Comparing Trotting and Turning Strategies on the Quadrupedal Oncilla Robot

Jonas Degrave, Michaël Burm, Tim Waegeman, Francis wyffels, and Benjamin Schrauwen

Electronics and Information Systems (ELIS), Ghent University

Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

Abstract—In this paper, we compare three different trotting techniques and five different turning strategies on a small, compliant, biologically inspired quadrupedal robot, the Oncilla. The locomotion techniques were optimized on the actual hardware using a treadmill setup, without relying on models. We found that using half ellipses as foot trajectories resulted in the fastest gaits, as well as the highest robustness against parameter changes. Furthermore, we analyzed the importance of using the scapulae for turning, from which we observed that although not necessary, they are needed for turning with a higher speed.

Keywords—*Quadrupedal robot, Oncilla, Trotting gait, Turning, Tracking*

I. INTRODUCTION

In the domain of robot locomotion, there is a growing interest in quadrupedal legged robots. One of the main reasons for this is that legged robots are better suited to walk over rough, irregular terrain compared to their wheeled counterparts [1]. Current examples of such legged robots used for research are for instance BigDog [2], Hyq [3] and StarlETH [4].

While the above examples are rather large, in this research a smaller robot is used: the Oncilla [5]. Advantages of a small robot platform include the reduced cost of the robot and surrounding infrastructure, as well as increased safety for the operators and overall less setup time and thus a faster development cycle. Another important small robot is for instance Little Dog [6].

Within the AMARSi project (see Section V), the Oncilla was designed with compliance in mind: several spring elements are included in each leg, in order to find more robust ways of locomotion that are at the same time easier to control. The possibility of storing and reusing energy in these springs should make more energy efficient locomotion possible [7]. This is very similar to the locomotion systems that can be found in nature, in obviously more advanced versions. On top of that, the added compliance means that the robot is more indulgent to external obstructions, and therefore safer for those working with it.

For this class of small, compliant, quadruped robots there already exists work where gaits were developed. Part of the previous research for this class of small compliant quadrupedal robots showed interest in slow but robust movement over very rough terrain, using techniques to find optimal foot placement [6]. Other research focussed on running in a bounding gait [8], [9], and more recently, trotting gaits have been developed for small compliant robots as well, e.g., for the Reservoir Dog [10] and the Cheetah-cub robot [11], two direct predecessors of the Oncilla.
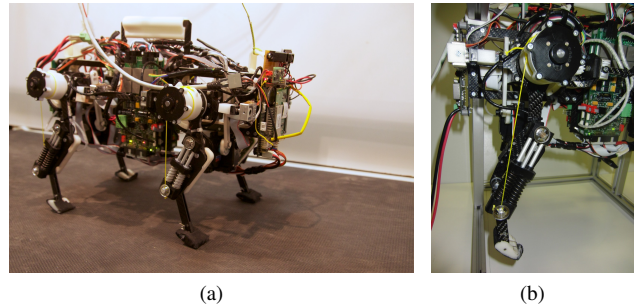

Fig. 1. In (a), the Oncilla robot on the treadmill is shown. Notice the three-segmented pantographic legs and the cable mechanism actuating the knee in (b).

The purpose of our research on the Oncilla is to continue this trend, but with more and improved actuators that enable us to develop faster movement and dynamical gaits, as well as more realistic turning behaviors. The Oncilla comes equiped with a large suite of sensors, which will enable future development of closed-loop gaits.

In this paper, we start off by comparing various ways to generate foot trajectories. These foot trajectories are inspired by the foot trajectories observed in our robot's biological counterparts. Moreover, we optimize the parameters for these trajectories using particle swarm optimization on the real robot and compare the speed-frequency characteristics for the different methods.

Secondly, we investigated the realisation of turning with various methods. For the two most promising methods, we compare the minimal turning radius and the performance when the robot is tracking an infrared led. Footage of the developed gaits and turning strategies is published in an online video[1].

II. METHODOLOGY

In this section we introduce our experimental setup, as well as the optimization algorithm used.

A. The Oncilla Robot

The Oncilla robot [5] is a quadrupedal robot with 12 actuated degrees of freedom, designed in the AMARSi project (see Section V). The robot has four legs, each with three actuators. It is light-weight, compliant, and has three-segmented pantographic legs, as shown in Fig. 1. The hip is actuated by a

---

[1]http://www.youtube.com/watch?v=A4MamwfcMFc

low-inertia actuator, and the knee joints are actuated through a cable mechanism by a second actuator in the main body. Each leg has a third servo actuator, serving as the robot's scapulae, enabling the robot to spread its legs (abduction and adduction). The leg design was loosely inspired by the legs of a cat [5].

To track the robot in our setup, we equipped the robot with a long-distance sensor and added the camera module from a Nintendo Wii remote. This camera module is capable of locating and tracking four infrared light sources at 100 Hz in an image with a resolution of 1024×768 pixels. In order to achieve this specifications, the module doesn't rely on further processing power of the receiver, because the tracking happens on-chip. Therefore this module is an excellent and cheap way of realising a vision function. Additionally, since we only use onboard sensors, it is possible to do future experiments outside the lab environment.

B. Experimental Setup

To test numerous gaits automatically and without interruptions, a treadmill is used. The measurements of the long-distance sensor on the robot are used to control the treadmill's speed, so the robot is kept in place. In this way, the robot can walk freely for an unlimited timespan. Walking for more than an hour at a time poses no severe problems. During the first experiments of gait optimization, the robot was kept in the middle of the track by having a light rope attached between the robot's head and an overhead rail in the middle of the track. This way the robot reorients itself in the direction of the track, while disturbances to the gait under test is kept to a minimum. An assistant is sitting next to the track in order to intervene if the robot's safety might be jeopardized by an unstable gait.

In order to perform the tracking, the robot follows an infrared led attached to the front of the treadmill, using its onboard camera. To evaluate the tracking properties, a second camera module was mounted on top of the treadmill, to provide an overhead view of two leds attached to the top of the robot (Fig. 2).

We run the algorithms controlling the robot's movements on a remote computer, which sends new commands to the robot every 10 ms. During the optimization process on the treadmill, we communicated with robot over an ethernet cable and powered the robot with an additional power cable. However, the same setup also proved to work equally good over wifi with the robot running on lithium polymer batteries. Complete wireless operation is thus possible, but not suited for long optimization runs due to the limited battery life (15-20 minutes on a 11.1 V, 1800 mAh battery).

C. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary optimization algorithm, which uses a set of candidate solutions that move towards a random combination of their own best solution and the best solution found so far by all particles [12]. PSO does not use gradients to optimize the parameters, which makes its use feasible for applications on real robots. The update equations of particle x_i at time step n are defined as follows [13]:

$$\begin{aligned} \mathbf{v}_i(n+1) \;=\; & \omega \mathbf{v}_i(n) + \phi_p r_p(n)(\mathbf{p}_i - \mathbf{x}_i(n)) \\ & + \phi_g r_g(n)(\mathbf{g}(n) - \mathbf{x}_i(n)) \end{aligned} \qquad (1)$$
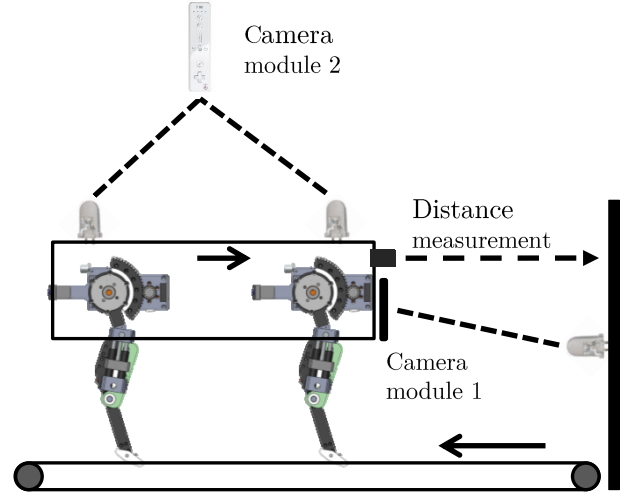

Fig. 2. The experimental setup used for optimization and measurements. The distance measurement is used to regulate the speed of the treadmill, to keep the robot in the center. Camera 1 is used to control the turning of the robot in order to make it track an infrared led, while camera 2 monitors the tracking performance.

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{v}_i(n+1). \qquad (2)$$

On timestep n, $\mathbf{x}_i(n)$ and $\mathbf{v}_i(n)$ are respectively the location and speed of particle i, $\mathbf{p}_i(n)$ and $\mathbf{g}(n)$ are respectively the particles previous best solution and the global best solution. $r_p(n)$ and $r_g(n)$ are randomly selected from a uniform distribution between 0 and 1. ω, ϕ_p and ϕ_g are parameters of the optimization algorithm. Here, ω is the inertia weight of a particle, ϕ_p and ϕ_g are the acceleration coefficients determining the magnitude of the random forces in the direction of respectively the personal best and the neighborhood best [12]. These were set at the values $\omega = 0.6571$, $\phi_p = 1.6319$ and $\phi_g = 0.6239$, which should yield good results for our search space, according to recent findings [14].

D. Gait Fitness Score

We optimize the gaits for speed. To achieve this, we let the robot run for ten seconds on the treadmill for each parameter set. During these ten seconds, we determine the walked distance by integrating the speed of the treadmill, while correcting for the difference between start and end position.

Secondly, we use the camera module on the robot to measure its stability. We attached an infrared led to the front of the treadmill, and track it with the camera module. We use this infrared light to register the robot movements in the transverse plane. By measuring the variance of the led's position in the image of the camera module, we obtain a good indicator of the stability of the robot's body during the gait, given that the distance from the robot to the front of the treadmill stays approximately the same.

The standard deviation of the led in the x- and the y-direction (resp. σ_x and σ_y) are then weighted and added to the travelled distance in order to obtain the fitness of the gait as shown in Equation 3. In this way, smooth gaits are favored over rougher ones. Horizontal movements are punished more severely than vertical movements, as vertical movements are

almost inevitable for fast gaits.

$$\text{fitness} = \text{distance} - \sigma_x/150 - \sigma_y/300 \qquad (3)$$

E. Gait transitions

Since our goal is to optimize gaits online on the treadmill, it is important to switch smoothly from one gait to the next. For this transition we change the phase velocity smoothly, while linearly interpolating between the other parameters of the gait as well. This way, the robot does not stumble or fall while transitioning from one gait to the next.

III. EXPERIMENTS

A. Trot Gait Optimization

1) Sine control signals: In a first approach, we used sine-based signals for the control of the gait, as it has previously been proven possible to achieve good result with such simple control signals [10], [15]. This means that the position of the hip and knee actuators are actuated with a sine wave, each with its own phase, amplitude and offset. One global frequency is used for the entire robot. To reduce the number of parameters, the left and right legs are given the same parameters, apart from the phase. The servo motors are given a constant signal, such that the plane in which the legs move does not change during the trot.

The problem with this approach is that stable and unstable gaits lay only small parameter changes apart. Therefore, many parameter combinations result in movements that can damage our robot. Optimization on the robot hardware was therefore infeasible and the approach of tuning the parameters automatically using PSO was not pursued further.

Despite these problems, a reasonable trot gait was found by careful manual tuning, with a resulting frequency characteristic shown in Fig. 4. This gait's velocity is proportional to the frequency up to about 1 Hz, at which point it flattens.

We conclude that sine signals do not work well on more heavy robot platforms such as the Oncilla which required careful tuning, this in contrast to smaller robots [10], [15]. Furthermore, the obtained sine based gaits for the Oncilla were not robust against small parameter changes, which makes live optimisation very hard.

2) Half ellipsoidal trajectories: In search for a better parameter space to optimize gaits, we changed our approach from defining the control signals directly, to defining the locus of the feet, and deriving the control signals from there. Therefore, we deduced the forward kinematics of the Oncilla and used these to solve the inverse kinematics numerically, because an algebraic solution does not exist.

We took inspiration from our robot's biological counterparts to choose an appropriate foot trajectory. Based on previous research in biology [16], [17], we decided to have our robot track simplified versions of foot trajectories of animals. As a first approximation, we used half ellipses. The flat bottom part for the stance phase and the half ellipse for the swing phase.

This approach relies on the tuning of multiple parameters which control the size and shape of the foot trajectory as
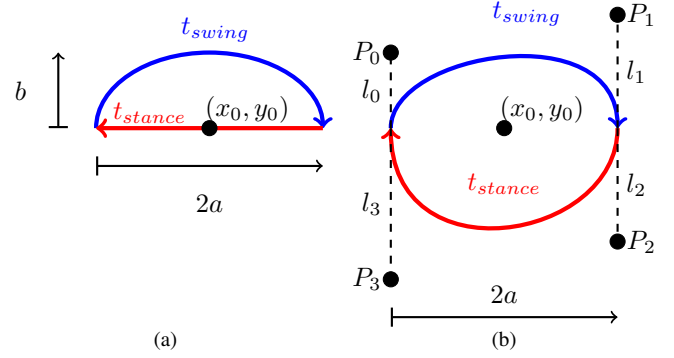

Fig. 3. The two foot trajectories used in this work. The left foot trajectory is shaped ellipsoidally and the curvature is tuned with the parameters a and b, the vertical and horizontal semi-axis of the ellipse. t_{stance} and t_{swing} are respectively the the time in which the foot does the stance and swing part of the trajectory. x_0 and y_0 are the coordinates of the location of the ellipse relative to the hip of the leg. The right foot trajectory is defined by two Bézier splines, controlled by four control points P_i.

depicted in Fig. 3a. To preserve symmetry, left- and right legs use the same ellipse shape. Adjacent legs have a phase shift of $180°$ in order to achieve a trot gait.

These parameters were optimized using PSO on the actual hardware. We allowed only parameter combinations that yield a half ellipse that fits in the reachable area of the foot. In contrast, the height b could be chosen larger than strictly possible. This results in a half ellipse with a dent in the top, because the knee cannot flex any further. This shape with a possible dent is comparable to observations made in dairy cows [16]. By making the assumptions that all feet have to move at the same speed when they are on the ground and that either two or four feet touch the ground, many parameters could be eliminated, leaving only the ones listed in Table I to be optimized.

By using 10 particles, 10 generations and a duration of 10 seconds per run with 1 second transition time, the optimizationwas done in less than 30 minutes. The values that were found are listed in Table I

Fig. 4 shows the frequency characteristic of this gait. The theoretical speed, based on the linear speed of the feet, is shown as a solid line. One can notice linear behavior, all the way up to 2 Hz. It is clear that this gait is usable up to higher frequencies than the manually tuned gait based on sine control signals. One can observe the discrepancy between the theoretical speed of the robot and the actual speed. Due to the compliant legs, the weight of the robot causes compression of the legs, with a shorter hip-foot distance than demanded as a consequence. This phenomenon leads to a lower ground speed of the foot, with a reduced robot speed as a result.

3) Bézier-curve trajectories: The sharp corners in the half ellipses are not biologically plausible. Therefore a third approach was pursued, using Bézier curves for the foot trajectories, see Fig. 3. For certain parameter values, these can approximate the half ellipses, but they provide the freedom for more elaborate, smooth trajectories with in particular a curved stance trajectory. This method gives the optimization algorithm a little more flexibility in the search for stable and good gaits.
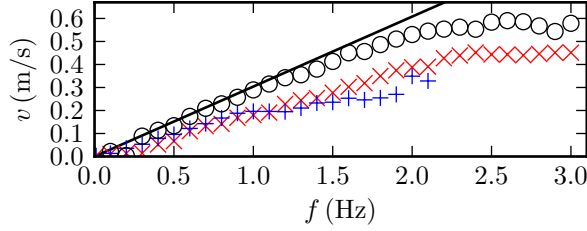
Fig. 4. Robot speed as function of gait frequency for the sine-based gait (+), the ellipse-based gait (×) and the spline-based gait (○). The solid line is the theoretically achievable speed based on the foot velocity during the stance phase.

In this approach, the parameters to be optimized are as follows: the major axis $2a$, 4 control points of the Bézier curves P_0, P_1, P_2, P_3 and the relative position of this shape to the knee of the robot (x_0, y_0). We only allow P_i to move vertically in order to reduce the number of parameters. The lengths of the segments are denoted as l_i. We give l_2 and l_3 the same values over the four legs, so the same stance trajectory is applied to each leg.

To optimize these parameters, we used the PSO algorithm with 20 particles, 20 generations and a duration of five seconds per run with one second of transition time. This way, we were able to optimize the gait in about 40 minutes, but with more parameter sets tried than in the previous case, to account for the increased number of parameters. After optimization, the gait attained a speed of $0.41 \, \text{m/s}$ at a frequency of $1.5 \, \text{Hz}$. When this result is compared to the other approaches in Fig. 4, it performs better than the gait based on sine-wave control signals ($0.24 \, \text{m/s}$) and the gait from our half ellipse approach ($0.27 \, \text{m/s}$). The maximum speed reached is $0.59 \, \text{m/s}$ at $2.6 \, \text{Hz}$. The final parameters are listed in Table I. The fore and hind indexes are used for parameters that differ between the fore and hind pair of legs.

Parameter	Value
x_0 (mm)	138.45
$y_{0,\text{fore}}$ (mm)	0.39
$y_{0,\text{hind}}$ (mm)	-6.83
a (mm)	75.95
b_{fore} (mm)	13.69
b_{hind} (mm)	26.12
$t_{\text{stance}} f$ (d.u.)	0.5
f (Hz)	1.94

(a)

Parameter	Value
x_0 (mm)	139
$y_{0,\text{fore}}$ (mm)	0
$y_{0,\text{hind}}$ (mm)	-18
a (mm)	91
$l_{0,\text{fore}}$ (mm)	68
$l_{0,\text{hind}}$ (mm)	11
$l_{1,\text{fore}}$ (mm)	63
$l_{1,\text{hind}}$ (mm)	32
l_2 (mm)	63
l_3 (mm)	32
$t_{\text{stance}} f$ (d.u.)	0.6
f (Hz)	2

(b)

TABLE I.   THE PARAMETERS OF THE HALF ELLIPSE GAIT (A) AND THE SPLINE GAIT (B) AS DEFINED IN FIG. 3 AND THEIR VALUES THAT ARE FOUND AFTER OPTIMIZATION WITH PSO

It is important to note that this new gait has a very linear characteristic, and follows the theoretical prediction almost perfectly up to $2 \, \text{Hz}$. Due to the non-linear stance trajectory, the compression of the springs in the compliant legs due to Oncilla's weight is largely compensated, with an increased speed as consequence.

B. Turning Strategies

After looking for the best gait approach, we searched for a good turning strategy with the gait based on half ellipses.

1) Turning by varying step size, keeping t_{stance} constant: In a first approach, we let the left side and the right side of the robot have a different step size by reducing a in the feet trajectories on one side, while keeping t_{stance} constant. This causes a slower ground speed on one side of the robot, with a rotation of the robot as consequence. The default step size of our ellips-based gait is $76 \, \text{mm}$ (Table I). The step size on one side of the robot could be reduced to $20 \, \text{mm}$ without causing disruptions in the gait dynamics and this setting was used in the further experiments.

2) Turning by varying step speed while keeping the step size constant: In a second approach, we use the same foot trajectories, but move the feet slower during the stance phase (increasing t_{stance}) and faster during the swing phase (decreasing t_{swing}) on one side of the robot, without changing a. We are interested in this approach, because the change in duty cycle during turning has also been observed biologically in running humans and mice [18], [19], even though humans and mice do use adduction and abduction for turning. This way, the distance travelled during the stance phase is the same on both sides, only the velocity differs. Subsequently we can evaluate whether the important part in the first approach was the decreased distance, or whether the important part is that the distance was travelled more slowly.

This approach failed because the feet of the robot stayed longer on the ground at one side of the body, causing the feet to bear less weight on average. The resulting extension of the compliant legs effectively cancels the intended turn.

3) Turning by varying step size while keeping the step speed constant: As third approach we varied the step size, but kept the step speed constant by varying t_{stance} and t_{swing} as well. This way, we can evaluate whether the changing step speed is an important part in the first approach.

This approach failed to work as well, because a $t_{stance} * f$ shorter than 0.5 implies a phase where both feet at the same side of the robot are off the ground, which isn't feasible. It appears that modifying t_{stance} and t_{swing} in a trot gait causes the robot to be hard to control.

4) Turning by varying motor torque: In a fourth approach, we lowered the maximum torque of the actuators on one side of the robot to find whether this technique was sufficient for turning the robot, inspired by a similar technique for bounding gaits [8]. We tried reducing the hip and the knee torque seperately and both at the same time. We found no stable way to do this with a trot gait. If the torque was not reduced enough, the robot didn't turn. When the torque was low enough to produce a visible difference, the robot stumbled and failed to produce a stable gait. We failed to find a good balance between these two extremes, and therefore this approach was not pursued any further.

5) Turning by abducting and adducting dynamically: A fifth strategy is to turn the robot by using the servos available in the scapulae, that can move the legs of the robot outwards from the saggital plane. Using this extra degree of freedom, it is possible to rotate the half ellipses around a vertical axis,
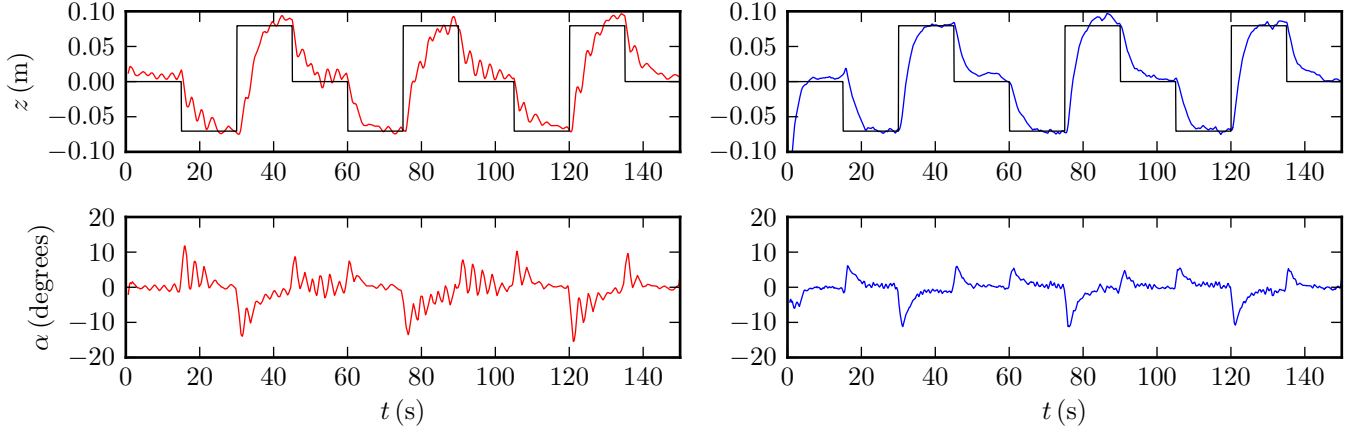
Fig. 5. On the top left, the lateral postition is plotted in function of time, while tracking by shortening the step length on one side of the robot. The black line indicates the set point and the red line the actual position. On the bottom left, the yaw of the robot compared to the forward direction is also shown. On the right side, the same charts are repeated while using the scapulae for turning. All these functions have been smoothed by averaging over the period of the gait, in order to reduce the noise caused by the movement of the body during the gait.

and make the feet move nonparallel to the saggital plane [20]. By rotating the front trajectories in one direction and the hind trajectories oppositely, the robot turns, comparable to a car with four wheel steering. This is also the behavior observed in mice [19].

This approach works very well on the Oncilla robot, even though the Oncilla scapulae have only a limited range of motion, about $5°$ of adduction and $10°$ of abduction, so the maximal theoretical sidewise step is about $4.2\,cm$. In our experiments, we used a maximum rotation of the foot trajectories aroud the vertical axis of $20°$. Larger rotations would result in going outside the reachable area of the foot.

C. Comparing the turning strategies

To compare both turning strategies, we first measured how quickly they can shift the Oncilla laterally on a treadmill. This means that the robot needs to end up walking straight ahead again, only shifted to the left or to the right.

In order to let the robot know whether it is heading in the right direction, we use the camera module on the robot. This way we can locate an infrared led mounted in the front of the treadmill, as shown in Fig. 2. By feeding this information inside a simple P-controller that controls the turning rate to keep the infrared dot in the middle of the view, the robot can follow this infrared light. We assumed the turning rate to be proportional to a in the case of varying step size, and proportional to the abduction in case of using the scapulae. We use only a simple P-controller, in order to test how controllable the different approaches are. By using more advanced controllers, the tracking capabilities will certainly improve, but this was not the intent of this paper. This P-parameter was consequently hand-tuned for the best performance in tracking.

The results of this experiment are shown in Fig. 5. It is apparent that using the scapulae for turning outperforms changing the step size. While it is definitely possible to track by only varying step size, one can observe that the Oncilla robot tends to oscillate more, both laterally and in yaw. It is

also slower to reach its goal position. On the other hand, the strategy using the scapulae tends to reach the desired position faster and more stable, with very little oscillations on the yaw.

In a second experiment, we also measured how fast both approaches could turn. To do this, we fixed the goal direction for both strategies at the same point, and observed their motion using a simple motion capture setup, with a single camera module mounted on the ceiling for a top view, $2.40\,m$ from the ground.

Strategy	Speed	Angular speed	Turning radius
Step size strategy	$0.068\,m/s$	$8.0°/s$	$0.448\,m$
Scapulae strategy	$0.22\,m/s$	$20°/s$	$0.610\,m$

TABLE II. THE SPEED, ANGULAR SPEED AND TURNING RADIUS OF THE CENTER OF THE ROBOT

The results are depicted in Fig. 6. From these experiments, we have also derived the data in Table II. Both strategies use the same gait at $1.5\,Hz$ which had a speed of $0.30\,m/s$ moving forwards. A first observation is that when turning the robot with smaller steps, the robot's body is not parallel to the velocity. We can observe that the robot's front is making a smaller circle than the robot's rear. This is not the case when the robot is turned using the scapulae, which results in a very slight speed loss compared to walking forwards ($8.3\,\%$) and a higher angular speed while turning. However, due to the faster motion, the turning radius is $36\,\%$ higher compared to using a smaller step size for turning.

IV. CONCLUSIONS

In this paper, we demonstrated that using a half ellipse as a biologically inspired base shape for the foot trajectory holds a good balance between the optimization time and the resulting gait performance. The fact that the gait optimized with this trajectory could maintain its performance up to higher speeds, also points to favoring this approach over sine-based control signal methods.
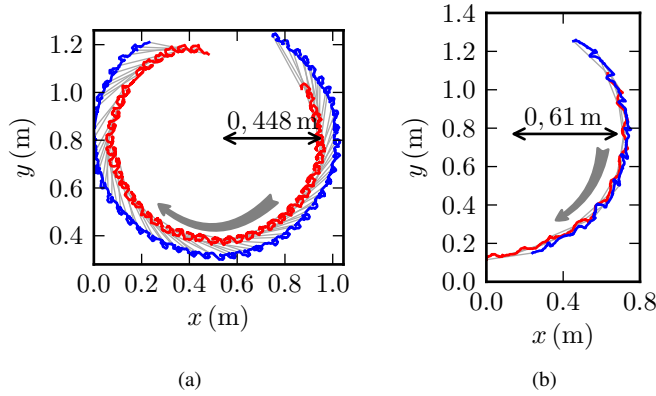
Fig. 6. In (a), the location of the robot turning by changing its step size is shown from a top view. The red line is the location of the front of the robot over time, the blue line the location of the hind of the robot. The turning radius and direction are also indicated. In (b), the same data is shown, but this time when using the scapulae for turning. The data was limited by the small viewing angle of the camera module. The turning radius was measured by fitting a circle to the measured center of the robot over time.

We have also shown the importance of having scapulae for turning, as was observed previously in nature. Turning without scapulae is also possible, albeit more slowly. Another drawback of turning without scapulae is that the robot's rotation is not aligned with its velocity. Using the scapulae, it is possible to maintain the robot's heading while turning and thus also its speed.

This research shows also that it is feasible to develop and optimize gaits without relying on models, using observations from nature. The models often oversimplify the physics involved in a complex robot, especially when they are small and compliant, making their results hard to transfer to the actual robot. Optimization using only hardware can be made possible by limiting the number of parameters to be optimized,

Since we only conducted our experiments on the Oncilla, our conclusions cannot be blindly transferred to other hardware platforms with different characteristics. We note however that qualitatively we obtain similar results to previously observed behaviour in quadrupedal animals. These findings also confirm comparable results in other robots.

We want to conclude that it is challenging to optimize a gait on a hardware robot, due to the limited system time available. On the Oncilla robot, we found that using a half ellipse approach was enough to obtain a good gait performance, because it has a small enough number of parameters for optimization on the actual robot. Additionally, we found that scapulae are not necessary for turning, but that they are needed in order to turn with higher speeds.

V. ACKNOWLEDGMENTS

REFERENCES

[1] P. González-de Santos, E. Garcia, and J. Estremera, *Quadrupedal locomotion: an introduction to the control of four-legged robots.* Springer Berlin, 2006.

[2] M. Raibert, K. Blankespoor, G. Nelson, R. Playter *et al.*, "Bigdog, the rough-terrain quadruped robot," in *Proceedings of the 17th World Congress*, 2008, pp. 10 823–10 825.

[3] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[4] M. Hutter, C. Gehring, M. Bloesch, M. Hoepflinger, C. D. Remy, and R. Siegwart, "Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *Int. Conf. on Climbing and Walking Robots (CLAWAR)*, 2012.

[5] A. Sproewitz, L. Kuechler, T. A, M. Ajallooeian, M. D'Haene, R. Moeckel, and A. Ijspeert, "Oncilla robot: a light-weight bioinspired quadruped robot for fast locomotion in rough terrain," in *Symposium on adaptive motion of animals and machines*, 2011.

[6] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 814–820.

[7] J. Hurst and A. Rizzi, "Series compliance for an efficient running gait," *Robotics & Automation Magazine, IEEE*, vol. 15, no. 3, pp. 42–51, 2008.

[8] X. Wang, M. Li, P. Wang, and L. Sun, "Running and turning control of a quadruped robot with compliant legs in bounding gait," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 511–518.

[9] M. de Lasa and M. Buehler, "Dynamic compliant quadruped walking," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 3153–3158.

[10] F. Wyffels, M. D'Haene, T. Waegeman, K. Caluwaerts, C. Nunes, and B. Schrauwen, "Realization of a passive compliant robot dog," in *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*. IEEE, 2010, pp. 882–886.

[11] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, 2013.

[12] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[13] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 69–73.

[14] M. E. H. Pedersen, "Good parameters for particle swarm optimization," *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001*, 2010.

[15] F. Iida and R. Pfeifer, "Cheap rapid locomotion of a quadruped robot: Self-stabilization of bounding gait," in *Intelligent Autonomous Systems*, vol. 8, 2004, pp. 642–649.

[16] F. Flower, D. Sanderson, and D. Weary, "Hoof pathologies influence kinematic measures of dairy cow gait," *Journal of dairy science*, vol. 88, no. 9, pp. 3166–3173, 2005.

[17] L. Shen and R. E. Poppele, "Kinematic analysis of cat hindlimb stepping," *Journal of neurophysiology*, vol. 74, no. 6, pp. 2266–2280, 1995.

[18] R. M. Walter, "Kinematics of 90 running turns in wild mice," *Journal of experimental Biology*, vol. 206, no. 10, pp. 1739–1749, 2003.

[19] E. Gruntman, Y. Benjamini, and I. Golani, "Coordination of steering in a free-trotting quadruped," *Journal of Comparative Physiology A*, vol. 193, no. 3, pp. 331–345, 2007.

[20] D. Golubovic and H. Hu, "Parameter optimisation of an evolutionary algorithm for on-line gait generation of quadruped robots," in *Industrial Technology, 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 221–226.