Algorithm for finding optimal gait patterns for a gecko-inspired soft robot

Lars Schiller and Arthur Seibel

***Abstract*— This paper presents an algorithm that can predict the pose of a robot consisting of soft bending actuators to a given reference input. This algorithm is then used to find optimal gait patterns for the straight gait and for a curve. The existing gait pattern for the straight gait could be improved by a factor of 1.5 and a new gait pattern for the rotation on the spot could be discovered.**

I. INTRODUCTION

Due to the fact that the behaviour of soft materials is difficult to predict with conventional methods, an algorithm based on a geometric optimization problem is presented. The algorithm can be used to predict the actual pose of a soft robot for a given reference input. Figure 1 is taken as an example. The initial position is shown in black. The individual limbs of the robot have a certain bending angle and all feet are fixed. Now the torso of the robot should be actuated. If only the bending angle of the torso is changed, the grey dashed pose is obtained. Obviously, the two rear feet are no longer in the same position. Since these feet are fixed, the robot will behave differently in reality. In fact, it's much more likely to take up the grey pose. Although the bending angles of all limbs have changed, the condition that all feet remain motionless has been fulfilled.

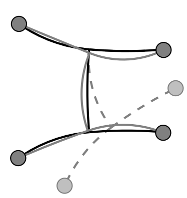

Fig. 1. Example Usage

II. PREDICTING THE POSE FOR A GIVEN REFERENCE INPUT

In order to let the robot take a pose the user has nine degrees of freedom at his disposal: the correspondending pressures for the five bending angles α_i of the limbs $i = 1, \dots, 5$ and the state of the fixation actuators $f_j \in \{0, 1\}$ of the feet $j = 1, \dots, 4$. For the unloaded state, a calibration function can be formulated for each limb, which maps the input pressure on the bending angle (under load, this function no longer needs to be valid). But the input pressure can be seen as an reference for the bending angle. Accordingly, a reference input $\boldsymbol{r}$ can be described by

$$\boldsymbol{\alpha}_{\text{ref}} = \left[\; \alpha_{\text{ref},1} \;\; \alpha_{\text{ref},2} \;\; \alpha_{\text{ref},3} \;\; \alpha_{\text{ref},4} \;\; \alpha_{\text{ref},5} \; \right]^\top \tag{1}$$

$$\boldsymbol{f} = \left[\; f_1 \;\; f_2 \;\; f_3 \;\; f_4 \; \right]^\top \tag{2}$$

$$\boldsymbol{r} = \left[\; \boldsymbol{\alpha}_{\text{ref}}^\top \;\; \boldsymbol{f}^\top \; \right]. \tag{3}$$

However, this information is not sufficient to describe the robot's actual pose. Experiments have shown that the bending angle of a limb can vary significantly at the same pressure level due to the softness of the used material. The length of a limb can also differ. In order to describe the pose of the robot, the actual bending angles $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \left[\alpha_1 \;\; \alpha_2 \;\; \alpha_3 \;\; \alpha_4 \;\; \alpha_5\right]^\top, \tag{4}$$

the actual lengths of the individual limbs $\boldsymbol{\ell}$:

$$\boldsymbol{\ell} = \left[\ell_1 \;\; \ell_2 \;\; \ell_3 \;\; \ell_4 \;\; \ell_5\right]^\top, \tag{5}$$

and the orientation of the robot's center point ε must be known (see Fig. 2). These quantities are defined as the variable to be optimized:

$$\boldsymbol{x} = \left[\; \boldsymbol{\alpha}^\top \;\; \boldsymbol{\ell}^\top \;\; \varepsilon \; \right]^\top. \tag{6}$$

Furthermore, the position of (at least) one fixed foot ($f_j \overset{!}{=} 1$) must be known. Then, the pose of the robot $\boldsymbol{\rho}$ can be determined under the assumption of a constant curvature by drawing arcs with the corresponding lengths and angles. A pose can therefore be described in generally as a function of $\boldsymbol{x}$ and the position $\boldsymbol{P} = [\boldsymbol{p}_1 \; \boldsymbol{p}_2 \; \boldsymbol{p}_3 \; \boldsymbol{p}_4]^\top = [\boldsymbol{p}_x \; \boldsymbol{p}_y]^\top \in \mathbb{R}^{4\times 2}$ and state $\boldsymbol{f}$ of all feet:

$$\boldsymbol{\rho} = [\boldsymbol{x} \;\; \boldsymbol{P} \;\; \boldsymbol{f}]. \tag{7}$$

For a given feasible initial pose, the next pose must be determined so that all fixed feet do not move. This can be achieved within a certain margin by deviating the bending angle from the reference angle and deviating the actual length from the nominal length. To describe this mathematically, a new index k is introduced, which assigns the quantities to a specific time step. The new positions of the fixed feet $\boldsymbol{P}_k$ are assumed to be the positions from the previous pose $\boldsymbol{P}_{k-1}$. This can be used to define the constraint for the next pose. All newly fixed feet must have the same position as in the previous step:

$$\left|\left|\text{diag}(\boldsymbol{f}_k)(\boldsymbol{P}_k - \boldsymbol{P}_{k-1})\right|\right|_2 \overset{!}{=} 0, \tag{8}$$

Authors are with Workgroup on System Technologies and Engineering Design Methodology, Hamburg University of Technology, 21073 Hamburg, Germany arthur.seibel@tuhh.de

It has already been mentioned that the bending angles $\boldsymbol{\alpha}$ and the lengths of the limbs $\boldsymbol{\ell}$ are quite variable. By defining

$$\boldsymbol{\ell}_n = [\ell_{n,1}\ \ell_{n,2}\ \ell_{n,3}\ \ell_{n,4}\ \ell_{n,5}]^\top \tag{9}$$

as the vector containing the nominal length of each limb, it is possible to quantify the length deviation. The orientation angles of the fixation actuators $\boldsymbol{\varphi}$ also have a certain margin. The value of the orientation angles can be calculated as a function of $\boldsymbol{\alpha}$ and ε (well, basically as a function of $\boldsymbol{x}$):

$$\boldsymbol{\varphi}(\boldsymbol{\alpha}, \varepsilon) = (25), \tag{10}$$

where the exact formula is given in the appendix (25). Now a objective function σ can be formulated which quantifies the deviations of length, angle and orientation:

$$\begin{aligned} \sigma(\boldsymbol{x}_k) \quad = \quad & w_\ell \left|\boldsymbol{\ell}_k - \boldsymbol{\ell}_n\right|_2 + w_\alpha \left|\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{\text{ref},k}\right|_2 \\ & + w_\varphi \left|\operatorname{diag}(\boldsymbol{f}_k)(\boldsymbol{\varphi}_k - \boldsymbol{\varphi}_{k-1})\right|_2 \end{aligned} . \tag{11}$$

The weighting factors can be interpreted physically. To do so, the weighting factor w_ℓ describes the elasticity of the limbs and w_α the bending stiffness of the limbs. The term weighted by w_φ describes the difference between the orientation of the newly fixed feet compared to the orientation in the previous time step. This can be seen as a dimension for the torsional stiffness of the fixation actuators. The objective function can be seen as a measure of the robot's inner stress. Therefore it is called σ referring to the nomenclature in mechanical engineering. The new pose can now be determined by solving the non-linear optimization problem:

$$\begin{aligned} & \min_{\boldsymbol{x}_k \in \mathcal{X}} \ \sigma(\boldsymbol{x}_k) \\ & \textit{subjected to} \quad \left|\left|\operatorname{diag}(\boldsymbol{f}_k)(\boldsymbol{P}_k - \boldsymbol{P}_{k-1})\right|\right|_2 = 0. \end{aligned} \tag{12}$$

Here $\mathcal{X}$ describes the set of allowed values. Each quantity inside $\boldsymbol{x}$ has bounds, which are given in the following table:

var	bounds
$\boldsymbol{\alpha}$	$[\boldsymbol{\alpha}_{\text{ref}} - b_\alpha,\ \boldsymbol{\alpha}_{\text{ref}} + b_\alpha]$
$\boldsymbol{\ell}$	$[(1 - b_\ell)\boldsymbol{\ell}_n,\ (1 + b_\ell)\boldsymbol{\ell}_n]$
ε	$[0°,\ 360°]$

These bounds can be tuned with the scalars b_α and b_ℓ. For solving the problem, for example the `SLSQP`-Algorithm provided by the python package `scipy.optimize` can be used. Note that the evaluation of the objective function (11) is quite cheap. The expensive part is the evaluation of the constraint function (8), since the calculation of all feet positions for a given $\boldsymbol{x}$ is opulent and outlined in the appendix (24)–(39).

In summary, it is possible to set up a function that can predict the next pose of the robot, depending on the reference input and the previous pose:

$$\boldsymbol{\rho}_k = [\boldsymbol{x}_k\ \boldsymbol{P}_k\ \boldsymbol{f}_k] = \text{fun}_{\mathcal{P}}\left(\boldsymbol{r}_k, \boldsymbol{\rho}_{k-1}\right) \tag{13}$$

This function can be tuned by the parameter set $\mathcal{P}$ given in the following table:

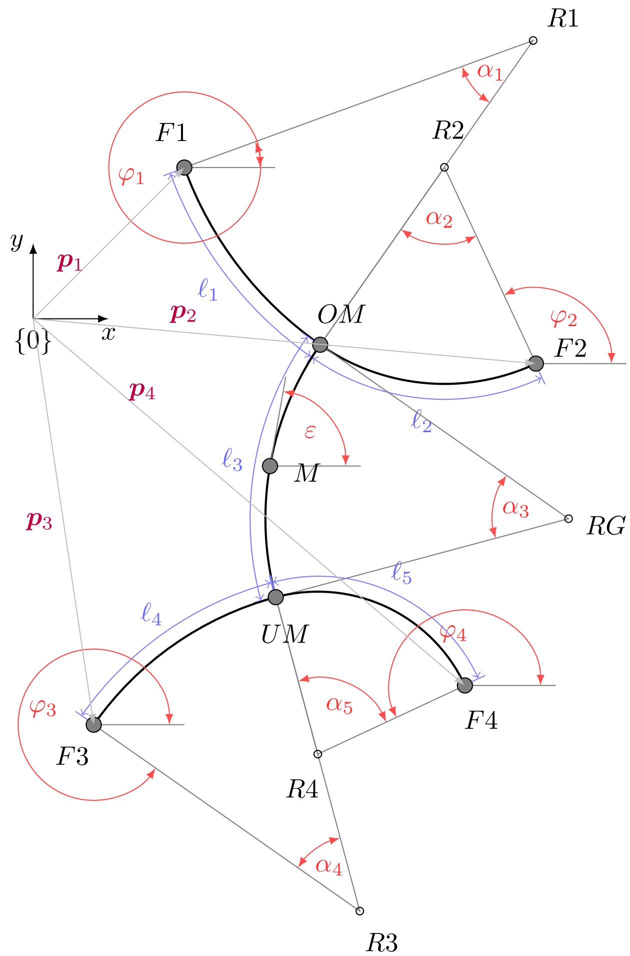

Fig. 2. Nomencalture

$\mathcal{P}$	description
b_α	allowed absolute deviation of the bending angle to the reference angle
b_ℓ	allowed percentage deviation of the length to the nominal length
w_ℓ	costs of the length deviation / Youngs-Modulus of the material
w_α	costs of the bending angle deviation / bending stiffness of the limbs
w_φ	costs of the orientation angle deviation / torsional stiffness of the fixation actuators

III. Finding optimal gait patterns

The presented algorithm can now be used to find new gait patterns. Since it provides the ability to predict the robot's pose for a given reference input, it is also able to predict all associated poses to a sequence of reference inputs by recursive application.

However, a valid initial pose is required to apply the presented algorithm recursively. Valid in the sense that the lengths and bending angles of the limbs are within the valid range. The condition (8), that the feet fixed in the previous step may not be moved, is void at this point, since no previous step exists. To calculate a valid initial pose, initial bending angles $\boldsymbol{\alpha}_0$, initial limb lengths $\boldsymbol{\ell}_0$ and orientation ε_0 can be freely defined. For the limb lengths $\boldsymbol{\ell}_0 = \boldsymbol{\ell}_n$ is recommended, as the " inner stress " of the initial pose is $\sigma(\boldsymbol{x}_0) = 0$. The feet positions $\boldsymbol{P}_0$ result then by application of equations (24)–(39). Thereby the initial coordinate of the front left foot $\boldsymbol{p}_{1,0}$ must be provided.

$$\begin{aligned} \boldsymbol{\rho}_0 &= \left[\begin{array}{c|c|c} \boldsymbol{x}_0 & \boldsymbol{P}_0 & \boldsymbol{f}_0 \end{array} \right] \\ &= \left[\begin{array}{c|c|c} \boldsymbol{\alpha}_0\ \boldsymbol{\ell}_n\ \varepsilon_0 & \boldsymbol{P}(\boldsymbol{\alpha}_0, \boldsymbol{\ell}_n, \varepsilon_0, \boldsymbol{p}_{1,0}) & 1\ 0\ 0\ 0 \end{array} \right] \end{aligned} \tag{14}$$

A gait pattern consists of a sequence of poses taken in a loop with a certain number of cycles. To find an optimal sequence of reference inputs, it is first necessary to define what is optimal. This depends entirely on the type of gait pattern we are looking for. Two examples are presented below.

A. Straight Gait

It is reasonable to assume that a symmetrical gait pattern will most likely lead to a straigth gait. Symmetrical in the sense that a certain pose is always followed by a pose mirrored to the longitudinal axis. In addition, the state of fixation of the feet is also known from the outset, since the diagonally opposite feet must always be fixed for a symmetrical gait (see previous research work - [1]). Thus, the running pattern has only five unknowns, namely the reference angles of the first pose – all other poses within the gait pattern are mirror images. Therefore the variable to be optimized can be defined as:

$$\boldsymbol{y} = \begin{bmatrix} \alpha_{\text{ref},1} & \alpha_{\text{ref},2} & \alpha_{\text{ref},3} & \alpha_{\text{ref},4} & \alpha_{\text{ref},5} \end{bmatrix}. \tag{15}$$

With these assumptions a structure for the still unknown straight gait pattern with n cycles $\boldsymbol{R}_S^n \in \mathbb{R}^{2n \times 9}$ can be given as:

$$\boldsymbol{R}_S^n(\boldsymbol{y}) = \left[\begin{array}{ccccccccc} y_1 & y_2 & y_3 & y_4 & y_5 & 1 & 0 & 0 & 1 \\ y_2 & y_1 & -y_3 & y_5 & y_4 & 0 & 1 & 1 & 0 \\ \hline y_1 & y_2 & y_3 & y_4 & y_5 & 1 & 0 & 0 & 1 \\ y_2 & y_1 & -y_3 & y_5 & y_4 & 0 & 1 & 1 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline y_1 & y_2 & y_3 & y_4 & y_5 & 1 & 0 & 0 & 1 \\ y_2 & y_1 & -y_3 & y_5 & y_4 & 0 & 1 & 1 & 0 \end{array} \right]. \tag{16}$$

If the robot is to move straight ahead, it is optimal to maximize the distance from start to end point. Assuming the longitudinal axis of the robot is initially aligned in positive y-axis ($\varepsilon_0 = 90°$), the performance of a gait pattern for straight motion with n cycles and for a given parameter set $\mathcal{P}$ can be quantified by using the following function $\Delta \boldsymbol{p}_y$:

$$\Delta \boldsymbol{p}_y^n(\boldsymbol{y}) = \begin{cases} \boldsymbol{\rho}_0 = \left[\begin{array}{c|c|c} \boldsymbol{y}\ \boldsymbol{\ell}_n\ \varepsilon_0 & \boldsymbol{P}(\boldsymbol{y}, \boldsymbol{\ell}_n, \varepsilon_0, \boldsymbol{p}_{1,0}) & 1\ 0\ 0\ 0 \end{array} \right] \\ \text{for } k = 1, \ldots, 2n: \\ \qquad \boldsymbol{\rho}_k = \text{fun}_{\mathcal{P}}(\boldsymbol{r}_k, \boldsymbol{\rho}_{k-1}) \\ \Delta \boldsymbol{p}_y = \left| \boldsymbol{p}_{y,2n} \right|_2 - \left| \boldsymbol{p}_{y,0} \right|_2 \end{cases} \tag{17}$$

where $\boldsymbol{P} = [\boldsymbol{p}_x\ \boldsymbol{p}_y]$ is divided into two column vectors, each containing the x and y coordinates of the four feet. The reference input of the k-th step $\boldsymbol{r}_k$ here is the k-th row of gait pattern $\boldsymbol{R}_S^n(\boldsymbol{y})$.

An optimal pattern for a straight motion of n cycles and a given parameter set $\mathcal{P}$ is the solution to the following minimization problem:

$$\min_{\boldsymbol{y} \in \mathcal{Y}} \ -\Delta \boldsymbol{p}_y^n(\boldsymbol{y}). \tag{18}$$

Here $\mathcal{Y}$ describes the set of allowed values. All reference angles of the legs should be $y_i \geq 0, \ i = 1, 2, 4, 5$. The reference angle of the torso is not constrained, since it can bend into both directions.

Most solvers require an initial guess to solve a minimization problem. Since the solution often depends strongly on this initial value, it is recommended to choose an initial value that is already familiar to be close to the optimum. From previous research [1] it is known, that the gait pattern corresponding to the reference input $\boldsymbol{y}_0 = [90°\ 0°\ -90°\ 90°\ 0°]$ is functional – theoretically, as well as practically. Therefore, $\boldsymbol{y}_0$ is chosen as the initial value for the minimization problem. For solving the problem, for example the `COBYLA`-Algorithm provided by the python package `scipy.optimize` can be used.

Figure 3 (a)–(c) shows the optimization results for different parameter sets $\mathcal{P} = [b_\alpha, b_\ell, w_\ell, w_\alpha, w_\varphi]$. The five poses from two and half cycles are depicted. The time course is color-coded – the darker the image, the newer it is. Start and end position of the left front foot are marked by a horizontal line. The limits of the bending angles $b_\alpha = 100°$ and lengths $b_\ell = .1$ are constant for all three optimizations. The costs for deviation from nominal length $w_\ell = 100$ and deviation from reference angle $w_\alpha = 10$ are also constant. The only difference between the three simulations is the weighting factor on the deviation of the feet orientations w_φ. In (a) $w_\varphi = 1$ is comparatively high. It is therefore *expensive* to twist a fixed foot relative to the previous pose. The resulting optimal gait pattern is the one corresponding to $\boldsymbol{y}_0$ and already known from [1]. If the cost of twisting the feet is reduced slightly ($w_\varphi = 0.1$), the gait pattern shown in (b) turns out to be optimal. Within the 2.5 cycles, the robot generates $1.46\times$ more shift in position compared to the simulation in (a). If the cost of twisting the feet is almost neglected ($w_\varphi = 0.0001$), the gait pattern shown in (c) results. Practically this gait is not possible, because the robot would trip over its own legs, but as a theoretical gimmick it is quite interesting. Compared to (a), more than twice as much ($2.11\times$) shift in position could be generated.

B. Case Study: Straight Gait

Since no values for the parameter set $\mathcal{P}$ are known so far, no statement can be made about what is really the optimal straight gait pattern. Therefore, one experiment was performed for each of the three running patterns. Since the robot and thus also the parameter set remain the same in all experiments, it is possible to determine which concrete values apply best to the parameter set. If the robot shows a

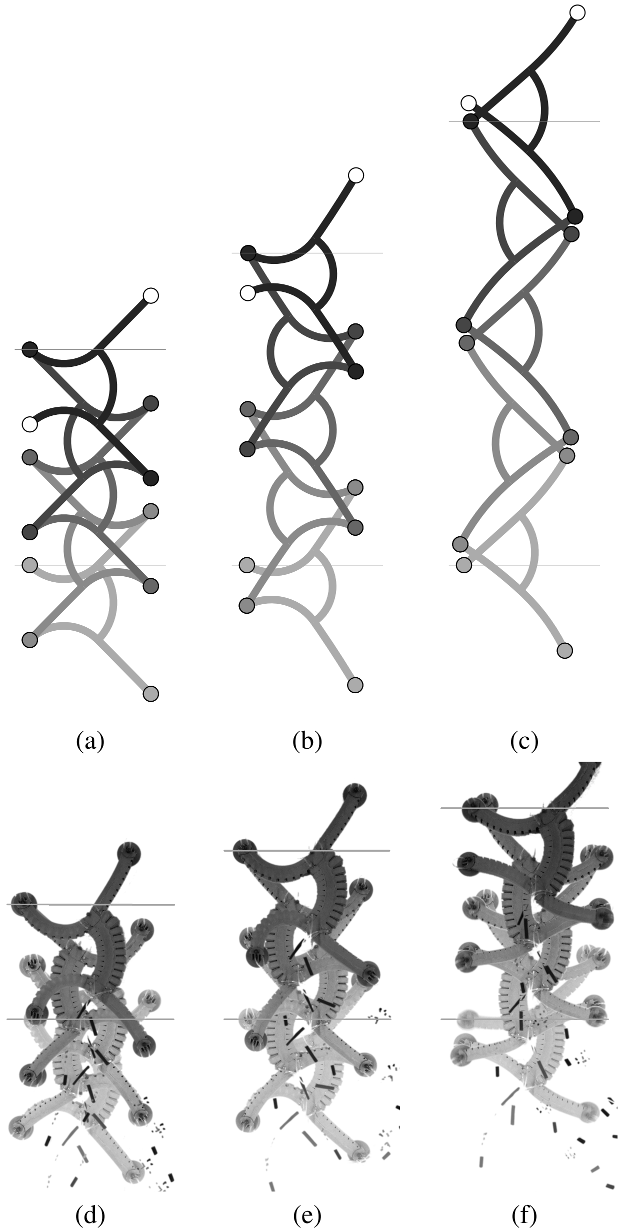

Fig. 3. Optimal straight gait patterns $\boldsymbol{R}_S^2(\boldsymbol{y})$ for different parameter sets $\mathcal{P} = [b_\alpha, b_\ell, w_\ell, w_\alpha, w_\varphi]$. (a) $w_\varphi = 1$, $\boldsymbol{y} = [90° \ 0° \ -90° \ 90° \ 0°]$. (b) $w_\varphi = .1$, $\boldsymbol{y} = [86° \ 4° \ -110° \ 83° \ 4°]$. (c) $w_\varphi = .0001$, $\boldsymbol{y} = [0° \ 18° \ -85° \ 10° \ 22°]$. (d)–(f) the corresponding experiment for the gait pattern.

similar behaviour in the experiment as in the simulation for a certain parameter set, it can be concluded that this parameter set reflects reality. If simulation and experiment are far apart, it can be assumed that the parameter set is incorrect.

In the experiment, an image was automatically taken each time the feet fixation were changed.These individual images were superimposed and also color-coded so that a comparison with the simulation is easily possible. The figures 3 (d)–(f) show the results. Again, the start and end position of the left forefoot was marked by a horizontal line. If we compare (a) and (d), we find that the shift in position in the experiment is only about half as high as in the simulation. The model presented here does not consider any friction effects or other external force influences. This is how this huge difference can be explained. The robot's poses, on the other hand, are almost identical in experiment and simulation. This speaks for the validity of this parameter set.

The poses in the experiment with the optimal gait pattern for slightly elastic ankles in (e) show a small deviation from those in the simulation (b). First, the robot is not able to achieve a torso bending angle of $110°$, otherwise the material could be damaged. On the other hand – and this is interesting – the deviation of the foot orientation between the individual poses in the experiment is a little smaller than in the simulation. This means that the weighting factor $w_\varphi = 0.1$ is a bit too low. However, it also shows that the parameter set in (a) is selected too conservatively, since the foot orientation has a certain margin and by using it, a shift in position can be generated that is around $1.5\times$ as high as in (d). Note that the same improvement also occurs in the simulation (b) compared to (a).

As assumed, the robot in the experiment for freely movable ankles in (f) shows clearly different behaviour than the simulation in (c). The individual poses can only be recognized with a lot of imagination. Nevertheless, about $1.9\times$ more feed could be generated than in the experiment in (d). Nevertheless, this running pattern is unsuitable, as there is great inner stress in the robot within the poses. This can be seen from the fact that a leg that has just been released abruptly deflects into the rest position. If one compares the start and end poses in (f), you will notice that they differ considerably. The distance between the two left feet is much greater in the end pose than in the starting pose. This is also an indication for the instability of this gait pattern.

C. Gait Pattern for a curve

In contrast to the straight motion, the gait pattern of a curve is not symmetrical. However, it still reasonable to assume that a cycle consists of two poses, each with diagonally opposite fixed feet. Accordingly, the running pattern of a curve has ten unknowns, namely the reference angles of the two poses of a cycle. This results in the variable to be optimized:

$$\boldsymbol{z} = \begin{bmatrix} \alpha_{\text{ref},1} & \alpha_{\text{ref},2} & \alpha_{\text{ref},3} & \alpha_{\text{ref},4} & \alpha_{\text{ref},5} \\ \alpha_{\text{ref},6} & \alpha_{\text{ref},7} & \alpha_{\text{ref},8} & \alpha_{\text{ref},9} & \alpha_{\text{ref},10} \end{bmatrix}. \quad (19)$$

As before, the structure of the gait pattern of a curve with n cycles $\boldsymbol{R}_C^n \in \mathbb{R}^{2n\times 9}$ can now be defined as:

$$\boldsymbol{R}_C^n(\boldsymbol{z}) = \left[\begin{array}{ccccc|cccc} z_1 & z_2 & z_3 & z_4 & z_5 & 1 & 0 & 0 & 1 \\ z_6 & z_7 & z_8 & z_9 & z_{10} & 0 & 1 & 1 & 0 \\ \hline z_1 & z_2 & z_3 & z_4 & z_5 & 1 & 0 & 0 & 1 \\ z_6 & z_7 & z_8 & z_9 & z_{10} & 0 & 1 & 1 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline z_1 & z_2 & z_3 & z_4 & z_5 & 1 & 0 & 0 & 1 \\ z_6 & z_7 & z_8 & z_9 & z_{10} & 0 & 1 & 1 & 0 \end{array}\right]. \quad (20)$$

To run a curve it is optimal to maximize or minimize the difference between the orientation of the initial pose ε_0 and that of the end pose after n cycles ε_{2n}, depending on whether a left or right curve is to be run. To define the initial pose $\boldsymbol{\rho}_0$, the reference angles of the first pose $\boldsymbol{z}_1$ of the searched gait pattern $\boldsymbol{R}_{\mathcal{C}}(\boldsymbol{z})$ are used in the same way as it already happened for straight gait. Now the performance of a gait pattern for a curve with n cycles can be quantified using the following objective:

$$\Delta\varepsilon^n(\boldsymbol{z}) = \begin{cases} \boldsymbol{\rho}_0 = \left[\ \boldsymbol{z}_1\ \boldsymbol{\ell}_n\ \varepsilon_0\ \middle|\ \boldsymbol{P}(\boldsymbol{z}_1, \boldsymbol{\ell}_n, \varepsilon_0, \boldsymbol{p}_{1,0})\ \middle|\ 1\ 0\ 0\ 0\ \right] \\ \text{for } k = 1, \ldots, 2n: \\ \qquad \boldsymbol{\rho}_k = \mathrm{fun}_{\mathcal{P}}(\boldsymbol{r}_k, \boldsymbol{\rho}_{k-1}) \\ \Delta\varepsilon = \varepsilon_{2n} - \varepsilon_0 \end{cases} \tag{21}$$

The optimal gait pattern for a left curve to a given parameter set $\mathcal{P}$ is then the solution to the optimization problem:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \ -\Delta\varepsilon^n(\boldsymbol{z}). \tag{22}$$

Here $\mathcal{Z}$ describes the set of allowed values. All reference angles of the legs should be $z_i \geq 0,\ i = 1, 2, 4, 5, 6, 7, 9, 10$. The reference angles of the torso $z_i,\ i = 3, 8$ are not constrained, since it can bend into both directions. For a right turn, multiply the objective function by -1. A starting value is required again to solve the minimization problem. Therefore $\boldsymbol{z}_0 = [\boldsymbol{y}_0\ \boldsymbol{y}_0]$ is chosen.

Figure 4 shows the optimization results for different parameter sets $\mathcal{P}$. The four poses of two cycles and the first pose of the third cycle are shown. The limits and costs for angle and length deviation are constant $[b_\alpha, b_\ell, w_\ell, w_alpha] = [100°, .1, 100, 10]$ in all simulations. Again, only the cost of the orientation deviation of the fixed feet w_φ was varied. The first three figures (a)–(c) show optimal left curves ($\min -\Delta\varepsilon$). In (a), twisting of the feet is relatively expensive $w_\varphi = 1$. Within the 2.5 cycles a rotation of $45°$ could be generated, which corresponds to $18°$/cycle. Note that the orientation of the fixed feet remains almost constant. In (b) the cost of twisting the feet was slightly reduced ($w_\varphi = .1$). The orientation deviation of the fixed feet is now slightly larger, but a rotation of $30°$/cycle was generated, almost twice as much as in (a). In (c) the robot has fully movable ankle joints ($w_\varphi = .0001$). In the 2.5 cycles it rotates around $235° \hat{=} 94°$/cycle. However, this gait pattern has bending angles of up to $221°$, making it practically difficult to move.

To convert a gait pattern for a left curve into that for a right curve, the reference angles must be mirrored. The mirror image $\tilde{\boldsymbol{r}}$ of a reference input $\boldsymbol{r}$ is defined as:

$$\tilde{\boldsymbol{r}}(\boldsymbol{r}) = \begin{bmatrix} r_2 & r_1 & -r_3 & r_5 & r_4 & \boldsymbol{f}^\top \end{bmatrix}. \tag{23}$$

If the mirroring is applied to all poses of the gait pattern for left turns, the robot runs forward right.

D. Case Study: Gait Pattern for a curve

As before, the different gait patterns for a curve are also tested in the experiment. The experiment is structured in the same way as for the straight gait. The figures 4 (d)–(f) show the result. Comparing the experiment in (d) with the simulation in (a), the main difference is that the robot in the experiment generated about half as much rotation within the 2.5 cycles as in the simulation ($25°$ instead of $45°$). We already know this factor of 2 from the comparison of simulation and experiment of the straight gait. Again, the lack of modelling of friction should be pointed out. The poses, on the other hand, are almost identical to the simulation. The deviation of the foot orientation between the individual poses is also very small, as demanded by the parameter set ($w_\varphi = 1$).

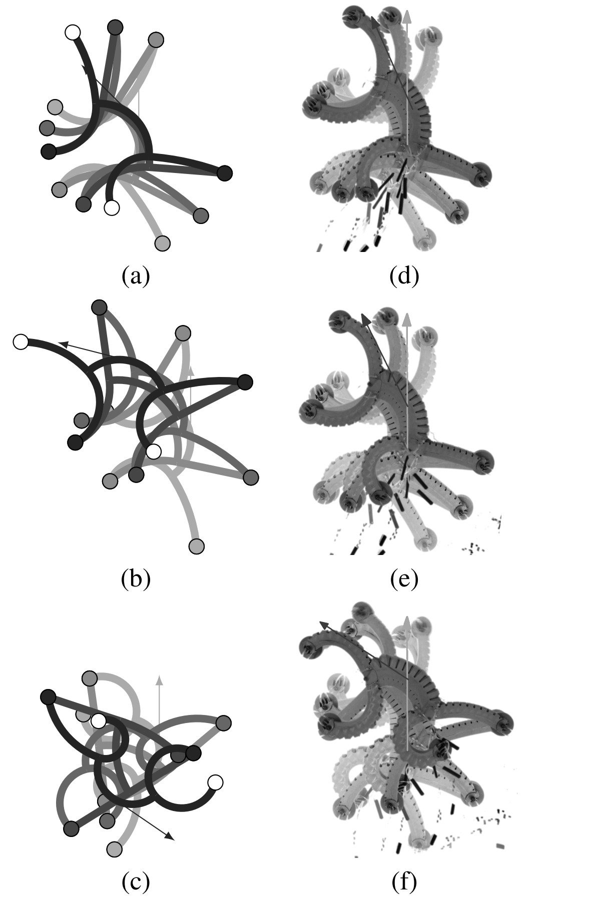

Fig. 4. Optimal left curve gait patterns $\boldsymbol{R}_{\mathcal{C}}^2(\boldsymbol{z})$ for different parameter sets $\mathcal{P} = [b_\alpha, b_\ell, w_\ell, w_\alpha, w_\varphi]$ ($\min -\Delta\varepsilon$): (a) $w_\varphi = 1$, $\boldsymbol{z} = [[97°\ 28°\ -98°\ 116°\ 17°]\ [79°\ 0°\ -84°\ 67°\ 0°]$. (b) $w_\varphi = .1$, $\boldsymbol{z} = [[104°\ 48°\ -114°\ 124°\ 27°]\ [72°\ 0°\ -70°\ 55°\ 0°]$. (c) $w_\varphi = .0001$, $\boldsymbol{z} = [[164°\ 124°\ -152°\ 221°\ 62°]\ [0°\ 0°\ -24°\ 0°\ 0°]$. (d)–(f) the corresponding experiment for the gait pattern.

In the second experiment (e) not much more rotation is generated ($28°$) than in the first. Compared to the simulation (b), this is only about a third. The deviation of the foot orientation is considerably lower than in the simulation, but slightly higher than in the first experiment (a). This in turn leads to the conclusion that the parameter set from (a) is too conservative and that from (b) is too loose.

In the third experiment twice as much rotation was generated as in the first two ($55°$). The poses match the simulation. However, it should be noted that the fixed feet change position. This can only be explained by the fact that the feet were not fixed over the entire movement period. This can be seen particularly well on the front right and rear left foot. Due to the strong actuation of the legs, the suction cup deforms end and fixation is no longer ensured. If the experiment is carried out on a horizontal plane, this has no great influence. On a vertical wall, on the other hand, this is the deciding factor for the crash.

IV. Conclusion

This paper presents an algorithm to predict the pose of the gecko-inspired soft robot from [?] for a given reference input. This algorithm is based on a geometric optimization problem. Essentially, arcs are arranged so that their end points correspond to the positions of the fixed feet. To be able to do this, the algorithm allows deviations in the length of the arcs, in their orientation and in their bending angle. The costs for these deviations are weighted with factors that can also be interpreted physically. Thus, the functional value of the objective function is a measure of the inner stress of a pose. The weighting factors are summarized in a parameter set $\mathcal{P}$.

This algorithm can now be used to find new gait patterns for the robot. In this case, the structure of the gait pattern is specified. For a straight gait, the structure consists of only two mirror-symmetrical poses that are repeated in a continuous loop. The optimal gait pattern can now be found by formulating a further optimization problem. The quantity to be maximized is the shift in position generated in a gait pattern with n cycles, which can be calculated by the recursive application of the algorithm. A stable pattern for the straight gait was found that generates $1.5\times$ more shift in position than the previously known one.

In the same way, the algorithm was used to find a gait pattern for a curve. The quantity to be maximized was the orientation of the torso. Again, a stable gait pattern was discovered that enables the robot to rotate on the spot.

With this knowledge, any trajectories can now be executed, since each curve can be described as a combination of straight lines and arcs with different radii. In future research work, the focus will be on tracking arbitrary trajectories.

Appendix

$$r_i = \frac{360\ \ell_i}{2\pi\ \alpha_i}, \quad \text{for } i \in [1, 2, 3, 4, 5] \tag{24}$$

$$\boldsymbol{\varphi} = \begin{bmatrix} \varepsilon - \alpha_1 - \frac{1}{2}\alpha_3 \\ \varphi_1 + \alpha_1 + \alpha_2 \\ 180^\circ + \alpha_3 - \alpha_2 + \alpha_4 + \varphi_2 \\ 180^\circ + \alpha_3 - \alpha_1 + \alpha_5 + \varphi_1 \end{bmatrix} \tag{25}$$

If f_1:

$$\boldsymbol{p}_{R1} = \boldsymbol{p}_1 + \begin{bmatrix} \cos(\varphi_1) \\ \sin(\varphi_1) \end{bmatrix} r_1 \tag{26}$$

$$\boldsymbol{p}_{OM} = \boldsymbol{p}_{R1} + \begin{bmatrix} \cos(\varphi_1 + \alpha_1) \\ \sin(\varphi_1 + \alpha_1) \end{bmatrix} r_1 \tag{27}$$

$$\boldsymbol{p}_{R2} = \boldsymbol{p}_{OM} + \begin{bmatrix} \cos(\varphi_1 + \alpha_1) \\ \sin(\varphi_1 + \alpha_1) \end{bmatrix} r_2 \tag{28}$$

$$\boldsymbol{p}_2 = \boldsymbol{p}_{R2} + \begin{bmatrix} -\cos(\varphi_2) \\ -\sin(\varphi_2) \end{bmatrix} r_2 \tag{29}$$

If f_2:

$$\boldsymbol{p}_{R2} = \boldsymbol{p}_2 + \begin{bmatrix} \cos(\varphi_2) \\ \sin(\varphi_2) \end{bmatrix} r_2 \tag{30}$$

$$\boldsymbol{p}_{OM} = \boldsymbol{p}_{R2} + \begin{bmatrix} -\cos(\varphi_2 - \alpha_2) \\ -\sin(\varphi_2 - \alpha_2) \end{bmatrix} r_2 \tag{31}$$

$$\boldsymbol{p}_{R1} = \boldsymbol{p}_{OM} + \begin{bmatrix} \cos(\varphi_2 - \alpha_2) \\ \sin(\varphi_2 - \alpha_2) \end{bmatrix} r_1 \tag{32}$$

$$\boldsymbol{p}_1 = \boldsymbol{p}_{R1} + \begin{bmatrix} -\cos(\varphi_1) \\ -\sin(\varphi_1) \end{bmatrix} r_1 \tag{33}$$

Both:

$$\boldsymbol{p}_{RM} = \boldsymbol{p}_{OM} + \begin{bmatrix} \sin(\varphi_1 + \alpha_1) \\ -\cos(\varphi_1 + \alpha_1) \end{bmatrix} r_3 \tag{34}$$

$$\boldsymbol{p}_{UM} = \boldsymbol{p}_{RM} + \begin{bmatrix} -\cos(\alpha_3 - 90 + \varphi_1 + \alpha_1) \\ -\sin(\alpha_3 - 90 + \varphi_1 + \alpha_1) \end{bmatrix} r_3 \tag{35}$$

$$\boldsymbol{p}_{R4} = \boldsymbol{p}_{UM} + \begin{bmatrix} \sin(\alpha_3 - 90 + \varphi_1 + \alpha_1) \\ -\cos(\alpha_3 - 90 + \varphi_1 + \alpha_1) \end{bmatrix} r_4 \tag{36}$$

$$\boldsymbol{p}_4 = \boldsymbol{p}_{R4} + \begin{bmatrix} -\cos(\varphi_4) \\ -\sin(\varphi_4) \end{bmatrix} r_4 \tag{37}$$

$$\boldsymbol{p}_{R3} = \boldsymbol{p}_{UM} + \begin{bmatrix} \sin(\alpha_3 - 90 + \varphi_1 + \alpha_1) \\ -\cos(\alpha_3 - 90 + \varphi_1 + \alpha_1) \end{bmatrix} r_3 \tag{38}$$

$$\boldsymbol{p}_3 = \boldsymbol{p}_{R3} + \begin{bmatrix} -\cos(\varphi_3) \\ -\sin(\varphi_3) \end{bmatrix} r_3 \tag{39}$$

References

[1] Lars Schiller. *Entwicklung eines Laufmusters und Entwurf einer weichen Laufmaschine nach biologischem Vorbild.* Projektarbeit, Arbeitsbereich Anlagensystemtechnik und methodische Produktentwicklung, Technische Universitt Hamburg, 2017.