

# Robust Control of Continuum Robots using Interval Arithmetic

Florian Hisch, Andrea Giusti and Matthias Althoff

*The authors are with the Department of Informatics,  
Technical University of Munich (TUM), 85748 Garching, Germany  
E-mail: [florian.hisch@tum.de](mailto:florian.hisch@tum.de), {[giusti](mailto:giusti@in.tum.de), [althoff](mailto:althoff@in.tum.de)}@in.tum.de*

**Abstract:** We consider the problem of controlling pneumatically actuated continuum robots with uncertain system dynamics and input disturbance. While such systems are intrinsically structurally safe due to soft and light-weight components, their structural flexibility challenges the control stability and performance. We present a robust tracking control approach using interval arithmetic. With this approach a user defined tracking performance can be ultimately met without the need for empirical estimation of bounds of perturbations from model uncertainty and input disturbances. We show the validity of our approach by simulating scenarios with different parametric uncertainty and by comparing the performance with an existing inverse-dynamics controller.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

**Keywords:** Robust Control, Interval Arithmetic, Continuum Robots and Soft Robots.

## 1. INTRODUCTION

Hyper-redundant, soft or continuum robots inspired by nature e.g. octopus arms (see Sfakiotakis et al. (2013)), snakes or elephant trunks (see Eder et al. (2014)) have been an active research area since the late 60's when Anderson and Horn (1967) first introduced such robotic design. Flexible robots can be used in environments where rigid robots are either not suitable in size or impose a danger to humans working nearby due to their inflexibility.

The intrinsic structural complexity of continuum robots makes their kinematics and dynamics hard to model. In the last decades, the first useful models, based on the constant curvature approach e.g. in Jones and Walker (2006) or static mechanical models e.g. in Camarillo et al. (2008) and Jung et al. (2011), were introduced. A review of methods focused on the constant curvature approach can be found in Webster and Jones (2010). Rolf and Steil (2012) presented an efficient simulation implementation, and Escande et al. (2015) showed model calibration techniques. Even if most work on modelling continuum robots mainly focused on kinematics, Godage et al. (2011) proposed a dynamic representation using modal analysis, and Falkenhahn et al. (2015a) recently introduced a dynamic model which is based on the constant curvature approach and the Euler-Lagrange formalism. Obviously, the model has to be calibrated for obtaining acceptable model-based control performance. This is a hard task considering the high number of dynamical parameters involved in a flexible robot (see e.g. Escande et al. (2015)). In practice, those parameters can only be measured up to a certain accuracy.

Early developments of controllers for such systems have been based on Proportional-Integral-Derivative (PID) control schemes, whose performance suffers due to the highly coupled nonlinear dynamics of continuum robots. Nordmann et al. (2012) acknowledge these limitations and

suggest the consideration of machine learning techniques. Among recent developments, Ansari et al. (2016) proposed a machine learning approach for inverse kinematics considering the precision positioning problem. However, machine learning techniques typically rely on extensive training phases to reach acceptable precision.

As an alternative, model-based control approaches can guarantee high performance, provable closed-loop stability and can handle the more general control problem of precision tracking of trajectories where dynamical effects are non-negligible. To guarantee stability and performance with imperfect knowledge of the dynamic parameters of the robot arm, several approaches can be found in fundamental textbooks like Chung et al. (2008). Considering continuum robots, however, no robust control strategy has been applied so far, to the best of our knowledge. Recently, Giusti and Althoff (2016) introduced the use of interval arithmetic for robust control of rigid robots. This approach allows one to guarantee in principle that any user-defined tracking performance is ultimately met. In this paper, we combine the idea of Giusti and Althoff (2016) with the dynamic modelling approach of Falkenhahn et al. (2015a) to provide for the first time a robust controller for continuum robots considering uncertain system dynamics and input disturbance. We show the superior performance of our proposed controller with respect to the previously proposed inverse-dynamics controller in Falkenhahn et al. (2014) by simulating different scenarios of uncertainty of the dynamic model parameters.

The remainder of this paper is structured as follows: In Sec. 2 we recall the preliminaries of the modelling approach we consider. The control problem we address is described in detail in Sec. 3, followed by our proposed approach in Sec. 4. Simulation results are shown and discussed in Sec. 5. Finally, we draw the conclusion in Sec. 6.

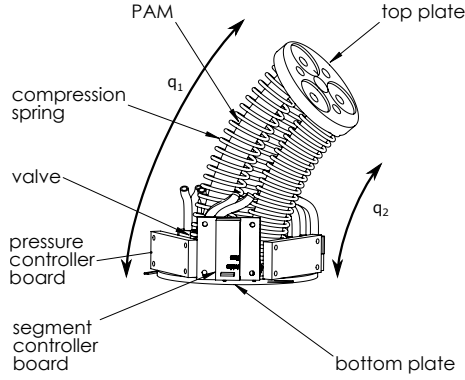


Fig. 1. Single section of a pneumatically-actuated continuum robot. Applying different pressures to the pneumatic artificial muscles (PAMs) allows the section to bend continuously.<sup>1</sup>

## 2. PRELIMINARIES: CONTINUUM ROBOT MODEL

We focus on a class of modular, pneumatic-actuated, worm-like robots, for which a representative example is Festo's Bionic Handling Assistant<sup>2</sup>. A similar robot to Festo's Bionic Handling Assistant has been developed by Eder et al. (2014) with the exception that instead of bellows, the robot consists of pneumatic artificial muscles (Festo's DMSP-10-160N). It is separated into  $n$  sections, each consisting of three pairs of a muscle and a spring. The latter is used to stabilize the section's movement. The sections can contract and bend according to the applied pressures. Figure 1 shows a single section of the robot. In the following we briefly recall the considered kinematic and dynamic modelling and introduce the notation.

A common approach for modelling the kinematics of such robots is the constant curvature approach (see Jones and Walker (2006)). Hereby, the section bending is described as a single arc of a certain radius, i.e. a certain curvature, which depends on the generalized positions  $\mathbf{q} \in \mathbb{R}^{3n}$ . As presented in Falkenhahn et al. (2015a), the kinematics can finally be described as

$${}^w\mathbf{H}_{ih} = {}^w\mathbf{H}_{1h} \prod_{x=1}^i {}^{xb}\mathbf{H}_{xh}(q_{x1}, q_{x2}, q_{x3}), \quad (1)$$

where  ${}^{xb}\mathbf{H}_{xh}$  represents the homogeneous, non-linear transformation from the base-frame  $xb$  of section  $x$  to its head-frame  $xh$ ,  ${}^w\mathbf{H}_{1h}$  is the constant world-origin transformation and  $q_{x1}$  is the length of the first of the three muscles of section  $x$ .  $q_{x2}$  and  $q_{x3}$  are the lengths of the second and third muscle, respectively.

Falkenhahn et al. (2015a) use the above kinematics and the Euler-Lagrange formalism to derive the dynamic model. The system dynamics can thus be modelled as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}(\mathbf{q}, \mathbf{p}), \quad (2)$$

with a positive-definite mass matrix  $\mathbf{M}(\mathbf{q})$ , the Coriolis matrix  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ , the homogeneous force vector  $\mathbf{N}(\mathbf{q})$  and the actuator force vector  $\boldsymbol{\tau}(\mathbf{q}, \mathbf{p})$ , which depends on the applied pressure  $\mathbf{p}$  along the generalised coordinates. It should be noted that the actuator force vector also depends

on the muscle lengths  $\mathbf{q}$ . The inversion of  $\boldsymbol{\tau}$  is shown by Falkenhahn et al. (2015b).

We consider the reduced model of Falkenhahn et al. (2015a), which is justified since rotational forces are rather small due to the limited velocity of the robot. Assuming that the mass is concentrated in the center of the section's head (i.e. at position  $\mathbf{t} = [0 \ 0 \ 0 \ 1]^T$ ), the mass matrix and Coriolis matrix entries are obtained as

$$M_{ik,\alpha\beta} = \sum_{x=\max(i,\alpha)}^n m_x \left( \frac{\partial {}^w\mathbf{H}_{xh}}{\partial q_{\alpha\beta}} \mathbf{t} \right)^T \frac{\partial {}^w\mathbf{H}_{xh}}{\partial q_{ik}} \mathbf{t}, \quad (3)$$

$$C_{ik,\alpha\beta} = \sum_{x=\max(i,\alpha)}^n m_x \left( \frac{\partial {}^w\dot{\mathbf{H}}_{xh}}{\partial q_{\alpha\beta}} \mathbf{t} \right)^T \frac{\partial {}^w\mathbf{H}_{xh}}{\partial q_{ik}} \mathbf{t}, \quad (4)$$

where the matrix column and row indexing is separated into section index  $(i, \alpha)$  and muscle index  $(k, \beta)$ , with  $3(i-1)+k$  as the row index and  $3(\alpha-1)+\beta$  as the column index. For example, the matrix element  $M_{12,23}$  describes the entry with row 2 and column 6. The  $\mathbf{M}$  and  $\mathbf{C}$  matrix defined as in (3) and (4) allows the matrix  $(\dot{\mathbf{M}} - 2\mathbf{C})$  to be skew-symmetric. We show a detailed proof in App. A.

The passive forces of gravity  $F_g$ , the spring forces  $F_{\text{spr}}$  and damping forces  $F_{\text{dmp}}$  are included in  $\mathbf{N}(\mathbf{q})$ . In addition, Falkenhahn et al. (2015a) also consider a bending momentum and external forces. The bending momentum describes how a system would resurrect itself if no pressure is applied, i.e. how strong the section tries to get into its resting position. A strong coupling between continuous actuators would increase the effect of the bending momentum. Since the muscles are only connected at the base and the head frame, we assume the influence of the bending momentum to be negligible and also no further external forces. We use a linear model for the springs with spring constants  $K_{ik}$ , resting positions  $L_{\text{spr}}$  and current spring / muscle lengths  $q_{ik}$ . In addition, we consider a linear damping model with coefficients  $D_{ik}$ :

$$F_{\text{spr},ik} = K_{ik} (q_{ik} - L_{\text{spr}}), \quad (5)$$

$$F_{\text{dmp},ik} = D_{ik} \dot{q}_{ik}. \quad (6)$$

The actuator forces  $F_{\text{act}}$  are combined in vector  $\boldsymbol{\tau}(\mathbf{q}, \mathbf{p})$ . The actuator force function is typically nonlinear and depends on the pressure and on the current contraction  $L_{\text{act}} - q_{ik}$ , where  $L_{\text{act}}$  is the resting length of the muscles. We redirect the reader to Falkenhahn et al. (2015a) for a detailed description of  $\mathbf{N}(\mathbf{q})$  and  $\boldsymbol{\tau}(\mathbf{q}, \mathbf{p})$ .

## 3. PROBLEM DESCRIPTION

Dynamic models match the real robot dynamics only up to a certain degree. Obviously, model reductions and non-measurable properties reduce the accuracy. We consider uncertainty in the dynamic parameters of the robot model in (2), which we collect in a vector  $\Delta$ :

$$\Delta = [\Delta_1, \Delta_2, \dots, \Delta_n]^T,$$

$$\Delta_i = [m_i, K_{i1}, K_{i2}, K_{i3}, D_{i1}, D_{i2}, D_{i3}]^T, \quad 1 \leq i \leq n,$$

where  $\Delta_i$  denotes the parameters of the  $i$ -th section. Additionally, we assume that a nominal vector  $\Delta_0$  is

<sup>1</sup> Thanks to Martin Eder for providing the image.

<sup>2</sup> <https://www.festo.com/group/en/cms/10241.htm>, last visited: 03.04.2017

available and that all uncertain parameters are bounded with a known lower  $\underline{\Delta}$  and upper bound  $\overline{\Delta}$ . We further consider the input disturbance  $\mathbf{d}$ , which is bounded by  $\beta_d$ .

For the above-mentioned model and assumptions, we address the problem of designing a controller that guarantees to ultimately (for a finite  $t_1 \geq 0$ ) track a trajectory  $\mathbf{q}_d$ , which is at least twice differentiable, with a user defined precision  $\epsilon$ :

$$\|\mathbf{q}_d - \mathbf{q}\| < \epsilon, \quad \forall t \geq t_1. \quad (7)$$

#### 4. PROPOSED METHOD

By considering the control approach presented by Giusti and Althoff (2016), we obtain the following control command:

$$\mathbf{p} = \boldsymbol{\tau}^{-1}(\mathbf{q}, \mathbf{M}(\mathbf{q}, \Delta_0) \ddot{\mathbf{q}}_a + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \Delta_0) \dot{\mathbf{q}}_a + \mathbf{D}(\mathbf{q}, \dot{\mathbf{q}}, \Delta_0) - \mathbf{v}), \quad (8)$$

where  $\boldsymbol{\tau}^{-1}$  is the inverse function of the homogeneous force vector,  $\mathbf{v}$  is a function which ensures robustness and the error control terms are

$$\dot{\mathbf{q}}_a = \dot{\mathbf{q}}_d + \mathbf{K}_r \tilde{\mathbf{q}}, \quad \tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}, \quad \mathbf{r} = \dot{\tilde{\mathbf{q}}} + \mathbf{K}_r \tilde{\mathbf{q}}, \quad (9)$$

with  $\mathbf{K}_r$  being a diagonal positive-definite tuning matrix.

It is not difficult to show that by applying the control law (8), a closed-loop perturbation arises from the difference between the nominal and real robot model, which can be expressed as

$$\begin{aligned} \mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a, \mathbf{d}, \Delta, \Delta_0) &= \underbrace{(\mathbf{M}(\mathbf{q}, \Delta) - \mathbf{M}(\mathbf{q}, \Delta_0))}_{\tilde{\mathbf{M}}} \ddot{\mathbf{q}}_a \\ &+ \underbrace{(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \Delta) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \Delta_0))}_{\tilde{\mathbf{C}}} \dot{\mathbf{q}}_a \\ &+ \underbrace{(\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}, \Delta) - \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}, \Delta_0))}_{\tilde{\mathbf{N}}} \\ &- \mathbf{d}. \end{aligned} \quad (10)$$

We consider hereafter multidimensional intervals. A multidimensional interval  $[\mathbf{x}]$  can be defined as

$$[\mathbf{x}] = [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \quad \underline{\mathbf{x}} \in \mathbb{R}^n, \overline{\mathbf{x}} \in \mathbb{R}^n, \forall i \in \{1, \dots, n\} : x_i \leq \overline{x}_i,$$

where  $\underline{\mathbf{x}}$  and  $\overline{\mathbf{x}}$  denote the infimum and supremum of the interval  $[\mathbf{x}]$ , respectively. We consider the following basic operations on intervals  $[x] = [\underline{x}, \overline{x}]$ ,  $[y] = [\underline{y}, \overline{y}]$ ,  $[z] = [\underline{z}, \overline{z}]$  and a scalar  $c \in \mathbb{R}$ :

$$[z] = [x] + [y] \Rightarrow [z, \overline{z}] = [\underline{x}, \overline{x}] + [\underline{y}, \overline{y}], \quad (11)$$

$$[z] = [x] - [y] \Rightarrow [z, \overline{z}] = [\underline{x}, \overline{x}] - [\underline{y}, \overline{y}], \quad (12)$$

$$[z] = [x] \cdot c \Rightarrow [z, \overline{z}] = \begin{cases} [\underline{x}, \overline{x}] \cdot c, & c \geq 0 \\ [\overline{x}, \underline{x}] \cdot c, & c < 0 \end{cases}. \quad (13)$$

Any specific disturbance  $\mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a, \mathbf{d}, \Delta, \Delta_0)$  is from a multidimensional interval  $[\Phi]$ , which is originating from the uncertain dynamic parameters  $\Delta \in [\Delta] = [\underline{\Delta}, \overline{\Delta}]$  and  $\mathbf{d} \in [\beta_d] = [-\beta_d, \beta_d]$ :

$$[\Phi] = \mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a, [\beta_d], [\Delta], \Delta_0). \quad (14)$$

The absolute maximal (worst-case) disturbance can be simply derived from the element-wise absolute maximum of the lower and upper bounds of  $[\Phi]$ :

$$\rho([\Phi]) = \max(|\Phi|, |\overline{\Phi}|). \quad (15)$$

The on-line evaluation of (14) and (15) can be done with an interval arithmetic toolbox, e.g. commercial Intlab by Rump (1999) or freely available CORA by Althoff (2015). Directly using (14) with those toolboxes would result in the following expression:

$$[\Phi] = [\tilde{\mathbf{M}}] \ddot{\mathbf{q}}_a + [\tilde{\mathbf{C}}] \dot{\mathbf{q}}_a + [\tilde{\mathbf{N}}] - [\beta_d], \quad (16)$$

where each matrix and vector element of  $\tilde{\mathbf{M}}$ ,  $\tilde{\mathbf{C}}$  and  $\tilde{\mathbf{N}}$  is an independent interval term itself. Thereby, the matrices and vectors lose their underlying dynamic context, e.g. that each  $3 \times 3$  submatrix of  $\mathbf{M}$  and  $\mathbf{C}$  is influenced by the same uncertain mass. This dependency problem results in generous lower and upper bounds of  $[\Phi]$ .

We reduce the reoccurrence of interval variables (and thus conservativeness) by including the knowledge of the physical system in our interval arithmetic calculation of  $[\Phi]$ . At first, we observe that for (15) only the element-wise absolute maximum is necessary, i.e. we can select our upper and lower bounds independently for every row of  $[\Phi]$  and thus also for  $\tilde{\mathbf{M}}$ ,  $\tilde{\mathbf{C}}$  and  $\tilde{\mathbf{N}}$ . Instead of using (16), we directly include the accelerations  $\ddot{\mathbf{q}}_a$  and the velocities  $\dot{\mathbf{q}}_a$  in the interval calculation of the mass and Coriolis matrix. The row elements of  $[\Phi]$  can thus be calculated using

$$[\Phi_{ik}] = \underbrace{[\tilde{\mathbf{M}}_{ik} \ddot{\mathbf{q}}_a]}_{\textcircled{1}} + \underbrace{[\tilde{\mathbf{C}}_{ik} \dot{\mathbf{q}}_a]}_{\textcircled{2}} + [\tilde{\mathbf{N}}_{ik}] - [\beta_{d_{ik}}]. \quad (17)$$

Now, we show the exemplary derivation of the first interval term on the right-hand side of (17) consisting of mass and acceleration (①). The other interval expressions (②) are omitted in this paper for brevity since the same approach of the first interval term can be used and their derivation is therefore straightforward. At first, we describe the deviation  $\tilde{\mathbf{M}}$  between the elements of the real mass matrix and their nominal counterparts as

$$\begin{aligned} \tilde{M}_{ik, \alpha\beta} &= M_{ik, \alpha\beta}(\mathbf{q}, \Delta) - M_{ik, \alpha\beta}(\Delta_0) \\ &= \sum_{x=\max(i, \alpha)}^n (m_x - m_0) \left( \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{\alpha\beta}} \mathbf{t} \right)^T \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{ik}} \mathbf{t} \end{aligned} \quad (18)$$

Further, we reshape the matrix vector product  $\tilde{M}_{ik} \ddot{\mathbf{q}}_a$  so that first we sum inside each section and finally all sections

$$\begin{aligned} [\tilde{\mathbf{M}}_{ik} \ddot{\mathbf{q}}_a] &= \sum_{c=1}^{3n} [\tilde{M}_{ik, c} \ddot{q}_{a_c}] \\ &= \sum_{\alpha=1}^n \sum_{\beta=1}^3 [\tilde{M}_{ik, 3(\alpha-1)+\beta} \ddot{q}_{a_{3(\alpha-1)+\beta}}] \\ &= \sum_{\alpha=1}^n \sum_{x=\max(i, \alpha)}^n \underbrace{[(m_x - m_{x0})] \mathcal{K}}_{[\tilde{m}_x]} \end{aligned} \quad (19)$$

with

$$\mathcal{K} = \left( \sum_{\beta=1}^3 \left( \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{\alpha\beta}} \mathbf{t} \right)^T \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{ik}} \mathbf{t} \cdot \ddot{q}_{a_{\alpha\beta}} \right). \quad (20)$$

Finally, we can calculate the lower  $\underline{\tilde{\mathbf{M}}_{ik} \ddot{\mathbf{q}}_a}$  and upper  $\overline{\tilde{\mathbf{M}}_{ik} \ddot{\mathbf{q}}_a}$  limits with interval arithmetic. Alg. 1 implements (19), using in particular (13), as can be seen from lines 5 to 9 of the algorithm.

**Algorithm 1.** Calculation of  $[\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a]$

```

1:  $[\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a] \leftarrow [0, 0]$ 
2: for  $\alpha = 1 \rightarrow n$  do
3:   for  $x = \max(i, \alpha) \rightarrow n$  do
4:      $\mathcal{K} \leftarrow \sum_{\beta=1}^3 \left( \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{\alpha\beta}} \mathbf{t} \right)^T \frac{\partial^w \mathbf{H}_{xh}}{\partial q_{ik}} \mathbf{t} \ddot{q}_{a\alpha\beta}$ 
5:     if  $\mathcal{K} < 0$  then
6:        $[\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a] \leftarrow [\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a] + [\tilde{m}_x, \tilde{m}_x] \cdot \mathcal{K}$ 
7:     else
8:        $[\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a] \leftarrow [\tilde{\mathbf{M}}_{ik}\ddot{\mathbf{q}}_a] + [\tilde{m}_x, \tilde{m}_x] \cdot \mathcal{K}$ 
9:     end if
10:   end for
11: end for
    
```

With the above mentioned approach we obtain the maximal disturbance (15) and use it to compute the feedback control term  $\mathbf{v}$  in (8), that enhances robustness:

$$\mathbf{v} = -(\kappa(t)\|\rho([\Phi])\| + \phi(t))\mathbf{r}, \quad (21)$$

where  $\kappa(t)$  and  $\phi(t)$  are positive increasing functions. Further detail on the selection of these functions can be found in Giusti and Althoff (2016).

## 5. SIMULATION RESULTS

In this section, we describe our simulation testbed, present the results for the proposed control technique and finally compare them to a previously proposed model-based, inverse-dynamics control method by Falkenhahn et al. (2014). The simulations have been performed using MATLAB and Simulink R2015b with a real-time target machine<sup>3</sup>. We consider a robot composed of two sections, sampling rate<sup>4</sup> of 1 ms and the following (nominal) dynamic parameters:  $m_1 = 1.5 \text{ kg}$ ,  $m_2 = 0.5 \text{ kg}$ ,  $K_{ik} = 1900$ ,  $D_{ik} = 1000$ ,  $\forall i \in \{1, 2\}, \forall k \in \{1, 2, 3\}$ . Following the assumptions stated in Sec. 3, we consider that the mass, the three spring constants and the three damping constants for each section are uncertain. For all these parameters we assume a variability of up to  $\pm 10\%$ .

We consider first an ideal scenario (scenario 1) where no uncertainty, disturbance and correct initial conditions are assumed. Subsequently, we consider multiple cases where we deal with varying uncertainty and unmatched initial conditions. All scenarios share the same desired trajectory (shown in Fig. 2):

$q_{d_{ik}}(t) = -a_i \cdot \sin(f_{ik} \cdot t)^4 + L_i, 1 \leq i \leq 2, 1 \leq k \leq 3$  (22)  
with amplitudes  $a = [0.01, 0.015]$  in meter, frequencies  $b = \pi \cdot [0.2, 0.21, -0.22, 0.25, 0.3, -0.3]$  and a constant offset  $L_i = [0.15, 0.145]$ . The initial condition of its derivatives is  $\dot{\mathbf{q}}_{d_{ik}}(0) = \ddot{\mathbf{q}}_{d_{ik}}(0) = 0$ . The target accuracy is  $\epsilon = 0.001 \text{ m}$ . The functions  $\kappa(t)$  and  $\phi(t)$  and the tuning parameter have been selected as:

$$\begin{aligned} \mathbf{K}_r &= 100 \mathbf{I}^{3n \times 3n}, \\ \phi(t) &= 1 + 500 \int_0^t f(\tau) d\tau, \\ \kappa(t) &= 2 + 1 \int_0^t f(\tau) d\tau, \end{aligned} \quad (23)$$

where  $\mathbf{K}_r$  is the positive definite tuning matrix and

$$f(\tau) = \begin{cases} 0, & \text{for } \|\mathbf{q}(\tau) - \mathbf{q}_d(\tau)\| < \epsilon \\ \|\mathbf{q}(\tau) - \mathbf{q}_d(\tau)\|, & \text{otherwise.} \end{cases} \quad (24)$$

For the first scenario (see Fig. 3) we consider exact knowledge of the dynamic parameters and no input disturbance. After tuning our interval-based controller (black) and the model-based controller (grey) by Falkenhahn et al. (2014) show similar behaviour. However, it should be noted that both controllers cannot reach exact tracking since we prevent the system from becoming singular by always adding a small offset to the reading of one of the muscle lengths to avoid singularity problems. The singularity is a side effect of a special case of the kinematics: if the curvature of at least one section  $i$  is zero, i.e. all lengths  $\mathbf{q}_{i1} = \mathbf{q}_{i2} = \mathbf{q}_{i3}$  are equal and the kinematics are replaced by a constant homogeneous transformation. This would imply that in this special case and at all following time steps, the section is only able to contract but not bend. Calculating the mass matrix (3) using the replacement for the homogeneous transformation reveals another problem: The matrix would not have full rank and is thus not positive-definite, not invertible and not usable for the dynamic simulations. The same problem holds for the Coriolis matrix, even though it is more acceptable since it does not need to be inverted. A practical solution to this problem is to always ensure that the robot cannot enter this state. Considering measurement accuracies and numeric issues, the singularity is unlikely to occur in a real system. For simulation, we check for the singular case and add a value in the range of the expected measurement accuracy if necessary.

In the second scenario we consider uncertainty, constant disturbance on the input force (along the general coordinates  $\mathbf{q}$ ) and wrong initial conditions. Fig. 4 shows four cases with rising uncertainty, i.e. we pick each dynamical parameter randomly from an interval of  $\pm 2\%, \pm 4\%, \pm 8\%, \pm 10\%$  around the real parameter value. The additive constant disturbance on the pressure  $\mathbf{p}$  is selected randomly as well as from an interval of  $\pm 1 \text{ bar}, \pm 2 \text{ bar}, \pm 4 \text{ bar}, \pm 5 \text{ bar}$ . Like in the first scenario, we compare our interval-arithmetic-based robust controller (black) with the inverse-dynamics controller (grey) from Falkenhahn et al. (2014). Fig. 4 (a) shows the error norm  $\|\mathbf{q}(t) - \mathbf{q}_d(t)\|$  of both controllers and the norm of the pressure  $\|\mathbf{p}\|$ . The simulation results show that the inverse-dynamics controller by Falkenhahn et al. (2014) is already not meeting the desired tracking performance bound of 1 mm (dashed line) for a 2% of dynamical parameters mismatch. For this reason, we have not included the results of this controller in the plots with larger uncertainty. On the other hand, our proposed controller allows the user-defined tracking performance to be met for all considered cases without manually retuning it, despite the wrong initial conditions and input disturbance. Additionally, as it is a nontrivial task for robust controllers in general, the upper plots in Fig. 4 (a) - (d) show a smooth development over time for the pressure commands.

<sup>3</sup> Intel Core i7-3770K (3.5 GHz), 4096 MB RAM

<sup>4</sup> max. calculation time of the proposed controller: 0.56 ms

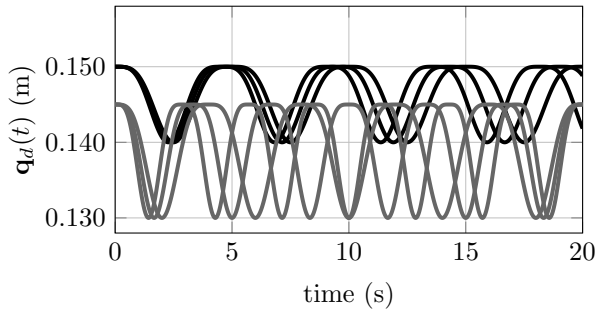


Fig. 2. Test trajectory. The three trajectories of the first section are shown in black and those of the second section in light grey.

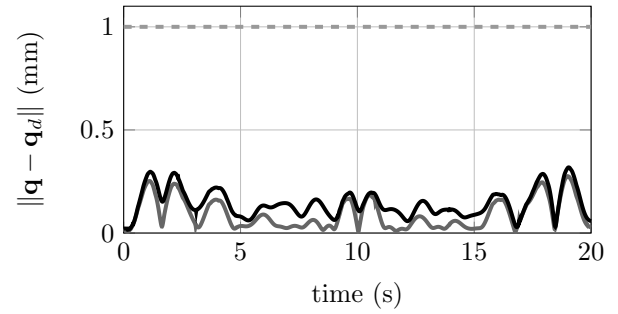


Fig. 3. First scenario without any uncertainty. The error norm  $\|\mathbf{q}(t) - \mathbf{q}_d(t)\|$  of both, interval-based (black) and model-based (gray), controllers are below the desired accuracy of 1 mm (dashed line).

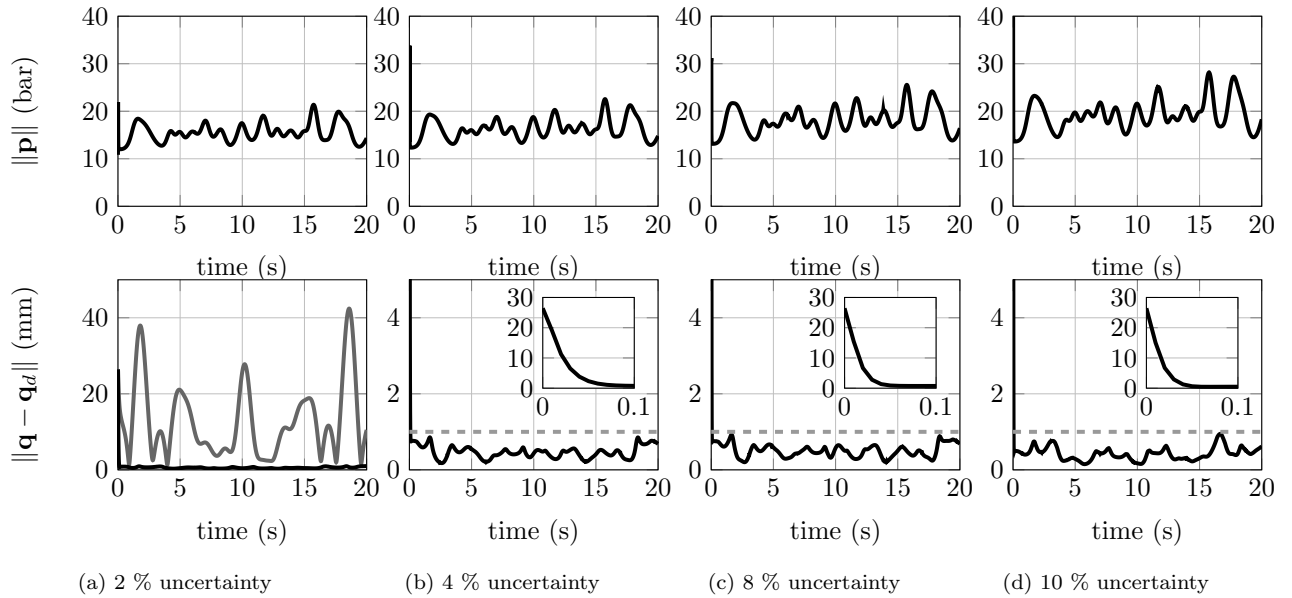


Fig. 4. Second scenario with varying uncertainty. The upper plots show the norm of pressure inputs  $\|\mathbf{p}\|$ . The lower plots contain the error norm  $\|\mathbf{q}(t) - \mathbf{q}_d(t)\|$  of the interval-based controller. In plots (b) - (d), the plot is enlarged for the first 0.1 seconds in the upper right corners. For comparison, the error norm of the model-based controller (gray) is included in (a).

## 6. CONCLUSION

We presented for the first time the implementation of an interval-arithmetic-based robust controller for continuum robots to handle imperfect knowledge of the dynamic parameters, e.g. masses or spring constants. Thus, an exact recalibration after small changes, such as changing the payload, is not required.

The comparison between our proposed interval-arithmetic-based robust controller and a model-based controller with varying amount of uncertainty shows that parametric uncertainty has a strong effect on the performance. The simple model-based controller has shown to be very sensitive even to small deviations from the exact dynamical parameters. On the other hand, our proposed robust controller shows a promising behaviour in our simulations. This encourages us to implement our proposed approach on a real continuum robot in the future.

## ACKNOWLEDGEMENTS

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007-2013/ under REA grant agreement number 608022 and the European Commission project UnCoVer-CPS under grant number 643921.

## REFERENCES

- Althoff, M. (2015). An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 120–151. URL <http://www6.in.tum.de/Main/SoftwareCORA>.
- Anderson, V. and Horn, R. (1967). Tensor arm manipulator design. *ASME Transactions*, 67-DE-57, 1–12.
- Ansari, Y., Falotico, E., Mollard, Y., Busch, B., Cianchetti, M., and Laschi, C. (2016). A multiagent reinforcement learning approach for inverse kinematics of high dimensional manipulators with precision posi-

- tioning. In *Proc. of the IEEE Int. Conf. on Biomedical Robotics and Biomechanics*, 457–463.
- Camarillo, D.B., Milne, C.F., Carlson, C.R., Zinn, M.R., and Salisbury, J.K. (2008). Mechanics modeling of tendon-driven continuum manipulators. *IEEE Transactions on Robotics*, 24(6), 1262–1273.
- Chung, W., Fu, L.C., and Hsu, S.H. (2008). *Motion Control*, 133–159. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Eder, M., Karl, M., Knoll, A., and Riesner, S. (2014). Continuum worm-like robotic mechanism with decentral control architecture. In *Proc. of the IEEE Int. Conf. on Automation Science and Engineering*, 866–871.
- Escande, C., Chettibi, T., Merzouki, R., Coelen, V., and Pathak, P.M. (2015). Kinematic calibration of a multi-section bionic manipulator. *IEEE/ASME Transactions on Mechatronics*, 20(2), 663–674.
- Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2014). Dynamic modeling of constant curvature continuum robots using the Euler-Lagrange formalism. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2428–2433.
- Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2015a). Dynamic modeling of bellows-actuated continuum robots using the Euler Lagrange formalism. *IEEE Transactions on Robotics*, 31(6), 1483–1496.
- Falkenhahn, V., Hildebrandt, A., Neumann, R., and Sawodny, O. (2015b). Model-based feedforward position control of constant curvature continuum robots using feedback linearization. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 762–767.
- Giusti, A. and Althoff, M. (2016). Ultimate robust performance control of rigid robot manipulators using interval arithmetic. In *Proc. of the American Control Conference*, 2995–3001.
- Godage, I.S., Branson, D.T., Guglielmino, E., Medrano-Cerda, G.A., and Caldwell, D.G. (2011). Dynamics for biomimetic continuum arms: A modal approach. In *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, 104–109.
- Jones, B.A. and Walker, I.D. (2006). Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1), 43–55.
- Jung, J., Penning, R.S., Ferrier, N.J., and Zinn, M.R. (2011). A modeling approach for continuum robotic manipulators: Effects of nonlinear internal device friction. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 5139–5146.
- Nordmann, A., Rolf, M., and Wrede, S. (2012). *Software Abstractions for Simulation and Control of a Continuum Robot*, 113–124. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Rolf, M. and Steil, J.J. (2012). Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3440–3446.
- Rump, S. (1999). INTLAB - INTerval LABoratory. In T. Csendes (ed.), *Developments in Reliable Computing*, 77–104. Kluwer Academic Publishers, Dordrecht. <http://www.ti3.tuhh.de/rump/>.

- Sfakiotakis, M., Kazakidi, A., Pateromichelakis, N., and Tsakiris, D.P. (2013). Octopus-inspired eight-arm robotic swimming by sculling movements. In *Proc. of the IEEE International Conference on Robotics and Automation*, 5155–5161.
- Webster, III, R.J. and Jones, B.A. (2010). Design and kinematic modeling of constant curvature continuum robots: A review. *Int. J. Rob. Res.*, 29(13), 1661–1683.

## Appendix A. PROOF OF SKEW-SYMMETRY

Given that the matrix  $\mathbf{M}$  is defined as in (3) and the matrix  $\mathbf{C}$  is defined as in (4), the matrix  $(\dot{\mathbf{M}} - 2\mathbf{C})$  is skew-symmetric.

**Proof.** We show that  $\dot{\mathbf{M}} - 2\mathbf{C}$  is skew-symmetric by showing that  $(\dot{\mathbf{M}} - 2\mathbf{C})^T = -(\dot{\mathbf{M}} - 2\mathbf{C})$ . For this purpose, we consider the following abbreviations:

$$\frac{\partial^w \mathbf{H}_{xh}}{\partial q_{ik}} = \mathcal{H}_{ik}, \quad \frac{\partial^w \dot{\mathbf{H}}_{xh}}{\partial q_{ik}} = \dot{\mathcal{H}}_{ik}, \quad \sum_{x=\max(i,\alpha)}^n \equiv \sum.$$

First, we have to determine the time derivative of the mass matrix:

$$\begin{aligned} \dot{M}_{ik,\alpha\beta} &= \frac{d}{dt} M_{ik,\alpha\beta} \\ &= \sum m_x \frac{d}{dt} \left( (\mathcal{H}_{\alpha\beta}^T \mathbf{t})^T \mathcal{H}_{ik} \mathbf{t} \right) \\ &= \sum m_x \frac{d}{dt} \left( \mathbf{t}^T \mathcal{H}_{\alpha\beta}^T \mathcal{H}_{ik} \mathbf{t} \right) \\ &= \sum m_x \mathbf{t}^T \left( \left( \frac{d}{dt} \mathcal{H}_{\alpha\beta}^T \mathcal{H}_{ik} \right) + \mathcal{H}_{\alpha\beta}^T \left( \frac{d}{dt} \mathcal{H}_{ik} \right) \right) \mathbf{t} \\ &= \sum m_x \mathbf{t}^T \left( \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \mathcal{H}_{\alpha\beta}^T \dot{\mathcal{H}}_{ik} \right) \mathbf{t}. \end{aligned}$$

The elements of the matrix  $(\dot{\mathbf{M}} - 2\mathbf{C})$  can be expressed as:

$$\begin{aligned} (\dot{\mathbf{M}} - 2\mathbf{C})_{ik,\alpha\beta} &= \sum m_x \mathbf{t}^T \left( \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \mathcal{H}_{\alpha\beta}^T \dot{\mathcal{H}}_{ik} \right) \mathbf{t} \\ &\quad - 2 \sum m_x \left( \left( \dot{\mathcal{H}}_{\alpha\beta} \mathbf{t} \right)^T \mathcal{H}_{ik} \mathbf{t} \right) \\ &= \sum m_x \mathbf{t}^T \left( \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \mathcal{H}_{\alpha\beta}^T \dot{\mathcal{H}}_{ik} \right) \mathbf{t} \\ &\quad - 2 \sum m_x \mathbf{t}^T \left( \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} \right) \mathbf{t} \\ &= \sum m_x \mathbf{t}^T \left( -\dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \mathcal{H}_{\alpha\beta}^T \dot{\mathcal{H}}_{ik} \right) \mathbf{t}. \end{aligned}$$

Finally, we conclude the proof as follows:

$$\begin{aligned} -(\dot{\mathbf{M}} - 2\mathbf{C})_{ik,\alpha\beta} &= \sum m_x \mathbf{t}^T \left( \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} - \mathcal{H}_{\alpha\beta}^T \dot{\mathcal{H}}_{ik} \right) \mathbf{t} \\ &= \sum m_x \mathbf{t}^T \underbrace{\left( -\dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} \right)}_{\in \mathbb{R}} \mathbf{t} \\ &= \sum m_x \left( \mathbf{t}^T \left( -\dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} \right) \mathbf{t} \right)^T \\ &= \sum m_x \mathbf{t}^T \left( -\dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} + \dot{\mathcal{H}}_{\alpha\beta}^T \mathcal{H}_{ik} \right)^T \mathbf{t} \\ &= \sum m_x \mathbf{t}^T \left( -\dot{\mathcal{H}}_{ik}^T \mathcal{H}_{\alpha\beta} + \mathcal{H}_{ik}^T \dot{\mathcal{H}}_{\alpha\beta} \right) \mathbf{t} \\ &= (\dot{\mathbf{M}} - 2\mathbf{C})_{\alpha\beta,ik}. \end{aligned}$$