# Strict Modes Everywhere – Bringing Order into Dynamics of Mechanical Systems by a Potential Compatible with the Geodesic Flow

Arne Sachtler[1,2] and Alin Albu-Schäffer[2,1]

***Abstract*—Strict nonlinear normal modes provide very regular families of oscillations within conservative mechanical systems. However, a strict normal mode will generally be an isolated curve within the configuration space of the system. In this paper, we design a potential that will densely fill the configuration space with strict normal modes such that each configuration belongs to one mode and each mode passes through a common point, the equilibrium. As the potential can be realized by (nonlinear) elastic elements it can be used to execute a variety of periodic trajectories very efficiently. Most of the required torques will come from the elastic elements in the system and not from the actuators. We also design a controller stabilizing the system to a desired target mode and a controller performing swing-up and compensating dissipated energy. Finally, we showcase the approach for a two DoF manipulator. The experiments show that the approach performed well for the example system.**

***Index Terms*—Compliant Joints and Mechanisms; Modeling, Control, and Learning for Soft Robots; Mechanism Design; Energy and Environment-Aware Automation; Dynamics**

## I. Introduction

SEVERAL applications of robotics can be seen as periodic or close-to-periodic motions. Consider a pick-and-place application where a robot picks objects from a conveyor belt and assorts them into a box. This can be seen as a sequence of close-to-periodic motions: a base oscillation between the conveyor belt and the center of the box is slightly modulated in order to reach the specific positions in the box. Beside the industrial case, also legged and non-legged locomotion involves robotic structures oscillating periodically.

Classical industrial robots and also many recent humanoid (e.g. [2]) or zoomorphic robots use stiff actuation without any (wanted) elastic elements. The actuation and controllers generate all the required forces to steer the system onto the desired trajectories. In that case, a lot of energy exchange occurs between the actuation (usually the electrical part) and the mechanical part. This comes with subobtimal energy efficiency, as energy exchange between these two parts is subject to losses. Especially, mechanical energy is usually not recovered, but dumped to heat in the motor drivers. Therefore, the question arises if elastic elements in the system may help to increase energy efficiency. Elastic elements store and release mechanical energy and create a variety of natural oscillations in the mechanical system. Optimally, these natural oscillations coincide with the desired trajectories. Thus, the actuation will only need to stabilize them and it is expected that the effort needed for corrective actions on natural oscillations is much less than for classical trajectory tracking on stiff robots. If the natural oscillations do not coincide with the desired trajectories, they might have the correct tendency and will only need to be modulated by the actuators.

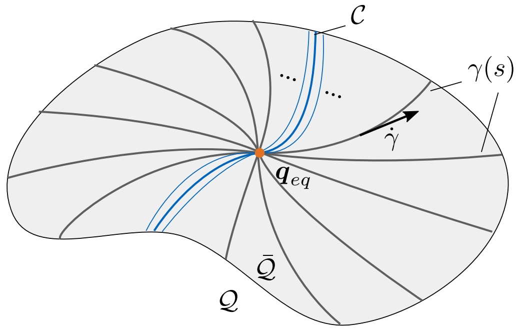


Fig. 1. Aim of this paper: find a potential function such that the configuration is densely packed with strict modes.

The combination of robotic structures and elastic elements leads to very complex systems showing a wide variety of nonlinear (including chaotic [3]) behaviors. Recently, the notion of nonlinear normal modes was introduced to the robotics community [1], [4]: Despite the dynamics being very complex, very regular and simple, i.e., low-dimensional intrinsic motions can be found. These can be collected on continuous families of periodic solutions. A special case of such periodic solutions are *strict nonlinear normal modes* [4], which can be seen as invariant curves containing the equilibrium. These curves lie within the configuration space of the elastic robotic system. In particular, when starting at a configuration on the strict normal mode, it will oscillate back and forth periodically on a breaking trajectory and the configuration path will trace out a subpath of the strict mode.

Strict normal modes provide very structured oscillations. However, they usually occur, if at all, as isolated curves within the configuration space. The trajectories will be very complex if not initialized directly on a strict normal mode.

Manuscript received: September 7, 2021; revised December 6, 2021; accepted December 23, 2021.

This paper was recommended for publication by Associate Editor T. Asfour and Editor C. Gosselin upon evaluation of the reviewers' comments. This work is supported by the ERC advanced grant *M-Runners* (ID: 835284) by the European Research Council (ERC).

[1] A. Sachtler and A. Albu-Schäffer are with the Technical University of Munich (TUM), Department of Informatics; Boltzmannstr. 3, 85748 Garching;

[2] They are also with the German Aerospace Center (DLR), Institute of Robotics and Mechatronics; Münchener Str. 20, 82234 Weßling, Germany

Several state of the art publications already deal with the exploitation of oscillations in elastic robotic systems. Especially, in the field of legged locomotion research the question arises if the oscillations can be exploited [5]–[8]. These publications usually deal with periodic orbits in general, not with the special case of normal modes. Moreover, a couple of publications deal with the question of how to control a robot onto a nonlinear normal mode [9], [10]. All these publications have one thing in common: the mechanics and the elastic elements are considered given and the main question is how to exploit existing nonlinear normal modes or periodic orbits.

In the present paper, we address another question: given the structure of a mechanical system, especially its multi-body dynamics; can we find elastic elements such that strict modes are everywhere, i.e., such that each point in configuration space belongs to a strict mode? Such a system would always oscillate periodically when initializing it with zero velocity (or another *compatible* velocity) from anywhere. Therefore, the elastic elements would help, in a sense, ordering the complex mechanical system with otherwise complicated solution behavior.

## II. Strict Nonlinear Normal Modes

Consider a conservative mechanical system and let $\mathcal{Q}$ denote the corresponding configuration space manifold. In coordinates it is characterized by the ordinary differential equation (ODE)

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \frac{\partial U}{\partial \boldsymbol{q}} = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ denotes the positive definite and symmetric mass tensor, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ denotes centrifugal and Coriolis forces and $U(\boldsymbol{q})$ denotes a scalar potential function.[1] As in [1] we use some abuse of notation and refer to $\boldsymbol{q}$ as both the point on the manifold $\boldsymbol{q} \in \mathcal{Q}$ as well as its coordinates $\boldsymbol{q} \in \mathbb{R}^{\dim \mathcal{Q}}$. In (1) $\boldsymbol{q}$ denotes the configuration variable coordinates and $\dot{\boldsymbol{q}} \in \mathcal{T}_{\boldsymbol{q}}\mathcal{Q}$ the joint velocities.

The ODE (1) can also be written coordinate-independently. Considering that, we endow the manifold $\mathcal{Q}$ with a Riemannian metric $g$ generated by the coordinate-independent version of the inertia tensor $\boldsymbol{M}(\boldsymbol{q})$. Then, let $\boldsymbol{\nabla}_{\cdot}\cdot$ denote the Levi-Civita connection [11] on the Riemannian manifold $(\mathcal{Q}, g)$ and let $\mathcal{U}(\boldsymbol{q})$ be the coordinate-independent version of the potential. We can then write the system dynamics as

$$\boldsymbol{\nabla}_{\dot{\boldsymbol{q}}}\dot{\boldsymbol{q}} = -\boldsymbol{\nabla}\mathcal{U}, \tag{2}$$

where $\boldsymbol{\nabla}$ (no subscript) denotes the gradient[2]. We restate here the definition of strict normal modes and a theorem from [1]:

**Definition** (Strict Normal Mode)**.** *Let $\mathcal{C} \subset \mathcal{Q}$ be a one-dimensional submanifold of $\mathcal{Q}$ with line segment topology. $\mathcal{C}$ can be parametrized by the parametric curve $\gamma : [0, 1] \to \mathcal{Q}$. If the tangent bundle $\mathcal{TC} \subset \mathcal{TQ}$ is an invariant set of the differential equation, it will be called a* strict normal mode.

**Theorem 1.** *$\mathcal{C}$ is a strict normal mode of (2) iff*

a) *$\mathcal{C}$ is a geodesic of the Riemannian manifold $(\mathcal{Q}, g)$ and*
b) *$\boldsymbol{\nabla}\mathcal{U}$ is tangential to $\mathcal{C}$ everywhere on $\mathcal{C}$.*

## III. Approach

Strict modes are generally isolated curves in the configuration space of a conservative mechanical system. In this work, we aim at finding a (nonlinear) elastic potential such that strict modes lie densely in the configuration space $\mathcal{Q}$. In other words, we require that each $\boldsymbol{q} \in \mathcal{Q}$ is touched by at least one strict mode. As will turn out this generally cannot be achieved on arbitrary large regions of $\boldsymbol{q} \in \mathcal{Q}$, so we may need to select a region $\bar{\mathcal{Q}} \subset \mathcal{Q}$.

The goal is summarized as follows (compare also Fig. 1):

**Problem Statement.** *Find a potential function $\mathcal{U}_{\mathrm{e}}$ such that a predefined region of the configuration space $\bar{\mathcal{Q}}$ containing the equilibrium $\boldsymbol{q}_{eq} \in \bar{\mathcal{Q}}$ is densely packed with strict normal modes, i.e., every configuration $\boldsymbol{q} \in \bar{\mathcal{Q}}$ is part of at least one strict mode $\mathcal{C} \subset \bar{\mathcal{Q}}$.*

From Thm. 1a) it is obvious that each strict mode must be a geodesic of $(\mathcal{Q}, g)$. Therefore, each of the lines in Fig. 1 can be computed by starting at the equilibrium, choosing an initial velocity and integrating the geodesic flow. The phrase *integrating the geodesic flow* is another term for simulating the system dynamics without potential.

When keeping the above mentioned statements in mind, we can interpret the results in [1] like this: integrating the geodesic flow starting from the equilibrium will create *candidate strict normal modes*. However, in the classical setting, where $\mathcal{U}$ is fixed, only a few or even no geodesics will also satisfy Thm. 1b), i.e., the choice of the potential field $\mathcal{U}$ selects the actual strict modes out of the candidates. In this paper, we desire a potential compatible with all of the candidates.

Take the system dynamics without the potential and let us write it in coordinates

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{0}. \tag{3}$$

The approach in this paper allows to freely select the equilibrium configuration of the target system. Let us call the desired target equilibrium $\boldsymbol{q}_{\mathrm{eq}}$. Using standard integration schemes like the Runge-Kutta method the potential-free dynamics (3) can be integrated. We set the system to the target equilibrium configuration $\boldsymbol{q}_{\mathrm{eq}}$ and shoot the system into arbitrary directions by setting the initial velocity $\dot{\boldsymbol{q}}_0$ to arbitrary vectors. Then, we integrate (3) in order to obtain trajectories $\boldsymbol{q}(t)$. Let us parameterize $\boldsymbol{q}(t)$ with respect to arc length and call it $\gamma : [0, l] \to \mathcal{Q}$ where $l$ is the arc length of the curve that $\boldsymbol{q}(t)$ traces out, i.e., we define a new time $s$ such that the system travels with unit-speed. For each initial velocity $\dot{\boldsymbol{q}}_0$ a parametrized curve $\boldsymbol{\gamma}(s)$ is obtained (Fig. 1). It is important to note that the curves $\boldsymbol{\gamma}(s)$ do not depend on the magnitude, but only the direction of the initial velocity $\dot{\boldsymbol{q}}_0$. This is a result of the definition of geodesics and the fact that the trajectories of the potential-free multi-body system are geodesics with respect to the inertia metric tensor [12], [13]. Each of the geodesics $\boldsymbol{\gamma}(s)$ should correspond to a strict nonlinear normal mode of the system including the potential $U$.

Let us split the potential $U$ into a fixed and known potential $U_g$ (usually gravity) and an unknown potential $U_e$ to be determined

$$U(\boldsymbol{q}) = U_g(\boldsymbol{q}) + U_e(\boldsymbol{q}). \tag{4}$$

[1] When clear from the context we will refer to $\frac{\partial U}{\partial \boldsymbol{q}}$ as a column vector.

[2] On Riemannian manifolds the gradient is a vector $\boldsymbol{\nabla}\mathcal{U}$ obtained by applying the inverse metric to the covector of partial derivatives: $\boldsymbol{M}^{-1}(\boldsymbol{q})\frac{\partial U}{\partial \boldsymbol{q}}$.

We now need to ensure that for every $\boldsymbol{\gamma}(s)$ computed according to the above scheme, the acceleration due to the generalized force $\boldsymbol{\tau} = \frac{\partial U}{\partial \boldsymbol{q}}$ is tangential to $\boldsymbol{\gamma}(s)$ for every point on $\boldsymbol{\gamma}$. Formally, the potential $U_e(\boldsymbol{q})$ must satisfy

$$\boldsymbol{M}^{-1}(\boldsymbol{\gamma})\frac{\partial(U_g + U_e)}{\partial \boldsymbol{q}} = \alpha(\boldsymbol{\gamma})\dot{\boldsymbol{\gamma}}, \tag{5}$$

where $\alpha(\boldsymbol{\gamma}(s))$ denotes an unknown scalar function introduced to write tangentiality as equality. Let $[\dot{\boldsymbol{\gamma}}]^{\perp}$ denote the orthogonal complement of the vector $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^n$ written in matrix form, i.e., a matrix of rank $n-1$ whose columns are orthogonal to $\dot{\boldsymbol{\gamma}}$ such that $\dot{\boldsymbol{\gamma}}^{\mathsf{T}}[\dot{\boldsymbol{\gamma}}]^{\perp} = \boldsymbol{0}$. Then, the requirement (5) can be reformulated to

$$\frac{\partial(U_g + U_e)}{\partial \boldsymbol{q}}^{\mathsf{T}} \boldsymbol{M}^{-1}(\boldsymbol{\gamma})[\dot{\boldsymbol{\gamma}}]^{\perp} = \boldsymbol{0}. \tag{6}$$

This eliminates the unknown scalar function $\alpha$ and allows to formulate the problem as an optimization problem where (6) can be used for a cost function.

The conditions (5) and also (6) are inherently local and do not provide any information on the overall shape and the global behavior of the final potential function. In order to achieve a final function $U(\boldsymbol{q}) = U_g(\boldsymbol{q}) + U_e(\boldsymbol{q})$ that does behave like a potential we specify further constraints on the potential:

$$\min(U_g + U_e)(\boldsymbol{q}) = U(\boldsymbol{q}_{\text{eq}}) = U_0 \tag{7a}$$

$$\forall \boldsymbol{q} \neq \boldsymbol{q}_{\text{eq}} : \mathcal{L}_{\dot{\boldsymbol{\gamma}}}(U_g + U_e)(\boldsymbol{q}) > 0. \tag{7b}$$

The first constraint (7a) requires that the potential has its minimum at the equilibrium $\boldsymbol{q}_{\text{eq}}$. The operator $\mathcal{L}_X f$ in (7b) denotes the Lie-derivative of $f$ along the vector field $X$. By enforcing positivity of the Lie-derivative along the geodesics we ensure that the potential strictly monotonically increases when following the geodesics starting at the equilibrium.

Various methods to find a solution to (6) and the constraints (7a) - (7b) are conceivable. These also include finite element methods [14] and optimization algorithms. For the latter the constraint (6) can be relaxed and turned into an optimization objective. Let us write the target potential $U_e(\boldsymbol{q})$ as parametric function $U_e(\boldsymbol{\omega}, \boldsymbol{q})$, where $\boldsymbol{\omega}$ is a collection of all the parameters of the function. Then training and test data can be generated by computing geodesics according to the proposed scheme. Assume that a total of $N$ geodesics have been computed and let the $i$-th geodesic $\boldsymbol{\gamma}_i$ be expressed by $T_i$ samples. The $t$-th sample of the $i$-th geodesic consist of a position $\boldsymbol{\gamma}_{it}$ and a tangent $\dot{\boldsymbol{\gamma}}_{it}$. We then write the cost function

$$L(\boldsymbol{\omega}) = \sum_{i=1}^{N}\sum_{t=1}^{T_i} \left\| \frac{\partial(U_g + U_e)}{\partial \boldsymbol{q}}^{\mathsf{T}} \boldsymbol{M}^{-1}(\boldsymbol{\gamma}_{it})[\dot{\boldsymbol{\gamma}}_{it}]^{\perp} \right\|^2 . \tag{8}$$

In combination with (7ab) the cost function (8) yields a constrained optimization problem.

Also for the choice of the parametric model $U_e(\boldsymbol{\omega}, \boldsymbol{q})$ a large palette of possibilities is available [15], [16].

One choice would be using linear regression methods with basis functions. Each of the basis functions can be chosen such that they are realizable by an elastic element. Assume a selection of nonlinear elastic elements are available - [17] and [18] review a broad set of methods to design such elements.

For instance, given linear and cubic springs a lot of functions can already be implemented. Besides the obvious direct connection of a springs to a joint, the springs can, among other possibilities, be attached via crankshaft mechanisms implementing a sine-like force profile. Additionally, when using one-side mounted compression springs non-linear ReLU-like behaviour can be implemented, which can also be combined with other mechanisms.

Coupling terms could be introduced by constructing mechanical components that compute sums or differences of adjacent joints, e.g., a shaft and two pairs of bevel gears combined with an elastic element provide coupled opposing torques in two adjacent joints. These mechanical implementations create a selection of possible potential functions that can be used as basis functions for the regression. This way the resulting potential can be implemented on the physical system by tuning the stiffnesses of the springs according to the optimized parameters. In order to achieve physically realizable elements, a non-negative least squares solver [19] must be used. Another advantage of non-negative least squares is sparsity, i.e., one can remove all the elastic elements not creating any force from the system.

However, due to lack of space, the physical implementation of the optimized potential is not in focus in this paper and it will be addressed in future work. We, therefore, here choose to employ a neural network model with a single output for the parametric function $U_e(\boldsymbol{\omega}, \boldsymbol{q})$.

When implementing the potential $U_e$ on the mechanical system it will provide infinitely many strict modes. For conservative systems the modes will neither be attractive nor repulsive. External forces and modelling errors result in the system not stably oscillating on a desired mode. Also, when starting at rest the modal oscillation of the system must be excited. Friction and other dissipative terms remove energy from the system which must be reinjected.

For these reasons a controller is required for practical applications. The controller will stabilize the system onto the desired mode and keep the total energy at a desired level. Let $\boldsymbol{\tau}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})$ be that controller, then the dynamics of the system can, in coordinates, be written as

$$\underbrace{\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \frac{\partial U_g}{\partial \boldsymbol{q}}}_{\text{conservative mechanical system}} + \underbrace{\boldsymbol{d}(\boldsymbol{q}, \dot{\boldsymbol{q}})}_{\substack{\text{damping}\\\text{and friction}}} + \underbrace{\frac{\partial U_e}{\partial \boldsymbol{q}}}_{\substack{\text{elas. pot.}\\\text{(passive)}}} = \underbrace{\boldsymbol{\tau}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})}_{\substack{\text{controller}\\\text{(active)}}}.$$

A classical robotic system would generate the sum of the elastic potential and the active controller using the actuators. The main advantage of the proposed approach is that torques due to the potential $U_e$ are large compared to the controller torques and that $U_e$ creates *useful* torques, i.e., torques steering the system on a desired natural oscillation. Because the potential $U_e$ can be implemented by (nonlinear) elastic elements on the physical system, only the torques due to $\boldsymbol{\tau}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})$ need to be generated by the actuators. Therefore, a large part of required torques will be provided by the mechanics requiring smaller actuators and less actuator power. This allows to perform all of the modal oscillations and desired trajectories that are close to the modes very efficiently.

## IV. Case Study: Double Pendulum

We showcase the approach for a serial system with two degrees of freedom (Fig. 2). Further, we take the system outside of a gravitational field ($U_g = 0$). You can either think of a double pendulum in space or a two degree of freedom manipulator on earth parallel to the earth's surface. The system parameters are summarized in Tab. III in App. A. The potential-free dynamics is governed by (3).

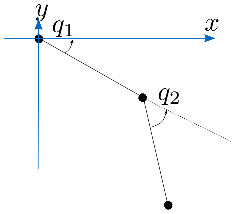
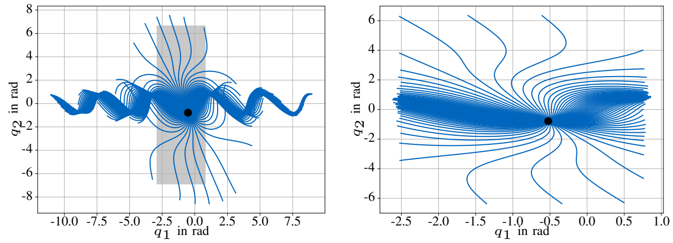


Fig. 2. Double Pendulum

An equilibrium $\boldsymbol{q}_{\text{eq}}$ for the target system must be selected. As derived before, the choice of equilibrium and the inertia tensor fully determines the shape of the strict normal modes of the target system. Therefore, the equilibrium configuration should be carefully selected, as it will determine the strict normal modes that can later be used.

We start with the choice $\boldsymbol{q}_{\text{eq}} = [-\pi/6, -\pi/4]^\mathsf{T}$ (Fig. 2). Using an Runge-Kutta integrator [20], we compute geodesics of the system by the proposed method. We set the initial configuration to $\boldsymbol{q}_0 = \boldsymbol{q}_{\text{eq}}$ and set an upper time limit in order to stop the integration. The initial velocity $\dot{\boldsymbol{q}}_0$ is set to unit vectors by sampling the unit-sphere equidistantly. Due to the nonlinearity, distributing the initial velocities uniformly does not imply that the configuration space is evenly covered with geodesics. However, for every point $\boldsymbol{q}$ in the subspace $\bar{\mathcal{Q}}$ a geodesic passing through it and the equilibrium can be found. Fig. 3a shows some of the computed geodesics. Note that the axes are in radians, so the trajectories correspond to long motions, where the joints are turned multiple times. In this large region $\mathcal{Q}$ the geodesics intersect, but when selecting a smaller region the intersections can be avoided. The gray rectangle in Fig. 3a shows one such region $\bar{\mathcal{Q}}$ and Fig. 3b shows it magnified.

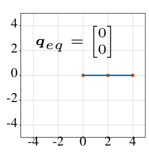
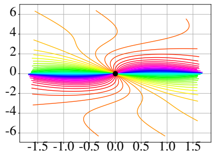
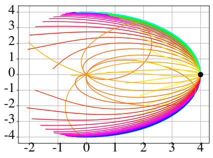
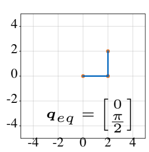
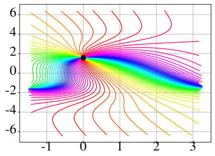
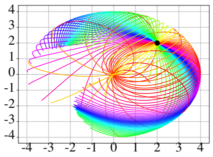
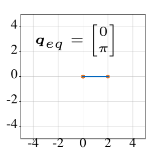
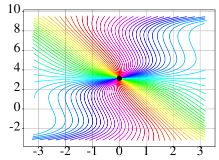
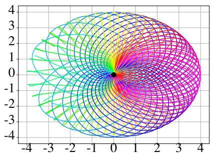


(a) Long Geodesics (b) Clipped to $\bar{\mathcal{Q}}$

Fig. 3. Geodesics of the example system. Restriction to a subregion results in a space where the geodesics only intersect at the equilibrium.

We perform this procedure for three different settings for the equilibrium $\boldsymbol{q}_{\text{eq}}$ and show resulting geodesics in Tab. I. The first column in Tab. I shows the equilibrium configuration, the second column the geodesics in a properly selected region $\bar{\mathcal{Q}}$ of the configuration space and the rightmost column shows the geodesics in Cartesian space. Using the color of the lines, the geodesics in configuration space can be matched to their Cartesian reflection. At this point we will continue the example with the elbow-up setting (row (b) in Tab. I) for the remainder of this section.

TABLE I
Comparison of different target equilibria.

| | **Equilibrium** | **Some Geodesics** | **In Cartesian Space** |
|---|---|---|---|
| (a) |  |  |  |
| (b) |  |  |  |
| (c) |  |  |  |

We choose a two-layer neural network with $\tanh$-activation functions and 200 neurons in the first and 100 neurons in the second layer as model function for $U_e(\boldsymbol{\omega}, \boldsymbol{q})$. The network model is implemented in TensorFlow [21] and the parameters are initialized randomly. In general we formulated the objective of finding the potential as constrained optimization problem in (7a)-(8). The cost function is only dependent on the output-to-input gradient of the neural network, not on the output itself. It is tuning the network very locally, but does not have a strong preference in the global shape of the potential. Therefore, we do not need to enforce the constraints (7ab) explicitly, but rather pre-train the network to a convex surface and then switch to the cost (8) to warp and stretch the initially trained surface to match the desired tangentiality condition (8).

In a first step the network is fitted to a parabola

$$V(\boldsymbol{q}) = (\boldsymbol{q} - \boldsymbol{q}_{\text{eq}})^\mathsf{T}(\boldsymbol{q} - \boldsymbol{q}_{\text{eq}}) \tag{9}$$

using an $L_2$-loss based on (9). The parameters $\boldsymbol{\omega}$ are optimized using the Adam algorithm [22]. A parabola is easy to fit, therefore we obtain a very low pre-training loss.

Afterwards, we switch to the loss function (8). During the training we frequently sample new training data in order to prevent overfitting. Although we have not explicitly enforced the constraints (7ab), the local nature of the loss function keeps the global bowl-like shape of the pre-trained parabola during the training.

In Fig. 5b, isolines of the resulting potential are shown in orange and Fig. 5a shows a surface plot. Additionally, the gray lines show some geodesics. We analyze how parallel $\boldsymbol{v}_1 = \boldsymbol{M}^{-1}(\boldsymbol{q})\frac{\partial U}{\partial \boldsymbol{q}}$ and $\boldsymbol{v}_2 = \dot{\gamma}$ are. Therefore, we compute the angle between the vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ for many points on randomly sampled geodesics and show a histogram in Fig. 4. The angles are computed in degrees, where an angle of 0° is optimal in a sense that $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are parallel.

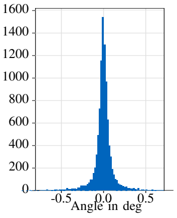
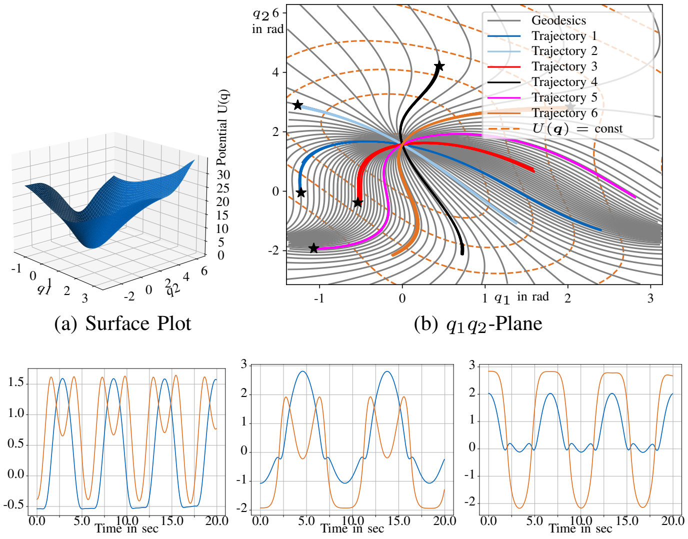


Fig. 4. Residuals



(a) Surface Plot (b) $q_1q_2$-Plane

(c) Trajectory 3 (d) Trajectory 5 (e) Trajectory 6

Fig. 5. Simulation of the double pendulum with the optimized potential for different initial conditions and visualization of the potential function.

For validation of the potential $U_e$ we apply the potential on the system dynamics, i.e., we simulate

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \frac{\partial U_e}{\partial \boldsymbol{q}} = \boldsymbol{0} \tag{10}$$

for some initial configurations and show the results in Fig. 5. The initial velocity is always set to zero. Panel (b) shows the simulated trajectories in the $q_1q_2$-plane. The black stars show the initial configurations. Moreover, we show time plots for three of the six trajectories (c-e). We would like to highlight that the results in Fig. 5 are without any active controller. The periodic motions are due to the properly tuned potential $U_e$.

The results in Fig. 5 confirm that the proposed approach leads to the desired result. We have shown six example modes, but infinitely many are present in the system. There is no preferred mode. Hence, when the system is slightly disturbed onto another mode, there is no mechanism pulling the same back onto the desired mode. When the system is subject to damping, the damping will dissipate energy from the system and the oscillation will decay. Moreover, the damping will most likely not be compatible with the mode, i.e., the damping forces may generally pull the system off the mode.

Hence, we design a controller stabilizing the system onto the desired mode and a controller injecting energy in the next section.

## V. Controllers

In this section two controllers will be derived and evaluated for the example system. The first controller will stabilize the system onto one of the infinitely many strict normal modes, while the other will swing up and keep the system at the desired energy level.

### A. Mode Selection Controller

In order to derive a mode selection controller some way of labeling the (infinitely many) strict normal modes is required. Generally, this can be achieved by a function $\boldsymbol{\theta} : \mathcal{Q} \to \mathcal{X}$, where $\dim \mathcal{X} = n-1$, i.e., the space of available strict modes is of dimension $n-1$. The value of $\boldsymbol{\theta}(\boldsymbol{q})$ will encode to which mode the point $\boldsymbol{q}$ belongs. Because all modes pass through the equilibrium $\boldsymbol{q}_{\mathrm{eq}}$, such a function cannot exist at that point.

Here, we have $n = 2$, and thus the labeling function will be a function into one dimension. The strict normal modes of the system can be identified by the direction of the velocity vector $\dot{\boldsymbol{q}}$ when it passes through the equilibrium. Note that $\dot{\boldsymbol{q}}$ and $-\dot{\boldsymbol{q}}$ identify the same mode and that the space of directions of a vector is not homeomorphic to the real numbers. Consequently, we design a function $\theta : \mathcal{Q} \to \mathbb{S}^1$.

Detailed derivations on the labeling function $\theta(\boldsymbol{q})$ are attached in the appendix App. B. We again used machine learning techniques to represent the function $\theta(\boldsymbol{q})$ by using a neural network model. Fig. 6 shows a color-coded rendering of the labeling function for our example. One can recognize from the rendering that, as expected, the level sets of $\theta(\boldsymbol{q})$ correspond to the strict modes of the system (see Fig. 5b).

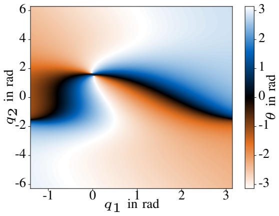


Fig. 6. Mode labeling function

Based on the labeling function $\theta(\boldsymbol{q})$ we derive the controller

$$\tilde{\boldsymbol{\tau}}_\theta = \boldsymbol{M}(\boldsymbol{q})\frac{\partial \theta}{\partial \boldsymbol{q}}\left(-k_\theta \Delta\theta(\boldsymbol{q},\theta_d) - d_\theta(\boldsymbol{q})\frac{\partial \theta}{\partial \boldsymbol{q}^\mathsf{T}}\dot{\boldsymbol{q}}\right), \tag{11}$$

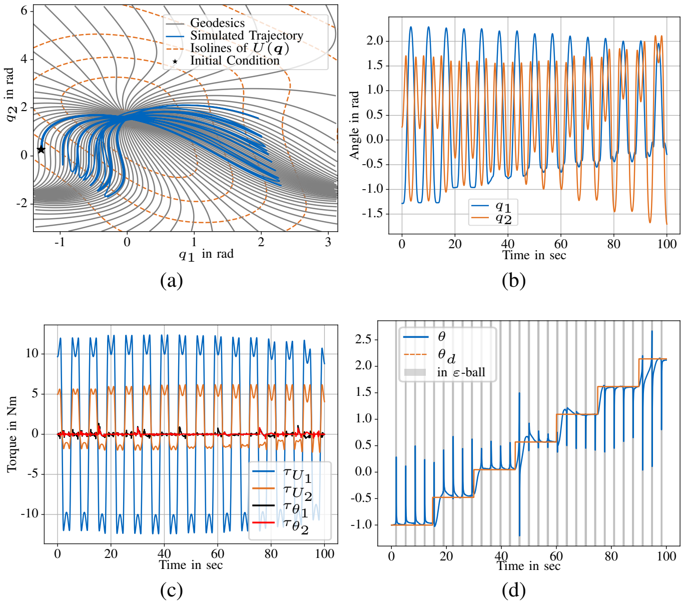
where $k_\theta$ is a proportional gain and the deviation from the desired strict mode $\theta_d$ is computed by the function $\Delta\theta(\boldsymbol{q},\theta_d)$. The equations for $\Delta\theta(\boldsymbol{q},\theta_d)$ (22b) and $\frac{\partial \theta}{\partial \boldsymbol{q}}$ (23) can also be found in App. B. The damping coefficient $d_\theta(\boldsymbol{q})$ (24b) is computed based on a desired damping ratio $\zeta$.

As the labeling function cannot exist at the equilibrium and does not perform well very close the equilibrium, we only enable the controller (11) when the system is outside an $\varepsilon$-ball centered at the equilibrium, i.e., we set

$$\boldsymbol{\tau}_\theta(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{cases} \boldsymbol{0} & \text{if } \|\boldsymbol{q}-\boldsymbol{q}_{\mathrm{eq}}\|_2 \leq \varepsilon, \\ \tilde{\boldsymbol{\tau}}_\theta(\boldsymbol{q},\dot{\boldsymbol{q}}) & \text{otherwise.} \end{cases} \tag{12}$$

The mode selection controller also allows to switch between various modes during operation. In (22b) the desired mode is specified by the real number $\theta_d$. Using this value the desired mode can be selected.

For the experiments, we set $k_\theta = 1$, $\zeta = 0.7$ and $\varepsilon = 0.1$ and apply the controller (12) together with the optimized potential on the system. We increase $\theta_d$ step-wisely by an amount of $\frac{\pi}{6}$ every $15s$. The results are shown in Fig. 7, where it becomes obvious that, compared to the torques due to the potential, the controller torques are quite low also when switching between modes. Fig. 7d shows a time evolution of the function $\theta(\boldsymbol{q}(t))$. Especially in this plot it becomes clear that the function cannot be used at the equilibrium. Each of the spikes in Fig. 7d corresponds to a passage through the equilibrium, where the function value is undefined. However, this does not compromise the performance as the controller (12) is disabled when close to the equilibrium (gray areas).



(a) (b) (c) (d)

Fig. 7. Simulation results with activated mode selection controller when switching the desired mode step-wisely.

## B. Energy Regulation

The task of the energy regulation controller is to swing up the system to the desired energy level. Additionally, during operation on the mode the energy regulation controller will compensate for energy losses by reinjecting dissipated energy.

The mechanical system in combination with the mode selection controller $\boldsymbol{\tau}_\theta$ will already be stabilized onto the desired mode. Therefore, negative damping can be used to inject energy into the system. We scale the amount of negative damping by the difference of the current energy estimate to the target energy $E_d$ and obtain

$$\tau_{\mathrm{nD}}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \beta_{\mathrm{nD}}\left(E_d - E(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}}, \tag{13}$$

where $\beta_{\mathrm{nD}}$ is a scalar constant and

$$E(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + U(\boldsymbol{q}). \tag{14}$$

This controller will in general not be compatible with the modes, i.e., the accelerations due to torques by this controller will not be tangent to the mode.

Let us consider the energy regulation controller by Della Santina et al. [10, III.D] as another candidate controller

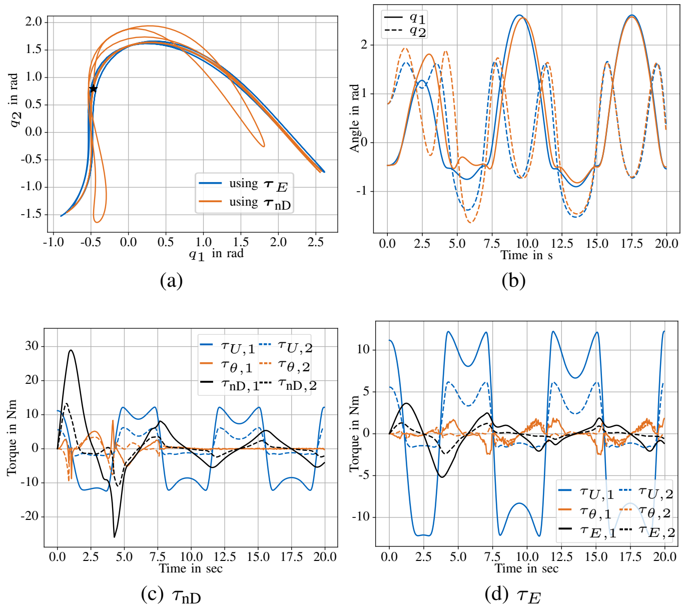
$$\boldsymbol{\tau}_E(\boldsymbol{q},\dot{\boldsymbol{q}}) = \beta_E \boldsymbol{M}(\boldsymbol{q})\left(E_d - E(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}}. \tag{15}$$

where $\beta_E$ is another scalar parameter. The controller (15) accelerates or decelerates the system along the velocity vector. Therefore, it will be tangent to the mode if we are on a mode.

We simulate the system with the potential $U_e$, the mode selection controller $\boldsymbol{\tau}_\theta$ and additionally model joint friction by a diagonal damping matrix $\boldsymbol{D} = 0.3\boldsymbol{I}$. Further, we initialize the system at a configuration $\boldsymbol{q}_0$ and set the initial velocity to zero. Our chosen $\boldsymbol{q}_0$ corresponds to the energy level of $E_0 = 5.0$J. The target energy level is set to $E_d = 20.0$J. We test the two candidate controllers $\boldsymbol{\tau}_{\mathrm{nD}}$ and $\boldsymbol{\tau}_E$ on this problem setting. The constant $\beta_{\mathrm{nD}}$ is set to $\beta_{\mathrm{nD}} = 0.4$ and we have tuned $\beta_E$ such that both controllers let the system converge to the same steady state energy after the same time. We got a value of $\beta_E = 0.13$.

The simulation results are shown in Fig. 8. It turns out that the two candidate controllers have advantages and disadvantages.

On the one hand, the mode-compatible controller $\boldsymbol{\tau}_E$ performs much better during the swing up phase. As can be seen in Fig. 8a (blue line) it accelerates the system along the mode. The control actions by $\boldsymbol{\tau}_\theta$ and $\boldsymbol{\tau}_E$ are dynamically decoupled, i.e., a control action by $\boldsymbol{\tau}_\theta$ will not create an immediate change in potential energy and a control action by $\boldsymbol{\tau}_E$ will not pull the system off the mode. In contrast, the simple negative-damping controller $\boldsymbol{\tau}_{\mathrm{nD}}$ shoots the system into a different direction (yellow line in Fig. 8a), which must be compensated by the mode stabilization controller $\boldsymbol{\tau}_\theta$ (Fig. 8c).



(a) (b)

(c) $\tau_{\mathrm{nD}}$ (d) $\tau_E$

Fig. 8. Damped system including the optimized potential, the mode selection controller and two types of energy regulation.

On the other hand, after the swing-up phase the ranking of performances is swapped. The negative damping controller $\boldsymbol{\tau}_{\mathrm{nD}}$ performs much better at compensating the damping as the structure is the same as the isotropic system damping. The mode-compatible controller $\boldsymbol{\tau}_E$ cannot compensate the system damping completely, as it has a different structure. Corrective torques by the mode selection controller $\boldsymbol{\tau}_\theta$ are required, as the system damping is not compatible with the modes (Fig. 8d).

We want to highlight, that the negative-damping controller performs so well after the swing up phase only because the modeled joint friction matches the structure of the controller. This is a match, which is quite unrealistic for real world scenarios. Therefore, for applications on hardware, a classical damping compensation should be used and the mode-compatible controller $\boldsymbol{\tau}_E$ should be used for energy regulation and swing up.

### C. Sensitivity Analysis

The final potential function $U_e$ is represented by a neural network, which provides a lot of flexibility and easily accessible algorithms for training. When moving from neural networks to actual mechanical elastic elements it will be hard to *exactly* implement the optimal potential. A mechanically implemented potential $\tilde{U}_e$ will in general not exactly resemble the optimal potential $U_e$. Although the mechanical implementation is not addressed in this paper, we perform a sensitivity analysis here. We add another parasitic potential to the optimal one and obtain the disturbed potential

$$\tilde{U}_e(\boldsymbol{q}) = U_e(\boldsymbol{q}) + \frac{\epsilon}{2}(\boldsymbol{q} - \boldsymbol{q}_{\mathrm{eq}})^{\mathsf{T}} \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} (\boldsymbol{q} - \boldsymbol{q}_{\mathrm{eq}}), \qquad (16)$$

i.e., we add two linear parasitic springs with stiffness $\epsilon\kappa_i$ to the system. The constants $\kappa_i$ are selected such that

$$\kappa_i = \frac{\tau_i^{\max}}{\Delta q_i^{\max}} \qquad (17)$$

where $\tau_i^{\max} = \max_{\bar{\mathcal{Q}}} \frac{\partial U_e}{\partial q_i}$ and $\Delta q_i^{\max} = \max_{\bar{\mathcal{Q}}} |q_i - q_{\mathrm{eq},i}|$, i.e., when setting $\epsilon = 1$ the torques to the parasitic potential grow as high as the ones due to the optimal potential $U_e$.

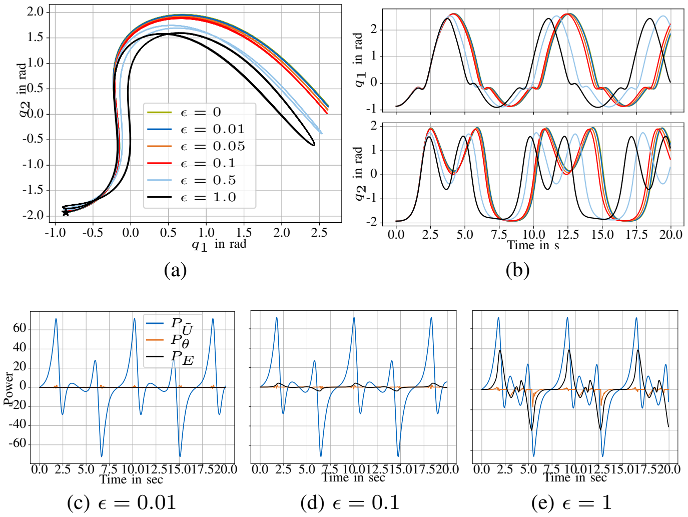
The system is simulated for different settings of $\epsilon$. Note that for $\epsilon = 0$ the system is the optimal one, i.e., the optimal potential $U_e$ is present. Both controllers, $\boldsymbol{\tau}_\theta$ and $\boldsymbol{\tau}_E$, are active with the same settings as before. Results are shown in Fig. 9. Up to a value of $\epsilon = 0.1$, the trajectory traced out in the $q_1 q_2$-plane (Fig. 9a) does not change significantly. However, the frequency of oscillation is slightly altered (Fig. 9b). For $\epsilon = 0.5$ and $\epsilon = 1.0$, we observe larger disturbances. We want to highlight that we introduced quite high parasitic forces. For $\epsilon = 1$ the torques due to the parasitic potential are on average as high as the torques due to the optimal potential $U_e$.

From the power plots (Fig. 9c-e) can be concluded that increased control action is required for increasing $\epsilon$. Especially the energy regulation controller (black lines) is removing and adding energy periodically. This is no surprise, as the control law (15) relies on the computation of the energy (14). Within (14) the optimal potential $U_e$ is used as the truly implemented potential $\tilde{U}_e$ is generally not known. Consequently, the energy regulation controller uses a false estimate of the energy.

In a next step we compute the total energy flux introduced by the individual torque components. Therefore, we integrate the absolute value of the powers using numerical integration

$$\int |P_\theta| \approx \sum \left|\boldsymbol{\tau}_\theta^{\mathsf{T}} \dot{\boldsymbol{q}}\right| \delta t. \qquad (18)$$

Likewise we compute the total energy fluxes $\int |P_{\tilde{U}}|$ for the disturbed potential and $\int |P_E|$ for the energy regulation controller. Table II shows the results. In a physical system the energy flux in the first row is handled by elastic elements, which store and release energy. Therefore, for ideal springs, the first row does not effect the energy efficiency. The last two rows correspond to energy fluxes through the actuators of the system. As electrical actuators and their drivers usually do not recover energy, this energy is lost. Hence, for energy efficient system the values in the last two rows should be as small as possible.



(a) (b)

(c) $\epsilon = 0.01$ (d) $\epsilon = 0.1$ (e) $\epsilon = 1$

Fig. 9. Results for the sensitivity analysis. The parasitical potential is increased over the experiments. The top row shows the evolution of the system in terms of the joint angles. The bottom row shows power flow due to the individual torques sources for three different values of $\epsilon$.

TABLE II
COMPARISON OF TOTAL ENERGY FLUXES

| $\epsilon$ | 0 | 0.01 | 0.05 | 0.1 | 0.5 | 1.0 |
|---|---|---|---|---|---|---|
| $\int \lvert P_{\tilde{U}} \rvert$ | 354.56 | 356.23 | 363.25 | 372.43 | 447.18 | 565.60 |
| $\int \lvert P_\theta \rvert$ | 2.86 | 2.85 | 2.94 | 3.74 | 11.72 | 20.92 |
| $\int \lvert P_E \rvert$ | 0.03 | 1.87 | 9.44 | 18.75 | 89.264 | 181.83 |

We conclude that the final system admits certain tolerances in the potential implemented by mechanical elements. The controllers are able to compensate inaccuracies in the potential to a certain extent. On the one hand, for moderate inaccuracies of the potential, the system trajectories change shape only slightly. However, the timing and frequency of oscillation changes. The aim of this work was not to do trajectory tracking, but to basically retain as much as possible from the dynamics of the mechanical system. On the other hand, the energy efficiency decreases when the implemented potential deviates from the optimized one.

## VI. CONCLUSION

In this paper we have developed a method to find a potential that densely endows the configuration space with strict normal modes, which describe natural periodic oscillations of the system including the potential. The main take-off message is that, by designing the potential field to harmonize with the inertial properties, i.e., with the natural Riemannian metric of the system, the highly nonlinear system can display a large variety of very regular motions.

The implementation of the potential directly in the mechanics by combining nonlinear elastic elements that create the optimized potential was not within the scope of this paper and will be addressed in future work. However, considering the sensitivity analysis, it can be concluded that certain discrepancies between the implemented and the optimal potential can be tolerated.

The strict modes allow to efficiently execute a variety of periodic oscillations. Stabilization onto a desired strict mode can be realized by the proposed mode stabilization controller. Further the system can be swung up to the desired energy level using the energy regulation controller. The latter also reinjects energy that is lost due to damping or interactions. We have shown in simulations experiments that the control torques are very low compared to the torques due to the elastic elements.

## Appendix A
## Parameters of the Double Pendulum

TABLE III

| | Link 1 | Link 2 |
|---|---|---|
| Mass | $m_1 = 1\,\mathrm{kg}$ | $m_2 = 1\,\mathrm{kg}$ |
| Length | $l_1 = 2\,\mathrm{m}$ | $l_2 = 2\,\mathrm{m}$ |
| Center of Mass from Origin | $l_{c1} = 1\,\mathrm{m}$ | $l_{c2} = 1\,\mathrm{m}$ |
| Moments of Inertia | $I_1 = 1\,\mathrm{kg\,m^2}$ | $I_2 = 1\,\mathrm{kg\,m^2}$ |

## Appendix B
## Mode Identification Coordinate

The direction of the velocity when passing through the equilibrium is used to identify a strict mode, where velocity vectors $\dot{\boldsymbol{q}}$ and $-\dot{\boldsymbol{q}}$ identify the same mode. Therefore, we set the label of the strict mode to

$$\theta = 2\,\mathrm{atan2}(\dot{q}_2, \dot{q}_1), \tag{19}$$

where $\dot{q}_1$ and $\dot{q}_2$ are the two components of the velocity vector at the equilibrium. The factor of 2 ensures that $\dot{\boldsymbol{q}}$ and $-\dot{\boldsymbol{q}}$ correspond to the same label $\theta$.

In order to obtain a function $\theta(\boldsymbol{q})$ encoding the label of the mode passing through $\boldsymbol{q}$, we generate system geodesics $\boldsymbol{\gamma}_i$ and assign a label $\theta_i$ to them according to (19). Each geodesic is represented by $T_i$ samples $\boldsymbol{\gamma}_{it} \in \bar{\mathcal{Q}}$. This creates an amount of $\sum_i T_i$ training tuples $(\boldsymbol{\gamma}_{it}, \theta_i)$. However, the raw function $\theta(\boldsymbol{q})$ cannot be approximated by a scalar function as it parametrizes $\mathbb{S}^1$ and it is not defined at the equilibrium. We therefore define another function

$$\Lambda(\boldsymbol{q}) = ||\boldsymbol{q} - \boldsymbol{q}_{\mathrm{eq}}||^2 \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix}, \tag{20}$$

which can be used to reconstruct $\theta(\boldsymbol{q})$ by

$$\theta(\boldsymbol{q}) = \mathrm{atan2}(\Lambda_2(\boldsymbol{q}), \Lambda_1(\boldsymbol{q})). \tag{21}$$

In contrast to $\theta(\boldsymbol{q})$ the function $\boldsymbol{\Lambda}(\boldsymbol{q})$ is smooth. We take a neural network with $\tanh$ activation functions, two hidden layers and two outputs and use it to approximate $\boldsymbol{\Lambda}(\boldsymbol{q})$ based on the training data $(\boldsymbol{\gamma}_{it}, \theta_i)$.

As $\theta \in \mathbb{S}^1$ some care must be taken when computing the difference to the desired mode $\theta_d$. Let

$$R_d = \begin{bmatrix}\cos\theta_d & -\sin\theta_d \\ \sin\theta_d & \cos\theta_d\end{bmatrix} \text{ and } A(\boldsymbol{q}) = \begin{bmatrix}\Lambda_1(\boldsymbol{q}) & -\Lambda_2(\boldsymbol{q}) \\ \Lambda_2(\boldsymbol{q}) & \Lambda_1(\boldsymbol{q})\end{bmatrix}.$$

The basic idea is to express $\theta(\boldsymbol{q})$ as rotation matrix. We need to orthonormalize $A(\boldsymbol{q})$ by using the singular value decomposition to obtain $U\Sigma V^{\mathsf{T}} = A(\boldsymbol{q})$ and set $Q(\boldsymbol{q}) = UV^{\mathsf{T}}$, then

$$R_{\Delta}(\boldsymbol{q}, \theta_d) = R_d^{\mathsf{T}} Q(\boldsymbol{q}) \tag{22a}$$

$$\Delta\theta(\boldsymbol{q}, \theta_d) = \mathrm{atan2}(r_{21}(\boldsymbol{q}, \theta_d), r_{11}(\boldsymbol{q}, \theta_d)), \tag{22b}$$

where $r_{ij}(\boldsymbol{q}, \theta_d)$ are the respective elements in $R_{\Delta}(\boldsymbol{q}, \theta_d)$. The derivative of $\Delta\theta(\boldsymbol{q})$ with respect to $\boldsymbol{q}$ can be computed using

$$\frac{\partial\theta}{\partial\boldsymbol{q}^{\mathsf{T}}} = \begin{bmatrix}-\sin\Delta\theta & \cos\Delta\theta\end{bmatrix} \frac{\partial\boldsymbol{\Lambda}}{\partial\boldsymbol{q}}. \tag{23}$$

For the damping design we first transform the mass-matrix $\boldsymbol{M}(\boldsymbol{q})$ of the manipulator to $\theta(\boldsymbol{q})$ coordinates and then compute the damping coefficient

$$m_{\theta}(\boldsymbol{q}) = \frac{1}{\left(\frac{\partial\theta}{\partial\boldsymbol{q}}\right)^{\mathsf{T}} \boldsymbol{M}^{-1}(\boldsymbol{q}) \frac{\partial\theta}{\partial\boldsymbol{q}}} \tag{24a}$$

$$d_{\theta}(\boldsymbol{q}) = 2\zeta\sqrt{k_{\theta} m_{\theta}(\boldsymbol{q})}\,. \tag{24b}$$

## References

[1] A. Albu-Schäffer, D. Lakatos, and S. Stramigioli, “Strict nonlinear normal modes of systems characterized by scalar functions on riemannian manifolds,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1910–1917, 2021.

[2] J. Englsberger *et al.*, “Overview of the torque-controlled humanoid robot toro,” in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 916–923.

[3] T. Shinbrot, C. Grebogi, J. Wisdom, and J. A. Yorke, “Chaos in a double pendulum,” *American Journal of Physics*, vol. 60, no. 6, pp. 491–499, 1992.

[4] A. Albu-Schäffer and C. Della Santina, “A review on nonlinear modes in conservative mechanical systems,” *Annual Reviews in Control*, vol. 50, pp. 49 – 71, 2020.

[5] D. C. Remy, “Optimal exploitation of natural dynamics in legged locomotion,” Ph.D. dissertation, Eidgenössische Technische Hochschule Zürich, 2011.

[6] Z. Gan and C. D. Remy, “A passive dynamic quadruped that moves in a large variety of gaits,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4876–4881.

[7] N. Kashiri *et al.*, “An overview on principles for energy efficient robot locomotion,” *Frontiers in Robotics and AI*, vol. 5, no. 129, 2018.

[8] B. Vanderborght *et al.*, “Exploiting natural dynamics to reduce energy consumption by controlling the compliance of soft actuators,” *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 343–358, 2006.

[9] C. Della Santina, D. Calzolari, A. M. Giordano, and A. Albu-Schäffer, “Actuating Eigenmanifolds of Conservative Mechanical Systems via Bounded or Impulsive Control Actions,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2783–2790, Apr. 2021.

[10] C. Della Santina and A. Albu-Schäffer, “Exciting Efficient Oscillations in Nonlinear Mechanical Systems Through Eigenmanifold Stabilization,” *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 1916–1921, 2021.

[11] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. Springer, New York, 1997.

[12] L. P. Eisenhart, “Dynamical trajectories and geodesics,” *Annals of Mathematics*, vol. 30, no. 1/4, pp. 591–606, 1928.

[13] V. Arnold, *Mathematical methods of classical mechanics*, S. Axler, F. W. Gehring, and K. A. Ribet, Eds. Springer, 1989.

[14] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, 3rd ed. Cambridge University Press, 2007.

[15] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Berlin, Heidelberg: Springer, 2006.

[17] B. Vanderborght *et al.*, “Variable impedance actuators: A review,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601–1614, 2013.

[18] S. Wolf *et al.*, “Variable stiffness actuators: Review on design and components,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2418–2430, 2016.

[19] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[20] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Berlin Heidelberg, 2008.

[21] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]: https://arxiv.org/abs/1603.04467

[22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Online]: https://arxiv.org/abs/1412.6980