{X}$
- including nonlinear constraints (or modification of the cost function) for alg. 3 to ensure a descent direction towards the origin
- runtime analysis especially on alg. 3
- comparisons to established nonlinear control techniques regarding typical control performance metrics
- analyzing alg. 3 in terms of convergence properties

References

[1] Sebastian A. Nugroho, Ahmad F. Taha, and and Vu Hoang. Nonlinear Dynamic Systems Parameterization Using Interval-Based Global Optimization: Computing Lipschitz Constants and Beyond, May 2022. arXiv:2004.12061 [cs, eess, math].