IX International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2021
E. Oñate, M. Papadrakakis and B. Schrefler (Eds)

FEEDBACK CONTROL FOR A COUPLED SOFT TISSUE SYSTEM BY KERNEL SURROGATES

TOBIAS EHRING* AND BERNARD HAASDONK*

*Institute of Applied Analysis and Numerical Simulation (IANS)
University of Stuttgart
70174 Stuttgart, Germany
e-mail: ehringts@mathematik.uni-stuttgart.de, haasdonk@mathematik.uni-stuttgart.de
web page: https://www.ians.uni-stuttgart.de/anm

Key words: Feedback control, Soft tissue robotics, Kernel methods, Surrogate modeling

Abstract. The problem of closed-loop control of coupled systems under state constraints is considered. A new method for generating an approximate feedback policy is used in which the gradient of the value function is replaced by a vector-valued kernel surrogate. This is built up from samples from open-loop control problems. Here, additional information about the system originating from the Pontryagin Maximum Principle is exploited. Furthermore, a multi-stage approach and the vectorial kernel orthogonal greedy algorithm are used. With this procedure we can overcome the curse of dimensionality that occurs in the determination of the value function via Hamilton-Jacobi-Bellman equation. Nevertheless, the resulting feedback control is very accurate, robust and real-time capable.

1 INTRODUCTION

Nowadays the control of dynamical systems has a very wide range of applications. Examples can be found in large industrial production plants for controlling robots, but also in almost every smartphone, as these, for instance, can adjust the screen brightness to the lighting conditions.

This work deals with the realization of a feedback control for a coupled system coming from the field of soft tissue robotics. The model is inspired from [1] and it describes a gripper that moves a soft tissue object to a prescribed target position while avoiding an obstacle in a cost-optimal manner (see Figure 1). Such processes can be described mathematically by the following optimal control problem (OCP) with state constraints:

$$\min_{\mathbf{u}\in\mathcal{U}_\infty} J(\mathbf{u}) = \min_{\mathbf{u}\in\mathcal{U}_\infty} \int_0^\infty r(\mathbf{x}(s),\mathbf{u}(s))\,\mathrm{d}s \tag{1}$$

$$\text{subject to } \dot{\mathbf{x}}(s) = f(\mathbf{x}(s),\mathbf{u}(s)),\ \mathbf{x}(0) = x_0 \tag{2}$$

$$\text{and } C_i(\mathbf{x}(s)) \leq 0 \text{ for all } s \in [0,\infty) \text{ and } i = 1, ..., c. \tag{3}$$

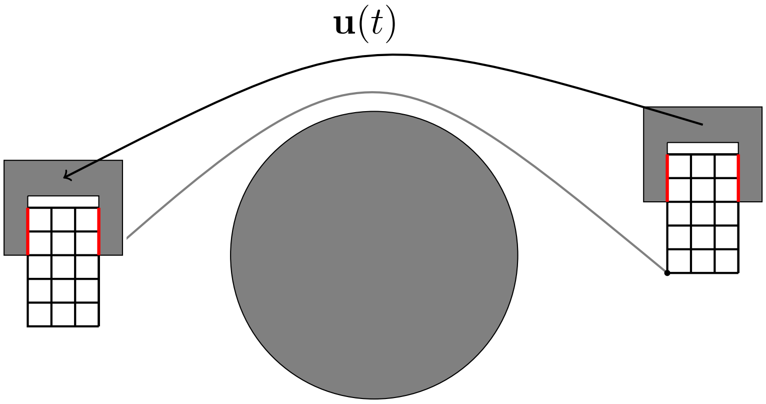


Figure 1: A gripper that transports soft tissue to a target point while avoiding an obstacle.

Here $\mathbf{x} : [0, \infty) \to \mathbb{R}^N$ is the state trajectory and $f : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$ a globally Lipschitz continuous function. In addition, the space of all possible control signals is defined as

$$\mathcal{U}_\infty := \{\mathbf{u} : [0, \infty) \to U \,|\, u \text{ measurable}\},$$

where $U \subseteq \mathbb{R}^M$ is a non-empty set. The function $J : \mathcal{U}_\infty \to \mathbb{R}_+$ is the cost function. It consists of the integral over the running payoff $r : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}_+$, which we assume to be continuous. As the constraint (2) uniquely determines the dynamical system for a given $\mathbf{u}$, the cost function depends notationally only on $\mathbf{u}$.

The key tool for providing a feedback policy is the cost-to-go depending on the initial state. This function is known as the value function (VF). From Bellman's Dynamic Programming Principle it can be deduced that it is the solution of the partial differential equation (PDE) called Hamilton-Jacobi-Bellman (HJB) equation. However, the problem with solving the HJB equation is that classical numerical solution algorithms for PDEs suffer from the curse of dimensionality. More advanced techniques, as in [2, 3], are required to get a feedback control for higher dimensional problems. Since the dimension N of our Gripper-Soft-Tissue (GST) model is $N > 6$, it is high-dimensional in this context and therefore we are also faced with this issue. Our approach to finding a feedback rule $\mathcal{K}$ that maps the current state to the optimal signal is based on the idea in [4] and proceeds as follows. First, many open-loop processes for different starting positions are performed in an offline phase utilizing the Pontryagin Maximum Principle (PMP). In an open-loop process, the optimal signal is determined in advance and no further adaptation of the signal to the current state is possible in an ongoing process. Therefore, in real-world applications, feedback control (closed-loop) is preferred because it can react to unpredictable disturbances. However, the data obtained through the open-loop processes contains information about the gradient of the VF (GVF). Kernel interpolation techniques are then applied to this data set to obtain a surrogate. Such kernel techniques are typically robust against the curse of dimensionality. In addition, they have the advantage that the data set can be arbitrary scattered and does not have to be structured like a triangulation or grid as in [5], for example. After the surrogate has been computed, it can be used for a closed-loop process in an online phase. This allows real time computation and gives still very accurate results. In addition, this procedure is much more robust to disturbances.

To the best of our knowledge, this is the first time that the kernel approximation technique has been applied to obtain a surrogate for the GVF to implement feedback control. A similar work is [6], where a neural network is trained on open-loop data in order to approximate the VF.

The paper is organized as follows. Section 2 provides a brief background on optimal control. The GST model is then presented in Section 3. In particular, with respect to this model, a numerical method is introduced in Section 4, which is used to generate a surrogate of the GVF and in Section 5, numerical tests are performed on the GST model. At the end, we give a conclusion in Section 6.

2 PRELIMINARIES

Solving a problem like (1)–(3) is very complex. Typically, direct methods [7] are used in which the problem is converted into a non-linear programming problem by discretization. The infinite horizon can be handled like in [8], through an exact time transformation to a finite interval. This, however, completely destroys the underlying structure of the problem. Furthermore, with direct methods we do not gain any additional information about the system, in particular not about the GVF. This requires indirect methods using PMP-like conditions. However, for these prior knowledge about the activation of the state constraints is necessary [7] and further these are difficult to implement. For this reason, we will transform the problem into a fast solvable, convenient formulation in two steps.

The first approximation step is to handle the state constraints through an exterior penalty method. For $\beta > 1$ we choose the unconstrained OCP as follows:

$$\min_{\mathbf{u}\in\mathcal{U}_\infty} J_{\text{pen}}(\mathbf{u}) = \min_{\mathbf{u}\in\mathcal{U}_\infty} \int_0^\infty \underbrace{r(\mathbf{x}(s),\mathbf{u}(s)) + \frac{1}{\sqrt{\beta}}\sum_{i=1}^{n} e^{\beta C_i(\mathbf{x}(s))}}_{=:r_{\text{pen}}(\mathbf{x}(s),\mathbf{u}(s))} \,\mathrm{d}s \tag{4}$$

$$\text{subject to } \dot{\mathbf{x}}(s) = f(\mathbf{x}(s),\mathbf{u}(s)) \text{ and } \mathbf{x}(0) = x_0 \text{ for all } s \in [0,\infty). \tag{5}$$

The optimal process of this infinite horizon OCP is denoted by $(\mathbf{x}^*, \mathbf{u}^*)$. In this, exterior means that the solution of (4) can slightly violate the state constraints. There are also interior penalty methods [9] in which the penalty term in the cost function diverges for states that reach the boundary from the feasible set. Therefore, only feasible processes are solutions of these methods. However, the latter are numerically much more difficult to handle and less robust. For this reason, we use this simple structured differentiable penalty term (4) for which, as a function of β, the cost increases exponentially for states that violate the constraints and decreases exponentially for states that do not. Thus, for a sufficiently large β less violation of the constraints by the optimal trajectory is ensured.

Next, we introduce the VF, which for us is the key to feedback control.

$$v(x) := \min_{u \in \mathcal{U}_\infty} \int_0^\infty r_{\text{pen}}(\mathbf{x}(s), \mathbf{u}(s))\, \mathrm{d}s \text{ subject to } \dot{\mathbf{x}}(s) = f(\mathbf{x}(s), \mathbf{u}(s)) \text{ and } \mathbf{x}(0) = x.$$

In contrast to the previous OCP formulations, here the initial state is denoted by x, since it is the free variable of the VF. The latter solves the HBJ equation (see [10]):

$$\min_{u \in U} \{f(x, u) \cdot \nabla_x v(x) + r(x, u)\} = 0 \tag{6}$$

Compared to (4), this is a local minimization problem and the signal u, which solves (6) for a given state x, is the optimal one. Hence, the feedback rule is

$$\mathcal{K}(x; \nabla v) := \arg\min_{u \in U} \{f(x, u) \cdot \nabla v(x) + r(x, u)\}\,. \tag{7}$$

So if the GVF ∇v is known, this can be used to obtain a feedback policy. Solving the HJB is not possible in higher dimensions. For this reason, we approximate the GVF with a kernel interpolant s_n. Here, n is the expansion size (see Sec. 4). This then leads to an approximate feedback rule $\mathcal{K}(x; s_n)$.

The second approximation step is a truncation of the infinite horizon in (4) for computational purposes. Therefore, for a sufficiently large T, we consider the new problem

$$v_T(x, t) := \min_{\mathbf{u} \in \mathcal{U}_T} J(\mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}_T} \int_t^T r_{\text{pen}}(\mathbf{x}(s), \mathbf{u}(s))\, \mathrm{d}s \tag{8}$$

$$\text{subject to } \dot{\mathbf{x}}(s) = f(\mathbf{x}(s), \mathbf{u}(s)) \text{ and } \mathbf{x}(t) = x \text{ for all } s \in [t, T]. \tag{9}$$

The optimal process of the OCP with a finite horizon is denoted by $(\mathbf{x}_T^*, \mathbf{u}_T^*)$. The hope now is that $(\mathbf{x}_T^*, \mathbf{u}_T^*)$ will mimic the behavior of $(\mathbf{x}^*, \mathbf{u}^*)$ sufficiently well. But an approximation to the infinite horizon OCP with finite horizon OCPs is generally not possible. This issue is closely related to the stability of model predictive control and is examined in this context in [11]. A more general investigation of this question can be found in [12, 13]. Since this is all going too far for this presentation, we just assume

$$\lim_{T \to \infty} \max_{t \in [0, T]} \|\nabla_x v_T(\mathbf{x}_T(t), t) - \nabla_x v_\infty(\mathbf{x}^*(t))\|_{\mathbb{R}^N} = 0, \tag{10}$$

for every initial state $x \in \mathbb{R}^N$, which is the only property that we will actually need in the following. It is a kind of uniform convergence of the GVF along optimal characteristics.

For the OCP with a finite horizon in (8), a common version of the PMP can be used to specify necessary conditions that an optimal process must satisfy. Under appropriate conditions (see [10]) there exists for an optimal process $(\mathbf{x}_T^*, \mathbf{u}_T^*)$ a co-state $\mathbf{p}_T^*$, such that

$$\dot{\mathbf{x}}_T^*(s) = \nabla_p H\left(\mathbf{x}_T^*, \mathbf{p}_T^*, \mathbf{u}_T^*\right), \quad \mathbf{x}_T^*(0) = x \tag{11}$$

$$\dot{\mathbf{p}}_T^*(s) = -\nabla_x H\left(\mathbf{x}_T^*, \mathbf{p}_T^*, \mathbf{u}_T^*\right), \ \mathbf{p}_T^*(T) = 0 \tag{12}$$

$$\mathbf{u}_T^*(s) = \arg\min_{u \in U} \left\{H\left(\mathbf{x}_T^*, \mathbf{p}_T^*, u\right)\right\} \tag{13}$$

are fulfilled for $s \in [0,T]$, where $H(x,p,u) = f(x,u) \cdot p + r(x,u)$ is the control Hamiltonian. Furthermore, the following relation is well known

$$\mathbf{p}_T^*(s) = \nabla_x v_T(\mathbf{x}_T^*(s), s) \approx \nabla_x v_\infty(\mathbf{x}^*(s)),$$

where the approximation due to Assumption (10) holds. This makes it possible to generate samples of the GVF along the entire optimal trajectory, when the co-state is known.

The conditions in (11) – (13) can be formulated into an inhomogeneous Dirichlet two-point boundary value problem, among whose solutions is the optimal one. To see this, we introduce the vector $\mathbf{z}_T^*(s) := \begin{bmatrix} \mathbf{x}_T^*(s) & \mathbf{p}_T^*(s) \end{bmatrix}^\top$ and define the right-hand side as

$$F(\mathbf{z}_T^*) := \begin{bmatrix} \nabla_p H\left(\mathbf{z}_T^*, \arg\min_{u\in U}\left\{H\left(\mathbf{z}_T^*, u\right)\right\}\right) \\ -\nabla_x H\left(\mathbf{z}_T^*, \arg\min_{u\in U}\left\{H\left(\mathbf{z}_T^*, u\right)\right\}\right) \end{bmatrix}.$$

The boundary conditions in (11) and (12) can also be represented compactly by

$$b(\mathbf{z}_T^*(0), \mathbf{z}^*(T)) := \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \mathbf{z}_T^*(0) + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \mathbf{z}_T^*(T) - \begin{bmatrix} x_0 \\ 0 \end{bmatrix}.$$

Here, $I \in \mathbb{R}^{N\times N}$ is the $(N \times N)$-unit matrix. All in all, we have to solve

$$\dot{\mathbf{z}}_T^*(s) = F(\mathbf{z}_T^*) \text{ with } b(\mathbf{z}_T^*(0), \mathbf{z}_T^*(T)) = 0.$$

Note that the solution $\mathbf{z}_T^*(s)$ provides also information about the GVF via the co-state.

3 A COUPLED SOFT TISSUE OPTIMAL CONTROL PROBLEM

The GST is a model with a two-dimensional physical domain, which describes a gripper that has gripped a soft tissue, for example a fruit or meat, and brings it to a prescribed target position. This target position is specified via the lower left corner. The soft tissue is modeled by an elastic body and the gripper by a rigid body. An external force $\mathbf{u}(s)$ controls the gripper in this process. Figure 1 shows a schematic representation. The coupling boundary Γ_N between the gripper and the soft tissue is marked in red.

First, we derive spatial discretized equations of motion for the solid body (gripper) and the elastic body (soft tissue) successively. For the latter the governing equation of a linear elastic body is used to describe the displacement field. This is given by

$$\rho \ddot{q}(s,x) = \operatorname{div}(\sigma(q(s,x))) \tag{14}$$

for each $x \in \Omega_e$ and $s \in [0,T]$ with the boundary condition $\sigma(q(s,x)) \cdot n = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ on Γ_N. The body forces are neglected. Here, ρ is the density of the elastic body and with $\sigma(q)$ we denote the Cauchy stress tensor. By converting (14) to the weak form and using the finite element method, the ODE systems

$$\begin{bmatrix} M_{cc} & M_{ce} \\ M_{ec} & M_{ee} \end{bmatrix} \begin{bmatrix} \ddot{\underline{\mathbf{q}}}_c(s) \\ \ddot{\underline{\mathbf{q}}}_e(s) \end{bmatrix} + \begin{bmatrix} K_{cc} & K_{ce} \\ K_{ec} & K_{ee} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{q}}_c(s) \\ \underline{\mathbf{q}}_e(s) \end{bmatrix} + \begin{bmatrix} -W_{cc} & -W_{ce} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{q}}_c(s) \\ \underline{\mathbf{q}}_e(s) \end{bmatrix} = 0$$

follows, where we now differentiate between nodes at the coupling boundary ($\underline{\mathbf{q}}_c(s)$) and inner nodes ($\underline{\mathbf{q}}_e(s)$). The matrix entries in this ODE system consist of integrals of products of the ansatz functions or their gradients.

The gripper is a rigid body that only has translational degrees of freedom. For this reason we can describe it, in contrast to the displacement field of the elastic body, by the movement of a point mass. With Newton's second law of motion we have

$$m_s \cdot \underbrace{\begin{bmatrix} \ddot{\mathbf{q}}_s^x(s) \\ \ddot{\mathbf{q}}_s^y(s) \end{bmatrix}}_{:=\ddot{\underline{\mathbf{q}}}_s(s)} = \underbrace{\begin{bmatrix} \mathbf{f}_s^x(s) \\ \mathbf{f}_s^y(s) \end{bmatrix}}_{=:\mathbf{f}_s(s)} - \underbrace{\begin{bmatrix} \mathbf{f}_c^x(s) \\ \mathbf{f}_c^y(s) \end{bmatrix}}_{=:\mathbf{f}_c(s)}. \tag{15}$$

Here, $m_s \in \mathbb{R}$ is the mass of the gripper and $\mathbf{q}_s^x, \mathbf{q}_s^y : [0,T] \to \mathbb{R}$ the displacement of the center of gravity in x- and y-direction. The right-hand side consists of two parts because there are two sources of forces acting on the rigid body. The first are the external forces $\mathbf{f}_s$, which arise through the controller. That is why we use the notation $\mathbf{f}_s^x(s) = \mathbf{u}^x(s)$ and $\mathbf{f}_s^y(s) = \mathbf{u}^y(s)$. The second part are forces which originate from the elastic body and act on the rigid body through the coupling over the coupling boundary. These are given by

$$\mathbf{f}_c(s) = \int_{\Gamma_D} \sigma(q(s,x)) \cdot n \, \mathrm{d}\mu. \tag{16}$$

Inserting the finite element approximation of $q(s,x)$ in (16) leads with (15) to

$$m_s \ddot{\underline{\mathbf{q}}}_s(s) + B_{sc} W_{cc} \underline{\mathbf{q}}_c(s) + B_{sc} W_{ce} \underline{\mathbf{q}}_e(s) = \mathbf{u}(s).$$

Next, we couple these two bodies. We want the displacement of the rigid body and the coupling boundary nodes to be the same. With the D'Alembert principle, this constraint leads to a constraint force in the momentum balance of the whole body system. As in this two-dimensional case, in which only planar movements of the GST model are permitted, the principle of virtual work applies. For this reason, this constraint force is orthogonal to the virtual displacement, which ultimately leads to a system of the form

$$M_{GS} \begin{bmatrix} \ddot{\underline{\mathbf{q}}}_e(s) \\ \ddot{\underline{\mathbf{q}}}_s(s) \end{bmatrix} + (\alpha_R M_{GS} + \beta_R K_{GS}) \begin{bmatrix} \dot{\underline{\mathbf{q}}}_e(s) \\ \dot{\underline{\mathbf{q}}}_s(s) \end{bmatrix} + K_{GS} \begin{bmatrix} \underline{\mathbf{q}}_e(s) \\ \underline{\mathbf{q}}_s(s) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{u}(s) \end{bmatrix}$$

with α_R, $\beta_R \in \mathbb{R}_+$. Note that we have also inserted Rayleigh damping [14] here by introducing a term depending on the first derivative of the displacement. This does not follow from the physics in (14)–(16), but is a common way to account for the overall damping of the system. The parameters α_R and β_R characterize the materials used.

By putting the system into a first order form, it becomes

$$\begin{aligned} E\dot{\mathbf{x}}(s) &= A\mathbf{x}(s) + D\mathbf{u}(s) \text{ for } s \in [0,T] \\ \mathbf{x}(0) &= x_0. \end{aligned}$$

Here, $\mathbf{x}(s) := \begin{bmatrix} \underline{\mathbf{q}}_e(s) & \underline{\mathbf{q}}_s(s) & \dot{\underline{\mathbf{q}}}_e(s) & \dot{\underline{\mathbf{q}}}_s(s) \end{bmatrix}^\top \in \mathbb{R}^N$, $x_0 \in \mathbb{R}^N$ is the starting state of the system and the matrices $E, A \in \mathbb{R}^{N\times N}$ and $D \in \mathbb{R}^{N\times 2}$ are chosen accordingly.

We now formulate an optimal control problem that determines the control signal $\mathbf{u}(s)$ in an optimal way. To define what optimal means, we need to specify the running payoff function r. It should penalize the distance to the target position, the velocity and also control signal magnitude. The optimal control signal is therefore the control signal that has the best balance between reaching the target position, the shakiness of the soft tissue and the the control signal energy. Such a running payoff function is given by

$$r(\mathbf{x}(s), \mathbf{u}(s)) = \mathbf{x}(s)^\top Q\mathbf{x}(s) + \mathbf{u}(s)^\top R\mathbf{u}(s)$$

where $Q \in \mathbb{R}^{N\times N}$ is a positive definite matrix and $R \in \mathbb{R}^{N\times 2}$. Notice that we are starting with an initial displacement and therefore the target position is in the state where we have zero displacement.

In addition, we need to introduce constraint functions that should lead to the avoidance of an obstacle, e.g. a ball (see Fig. 1). Therefore, we need matrices $\tilde{C}_i \in \mathbb{R}^{2\times N}$ with

$$\tilde{C}_i\mathbf{x}(s) = \begin{bmatrix} \left(\underline{\mathbf{q}}_e^x(s)\right)_{l_i} & \left(\underline{\mathbf{q}}_e^y(s)\right)_{l_i} \end{bmatrix}^\top,$$

where l_i is the index of the ith-node, to extract the location of each node, for $i = 1, ..., n$. Then we fix a radius $r_{\text{ob}} \in \mathbb{R}_+$ and a center $c_{\text{ob}} \in \mathbb{R}^2$. With this the resulting function

$$C_i(\mathbf{x}(s)) = r_{\text{ob}}^2 - \left\| c_{\text{ob}} - \tilde{C}_i \left(\begin{bmatrix} \underline{r}_{e,0} & \underline{r}_{s,0} & 0 \end{bmatrix}^\top + \mathbf{x}(s) \right) \right\|^2$$

is positive when the ith-node is in the obstacle and negative otherwise. In this, $\underline{r}_{e,0} \in \mathbb{R}^{\frac{1}{2}(N-1)}$ is the initial position of the elastic body and $\underline{r}_{s,0} \in \mathbb{R}^2$ is the initial position of the rigid body.

Finally, the optimal control problem reads

$$\min_{\mathbf{u}\in\mathcal{U}_\infty} \int_0^\infty \mathbf{x}(s)^\top Q\mathbf{x}(s) + \mathbf{u}(s)^\top R\mathbf{u}(s)\, \mathrm{d}s \tag{17}$$

$$\text{subject to } E\dot{\mathbf{x}}(s) = A\mathbf{x}(s) + D\mathbf{u}(s),\ \mathbf{x}(0) = x_0 \tag{18}$$

$$\text{and } C_i(\mathbf{x}(s)) \le 0 \text{ for all } s \in [0, \infty) \text{ and } i = 1, ..., n. \tag{19}$$

4 BUILDING A KERNEL SURROGATE

As presented in Section 2, we solve the approximate system (8)–(9) of (17)–(19). With the PMP, this leads to a two-point BVP, which can be solved numerically using MATLAB's *bvp5c* algorithm [15]. It discretizes the trajectory equidistantly in time and formulates a root finding problem. In addition, a non-trivial initial solution is required.

For simplicity, we approximate the value function only for states above the center of the obstacle. Furthermore, the initial state of the optimal trajectories in the data set have a constant displacement and zero initial velocity. By discretizing the area above the

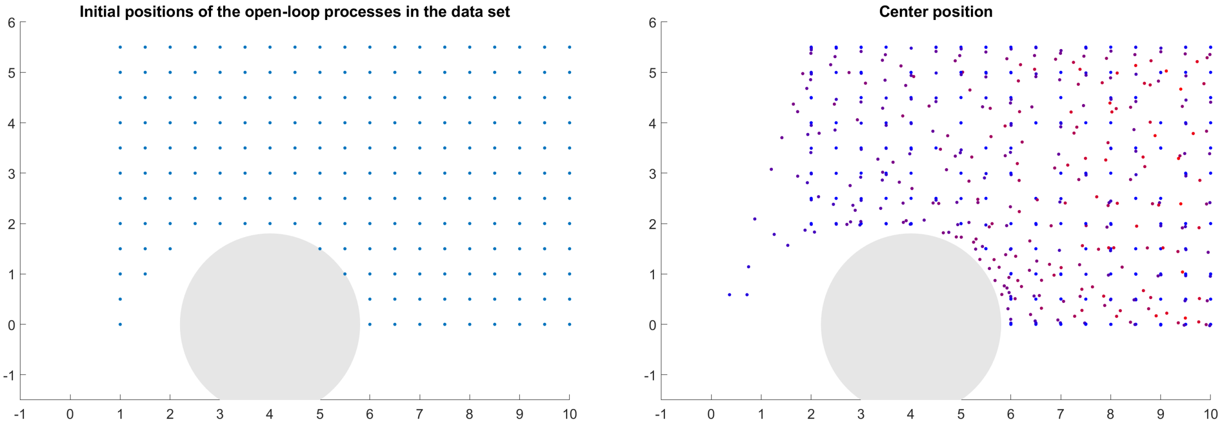


Figure 2: The left plot shows the lower left corner of the starting states that occur in the data set. The right plot shows the positions of the lower left corner of states chosen as centers by the VKOGA. The color gradient shows the velocity, where light red means high and deep blue means low velocity.

middle of the obstacle, while omitting states that reach into obstacles (see Fig. 2 left), we define starting positions, characterized by the position of the lower left corner. Using *bvp5c* we then get the corresponding optimal trajectories and obtain information about the GVF along the entire trajectory via the co-state. The resulting data set $\mathcal{D} := (\mathcal{X}, \mathcal{Y})$ therefore consists of states as inputs ($\mathcal{X}$), which belong to discretized optimal trajectories, and the corresponding values of the GVF as output ($\mathcal{Y}$).

A vector-valued kernel interpolant has the form

$$s_n(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \tag{20}$$

with $\alpha_i \in \mathbb{R}^N$ and a kernel k. A kernel is a symmetric function $k : \Omega \times \Omega \longrightarrow \mathbb{R}$ for a non-empty set Ω. In our case $\Omega = \mathbb{R}^N$. To generate an interpolant, the n-equation system

$$\mathcal{A} K_{\mathcal{X}_n} = \mathcal{Y}$$

must then be solved, where

$$\mathcal{A} := [\alpha_1, ..., \alpha_n] \in \mathbb{R}^{N \times n}, \;\; \mathcal{Y} := [\nabla v(x_1), ..., \nabla v(x_n)] \in \mathbb{R}^{N \times n} \;\text{ and }\; (K_{\mathcal{X}_n})_{i,j} := k(x_i, x_j).$$

In this, $K_{\mathcal{X}_n}$ is the Gramian matrix and $\mathcal{X}_n = \{x_1, ..., x_n\}$ is a set of pairwise distinct centers. The latter is a subset of the data input set. In these points, the interpolant equals the GVF, so the centers must be selected appropriately in order to ensure a good approximation. One way to do this is to use the so-called vectorial kernel orthogonal greedy algorithm (VKOGA) [16] (see Algo. 1) with f-greedy selection criterion. Here, $\mathcal{X}$ is a discrete input set for which the values of the target function f are known. The algorithm picks the point with the greatest deviation between the target function f and

Algorithm 1 VKOGA

```
Input: $(\mathcal{X}, f(\mathcal{X}))$, $\mathcal{X}_0 := \emptyset$, $V(\mathcal{X}_0) := \{0\}$, $s_0 := 0$, $n := 1$, $n_{\max}$, $\epsilon_{\text{tol}}$
1. While $n \leq n_{\max}$ and $\max_{x \in \mathcal{X} \setminus \mathcal{X}_{n-1}} \|f(x) - s_{n-1}(x)\|_{\mathbb{R}^N} > \epsilon_{\text{tol}}$
2.     $x_n := \arg\max_{x \in \mathcal{X} \setminus \mathcal{X}_{n-1}} \|f(x) - s_{n-1}(x)\|_{\mathbb{R}^N}$
3.     $\mathcal{X}_n := \mathcal{X}_{n-1} \cup \{x_n\}$
4.     $V(\mathcal{X}_n) := \text{span}\{k(x, \cdot) \mid x \in \mathcal{X}_n\}$
5.     $s_n := \text{interpolant}(\mathcal{X}_n, f(\mathcal{X}_n))$
Output: $s_n \approx f$
```

the interpolant. Thus the new interpolant s_n has no deviation there. The VKOGA is easy to implement and often delivers extremely good results. Theoretical results on the observed excellent convergence rates can be found in [17].

However, our surrogate for the GVF for the GST is only partially in the form of (20), as we use a multi-level scheme [18] for which we give some heuristic arguments next. First, it must be noted that many parts of the GVF are not influenced by the obstacle. These are at least all states to the left of the obstacle in the scenario of Figure 1. But even if the starting position is very far from the target position, the GVF is again almost not influenced by the obstacle, since the costs of the detour around the obstacle become lower compared to the total cost of the path. Therefore, the structure of the GVF for the problem without an obstacle should play an important role, i.e. no state constraints, which is a classic linear-quadratic regulator (LQR). For such a problem, the form of the VF is $v(x) = x^\top Kx$, where the matrix $K \in \mathbb{R}^{N \times N}$ can be computed from known quantities and the solution of the algebraic Riccati equation. Thus, the gradient is given by $\nabla v(x) = 2Kx$. Because of that, we use the linear kernel $k_{\text{lin}}(x, y) = \langle x, y \rangle$ to approximate it. Only a few open-loop trajectories are required to obtain a very precise approximation $s_{\text{NoOb}}(x)$ for this problem. The subscript "NoOb" stands for no obstacle. Using this approximation we compute a residual data set $\mathcal{D}_{\text{Res}} := (\mathcal{X}, \mathcal{Y}_{\text{Res}})$ of the problem with the obstacle by subtracting $s_{\text{NoOb}}(x)$ from the output ($\mathcal{Y}_{\text{Res}} := \mathcal{Y} - s_{\text{NoOb}}(\mathcal{X})$). The resulting data set describes the pure influence of the obstacle on the GVF. Then we use the locally operating kernel $k_{\text{linGauss}}(x, y) := \langle x, y \rangle_{\mathbb{R}^N} e^{-\gamma \|x-y\|^2}$ to approximate it, where the kernel width γ can be chosen using cross-validation. It is a Gaussian kernel multiplied by a linear kernel. The latter ensures that the surrogate in the target position is zero, which is necessary for the GST to stand still there. With this the overall surrogate for the problem with obstacle has the form

$$s_{\text{WithOb}}(x) = \sum_{i=1}^{n_{\max}} \alpha_i \langle x_i, x \rangle + \sum_{j=1}^{m_{\max}} \beta_j \langle \bar{x}_j, x \rangle_{\mathbb{R}^N} e^{-\gamma \|\bar{x}_j - x\|^2}.$$

We have chosen this two-step approach because it ensures that the linear term is created independently of the obstacle. So if we have a good, hopefully global, approximation here, it is guaranteed that many states in the residual data set are mapped to near zero, and therefore the Gaussian kernel term can concentrate only on the local part of the GVF which is influenced by the obstacle.

5 NUMERICAL EXPERIMENTS

Now we perform the numerical test. As defined in Section 2, we have in $\mathcal{K}(x; s_{\text{WithOb}}(x))$ an approximate feedback rule. Solving the ODE

$$E\dot{\mathbf{x}}(s) = A\mathbf{x}(s) + D\mathcal{K}\left(\mathbf{x}(s); s_{\text{WithOb}}\left(\mathbf{x}\left(s\right)\right)\right), \tag{21}$$

then leads to an approximate optimal trajectory. We set $T = 22$, since this is sufficiently large for everything considered below. For solving (21), we use MATLAB's *ode23* algorithm, which adaptively adjusts the width of the time steps. We therefore resample the resulting discrete solution and obtain $(\mathbf{x}_{\text{SR},i})_{i=1}^{M} \subset \mathbb{R}^N$ on the same equidistant time grid that we have for the open-loop controlled solution $(\mathbf{x}_{\text{OL},i})_{i=1}^{M} \subset \mathbb{R}^N$. To compare these, we introduce

$$L^2\text{-Error} = \sqrt{\frac{\sum_{i=1}^{M} \|\mathbf{x}_{\text{OL},i} - \mathbf{x}_{\text{SR},i}\|_{\mathbb{R}^N}^2}{\sum_{i=1}^{M} \|\mathbf{x}_{\text{OL},i}\|_{\mathbb{R}^N}^2}}.$$

For the multi-stage method, we first approximate the GVF without the obstacle. The LQR problem structure allows a linear combination of optimal trajectories to obtain an optimal trajectory for a new starting state. If we consider only initial states with constant displacement and constant velocity in space, we have four degrees of freedom. Hence, it is sufficient to compute four open-loop processes with linear independent initial states. With them we generate a set of 600 trajectories with randomly selected starting states. The VKOGA selected 28 states as centers from it, which stem from 8 trajectories. To see how good the resulting trajectories controlled by the surrogate are, we consider the starting positions

$$x_{0,1} = \begin{bmatrix} 8 & 4 & 0 & 0 \end{bmatrix}^\top, \; x_{0,2} = \begin{bmatrix} 9 & 0 & 0 & 0 \end{bmatrix}^\top \text{ and } \; x_{0,3} = \begin{bmatrix} 8 & 4 & 0 & -10 \end{bmatrix}^\top.$$

$x_{0,1}$ and $x_{0,2}$ are included in the training data set for a reproduction test, and $x_{0,3}$ is not in the training data set. Thus, we perform a generalization test with it. The results can be seen in Table 1 in the second and third column for the GST without obstacle. There we also report the execution times for the online phase, which gives in case of the open-loop controlled solution the time for solving the BVP and for the surrogate controlled solution the time for solving the ODE (21). The tests were performed on a computer with an Intel Core i7-9700K CPU and 32GB RAM. With feedback control using the surrogate it is clearly much faster, which is one reason why we are interested in such an online and offline decomposition. Additionally, we observe that the errors are very small and thus the solutions are accurate. This also applies to the generalization case in which we have

	$\underline{s}_{\text{NoOb}}(x)$		$\underline{s}_{\text{WithOb}}(x)$	
	$x_{0,1}$	$x_{0,3}$	$x_{0,1}$	$x_{0,3}$
L^2-Error	2.1520e-04	3.6803e-04	4.6898e-03	5.8210e-02
ExTimeOL	1.4424e+02 s	1.4849e+02 s	3.9195e+02 s	3.8108e+02
ExTimeSR	1.7678e-01 s	1.8362e-01 s	1.1296 s	1.1388 s

Table 1: The error between the open-loop (OL) controlled trajectory and the surrogate (SR) controlled trajectory for the starting states $x_{0,1}$ (Reproduction) and $x_{0,3}$ (Generalization) are shown for the problem without and with obstacle. There are also the associated execution times, which are an average of ten runs.

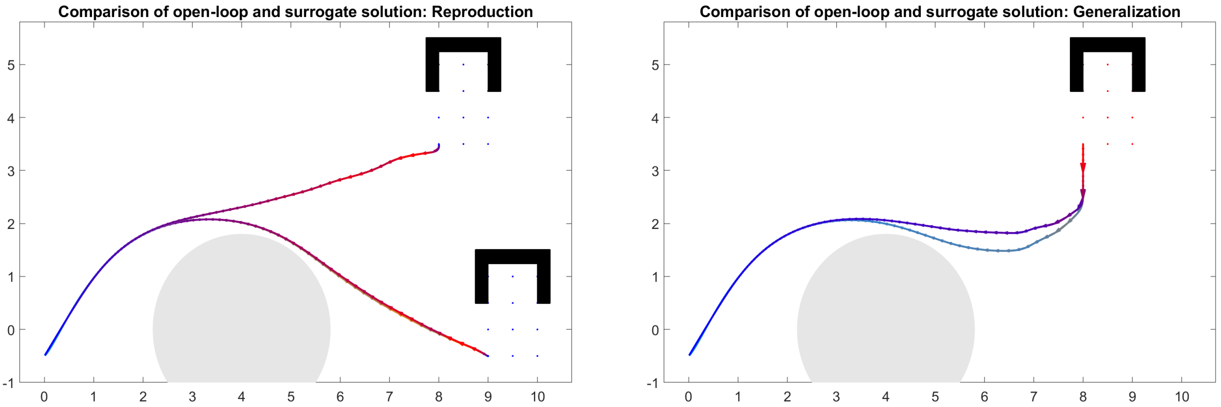
a high starting velocity. One can therefore conclude that the linear structure of $s_{\text{NoOb}}(x)$ gives very good global approximation of the feedback control.

$s_{\text{NoOb}}(x)$ provides one part of the overall surrogate. Next, we compute the residual data set and the Gaussian kernel part of $s_{\text{WithOb}}(x)$ again by kernel interpolation. For this, $m_{\max} = 566$ centers were used, which were selected by the VKOGA as shown in Figure 2 right. The centers originate from 165 trajectories and the kernel width used was $\gamma = 0.03$. In fact, the surrogate controlled solutions for the starting states $x_{0,1}$ and $x_{0,2}$ are so similar that there is no visual difference to the open-loop trajectories (see Fig. 3 left). So the reproduction works very well, which is also shown by the data in Table 1 (fourth column).

The result for the generalization experiment (see Fig. 3 right) is very comprehensible in terms of the generalization test for the problem without obstacle. The GST does not seem to bypass the obstacle at first. This is because $s_{\text{NoOb}}(x)$ initially dominates in order to slow it down. That brings it to an area where the second term of $s_{\text{WithOb}}(x)$ has an influence and thus directs it around the obstacle. Since we have very accurate optimal control via the surrogate for the GVF for states with low velocity, as the reproduction test shows, the surrogate controlled solution is quickly returning to the optimal path. The result, therefore, is a trajectory that deviates only for some time.

6 CONCLUSION

In total we managed to generate with the approach a very well functioning feedback control for the GST problem, which is high-dimensional as $N = 36$. It is real-time capable, accurate and robust as the generalization test suggests. The reason for this lies in the multi-stage structure, a proper choice of kernels and the approximation of the GVF as a whole, which according to (7) is the key to a feedback policy. As the considered dimension of 36 of the system is still decent from a finite element viewpoint, future work will consist of combining the approach with model order reduction (MOR) techniques to obtain a finer resolution of the soft tissue.



Comparison of open-loop and surrogate solution: Reproduction

Comparison of open-loop and surrogate solution: Generalization

Figure 3: The trajectories of the lower left corner of the surrogate and open-loop controlled solution for the starting states $x_{0,1}$, $x_{0,2}$ (left) and $x_{0,3}$ (right) are shown. These have a color gradient from orange (high velocity) to light blue (low velocity) for the surrogate controlled solutions and a color gradient from light red (high velocity) to deep blue (low velocity) for the open-loop controlled solutions.

7 ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support of this project by the International Research Training Group 2198 (IRTG) ”Soft Tissue Robotics”. Further, we thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding - EXC2075 – 390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

REFERENCES

[1] S. Shuva, P. Buchfink, O. Röhrle, and B. Haasdonk. Reduced basis methods for efficient simulation of a rigid robot hand interacting with soft tissue. arXiv 2103.15422, 2021.

[2] O. Junge and A. Schreiber. Dynamic programming using radial basis functions. *Discrete & Continuous Dynamical Systems*, 35:4439–4453, 2015.

[3] A. Alla, A. Schmidt, and B. Haasdonk. Model order reduction approaches for infinite horizon optimal control problems via the HJB equation. In *Model Reduction of Parametrized Systems*, pages 333–347. Springer International Publishing, Cham, 2017.

[4] A. Schmidt and B. Haasdonk. Data-driven surrogates of value functions and applications to feedback control for dynamical systems. *IFAC-PapersOnLine*, 51(2):307 – 312, 2018. 9th Vienna International Conference on Mathematical Modelling.

[5] W. Kang and L. C. Wilcox. Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Computational Optimization and Applications*, 68(2):289–315, Nov 2017.

[6] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. Adaptive deep learning for high dimensional Hamilton-Jacobi-Bellman equations. *CoRR*, abs/1907.05317, 07 2019.

[7] J. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance Control and Dynamics*, 21:193–207, 1998.

[8] D. Garg, W. W. Hager, and A. V. Rao. Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica*, 47(4):829–837, 2011.

[9] P. Malisani, F. Chaplais, and N. Petit. An interior penalty method for optimal control problems with state and input constraints of nonlinear systems. *Optimal Control Applications and Methods*, 37, 09 2014.

[10] S.P. Sethi. *Optimal Control Theory: Applications to Management Science and Economics*. Springer International Publishing, 2018.

[11] A. Jadbabaie and J. Hauser. On the stability of receding horizon control with a general terminal cost. *IEEE Transactions on Automatic Control*, 50(5):674–678, 2005.

[12] D. Carlson, A.B. Haurie, and A. Leizarowitz. *Infinite Horizon Optimal Control*. Springer-Verlag Berlin Heidelberg, 1991.

[13] E. F. Costa and J. B. R. do Val. Optimal cost convergence with respect to the time horizon. In *2003 European Control Conference (ECC)*, pages 2185–2189, 2003.

[14] R.W. Clough and J. Penzien. *Dynamics of Structures*. McGraw-Hill, 1993.

[15] J. Kierzenka and L. Shampine. A BVP solver that controls residual and error. *European Society of Computational Methods in Sciences and Engineering (ESCMSE) Journal of Numerical Analysis, Industrial and Applied Mathematics*, 3:27–41, 01 2008.

[16] D. Wirtz and B. Haasdonk. An improved vectorial kernel orthogonal greedy algorithm. *Dolomites Research Notes on Approximation*, 6:83–100, 2013.

[17] T. Wenzel, G. Santin, and B. Haasdonk. Analysis of data dependent greedy kernel algorithms: Convergence rates for f -, f·p - and f /p -greedy, 2021. In preparation.

[18] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.