Navigation of Miniature Legged Robots Using a New Template

Konstantinos Karydis, Yan Liu, Ioannis Poulakakis, and Herbert G. Tanner

***Abstract*— This paper contributes to the area of miniature legged robots by investigating how a recently introduced bio-inspired template for such robots can be used for navigation. The model is simple and intuitive, and capable of capturing the salient features of the horizontal-plane behavior of an eight-legged miniature robot. We validate that the model can be combined with readily available navigation techniques, and then use it to plan the motion of the eight-legged miniature robot, which is tasked to crawl at low speeds, in obstacle-cluttered environments.**

I. INTRODUCTION

Legged robots have the potential to traverse a wide range of challenging terrains, where their wheeled counterparts may not be successful. Miniature legged robots, in particular, can also reach areas where larger ones cannot fit. In addition, they can be manufactured in a fast and relatively inexpensive manner, thus allowing for deployment in large numbers. These features create new opportunities in applications involving building and pipe inspection, search-and-rescue, as well as Intelligence, Surveillance, and Reconnaissance (ISR).

These new opportunities have promoted the development of a variety of bio-inspired multi-legged robots of small scale. Examples include the cockroach-inspired hexapod [1], the three-spoke rimless wheeled Mini-Whegs [2], [3], the 3D-printed robots STAR [4] and PSR [5], and i-Sprawl [6]. Using the Smart Composite Microstructure (SCM) technique [7], several minimally actuated palm-sized crawling robots have been fabricated; see for example DynaRoACH [8] and OctoRoACH [9] (Fig. 1).

Despite the introduction of a large number of small legged robots, our understanding on how autonomous navigation can be performed at these scale is still limited. Indeed, with few exceptions [10], [11], analysis is generally scarce. The few available robot modeling approaches have been motivated by car-like robot methodologies. Yet, bio-inspired models may be more suitable for capturing intrinsic robot behaviors associated with the legs.

Most of the existing bio-inspired modeling approaches yield horizontal-plane reduced-order dynamical models. The Lateral Leg Spring (LLS) model [12]–[14] offers justification for lateral stabilization [15], and is used for deriving turning strategies [16], [17] for hexapedal runners. The most common configuration of the LLS consists of a rigid torso and two prismatic legs modeled as massless springs, each representing the collective effect of a support tripod. The Sliding Spring Leg (SSL) model [18] includes the sliding effects of the leg-ground interaction in hexapedal robots.

This work is supported in part by NSF under grants CMMI-1130372 and CNS-1035577, and by ARL MAST CTA # W911NF-08-2-0004.

The authors are with the Department of Mechanical Engineering, University of Delaware. Email: {kkaryd, liuyan, poulakas, btanner}@udel.edu

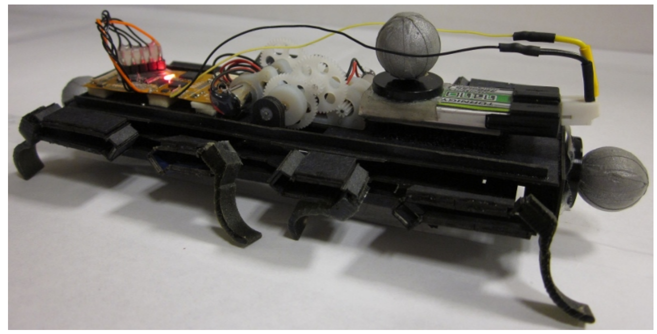


Fig. 1. The OctoRoACH, designed at the University of California, Berkeley. It is 130 mm-long, weights 35 g, and reaches a maximum speed of 0.5 m/s.

However, the dynamic nature of these models presents challenges to navigation at *low* crawling speeds. In this regime, surface forces dominate over inertia effects [19], [20]—yet, detailed ground interaction descriptions for integration into a dynamical model are unavailable [21], and the connection between model parameters and robot control parameters is unclear [8]. To tackle these issues, a *kinematic* template called the Switching Four-bar Mechanism (SFM) is introduced in [22] and studied in [23]. Motivated by the footfall pattern of the OctoRoACH [9], the model captures the average behavior of the robot when crawling at low speeds in a quasi-static fashion, and allows for a direct mapping between model parameters and robot kinematics.

This paper validates the suitability of the considered template for the OctoRoACH. The procedure involves the use of a motion capture system to record three curvature-parameterized motion primitives: (i) *straight line*, (ii) *clockwise turn*, and (iii) *counter-clockwise turn*. Solving a constrained optimization problem yields nominal model parameter values that make the model's behavior match experimentally observed robot data, on average. These data-based primitives are then used in an RRT solver [24, Section 7.2.2] for finding paths in environments populated with obstacles.

The work in this paper is part of our effort to port navigation and planning tools into the domain of miniature legged robots, and extends earlier work [10] by investigating the potential of bio-inspired models for navigation. The efficacy of the SFM template in navigation at the miniature scale opens the way for linking high-level navigation objectives to control strategies implementable at the physical platform. In principle, this template-based approach can be applied to a range of miniature crawling robots.

II. The Switching Four-bar Mechanism

The SFM template (Fig. 2(a)) is a horizontal-plane model consisting of a rigid torso and four rigid legs [23]. The legs move according to the foot-fall pattern depicted in Fig. 2(b). As the gait is executed, the torso and the legs form two alternating four-bar linkages, parameterized by the leg touchdown and liftoff angles, and the leg angular velocity.

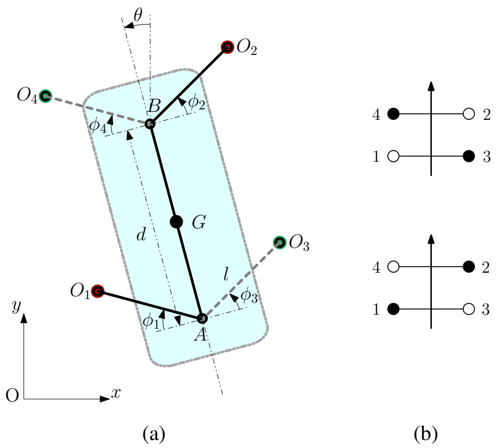


Fig. 2. (a) The SFM template. It consists of two pairs of legs that become active in turns, forming two fourbar linkages, $\{O_1A, AB, BO_2\}$ and $\{O_3A, AB, BO_4\}$. d is the distance between the two hip-point joints A and B, l denotes the leg length, and G is its geometric center. (b) The foot-fall pattern followed by the model.

A. Key Features

With respect to Fig. 2(a), the two leg pairs $\{AO_1, BO_2\}$, and $\{AO_3, BO_4\}$ are denoted as right and left pair, respectively. It is assumed that no slipping occurs between the tips of the legs and the ground, and that only one pair is active at all times—resulting to a 50% duty factor between legs.

The state of the model is a tuple $(x_G, y_G, \theta) \in \mathbb{R}^2 \times \mathbb{S}$, where (x_G, y_G) denotes the position of the geometric center of the model, G, with respect to some inertial coordinate frame $\{O\}$, and θ is the angle formed between the longitudinal body-fixed axis and the y-inertial axis. The evolution of the state during each step is determined by the kinematics of the respective active pair. Since each pair is kinematically equivalent to a four-bar linkage, the motion at every step is fully determined by one degree of freedom, taken here to be the angle ϕ_1 for the right pair, and the angle ϕ_3 for the left.

The geometric characteristics of a model path strongly depend on the values of the model parameters. Different combinations of touchdown and liftoff angles can produce very different path profiles, the geometric features of which are characterized by tools from differential geometry [25].

B. Characterizing the Geometry of a Model Path

Due to the fact that the model involves switching between two different four-bar mechanisms, the resulting geometric paths are piecewise differentiable. At the switching point, however, we observe an instantaneous change in the direction of motion. To calculate the curvature of such paths, produced by a specific combination of parameters values, we use the *Gauss-Bonnet Theorem* [25, Section 4-5]. In our planar configuration it results in

$$\sum_{j=0}^{L} \int_{s_j}^{s_{j+1}} k(s)\mathrm{d}s + \sum_{j=0}^{L} \chi_j = 2\pi \ , \tag{1}$$

where, L is the total number of steps taken by the SFM template, s_j is the arc length of step j, and $k(s)$ is the curvature of the curve component produced at each step:

$$k(s) = \frac{x'y'' - x''y'}{((x')^2 + (y')^2)^{\frac{3}{2}}} \ .$$

The quantity χ_j is the instantaneous change in the direction of motion of G when the model transitions from step j to step $j+1$ (Fig. 3).

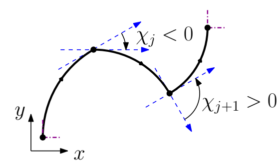


Fig. 3. Instantaneous change in the direction of motion when switching between steps. The sign of the angle is determined by the right-hand rule.

An appropriate number of steps allows the model to transcribe a closed circular curve, for which

$$2\pi R = \int_c \mathrm{d}s \ . \tag{2}$$

Combining (1) and (2) yields

$$R = \frac{\int_c \mathrm{d}s}{\sum_{j=0}^{L} \int_{s_j}^{s_{j+1}} k(s)\mathrm{d}s + \sum_{j=0}^{L} \chi_j} \ . \tag{3}$$

Then, the *average* path curvature produced by specific values for the touchdown and liftoff angles is obtained from (3) as $k_{\mathsf{path}} = 1/R$. In principle, (3) allows one to translate "macroscopic" requirements regarding desired path curvatures into model parameters realizing them; see Section III-D below.

C. Model Properties

Figure 4 graphically presents the set of reachable states from an initial state $q_0 \in \mathcal{C} \subset \mathbb{R}^2 \times \mathbb{S}$ at time T, denoted $R(q_0, T)$, for $T = 3$ seconds. Without loss of generality, we pick $q_0 = (0, 0, 0)$ in units of [cm, cm, deg], set the parameters d and l to 13 cm and 3 cm respectively, and choose the angular velocity of both leg pairs to $\dot{\phi}_{RL} = 8.02$ [deg/sec]. We also assume that all four legs are initiated with the same touchdown angle (i.e. $\phi_1^{\mathrm{td}} = \phi_2^{\mathrm{td}} = \phi_3^{\mathrm{td}} = \phi_4^{\mathrm{td}}$).

The reachable set gives us insight into the controllability properties of the model. From the graph produced in Fig. 4, it follows that this system is *accessible* [24]. In fact, we can achieve *small-time local accessibility* if we restrict the problem to $\mathbb{R}^2$, and treat the orientation θ as a parameter.

III. Adapting the SFM to Octoroach Behaviors

Experimental data is used to identify the SFM parameters that enable the model to match the average robot behavior.

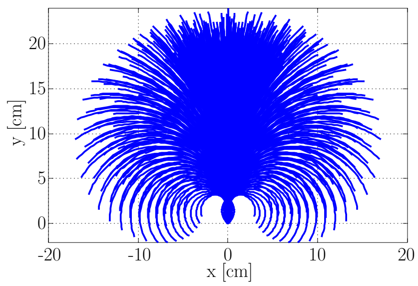


Fig. 4. The set of reachable states produced by the SFM template, starting at the initial state $x_0 = (0, 0)$ cm, $\theta_0 = 0^{\text{o}}$, and for a time span of 3 sec. Not all possible combinations of touchdown and liftoff angles are shown; this leads to some small areas not being covered by the paths.

A. Model–Robot Relation

The OctoRoACH is designed to follow an alternating tetrapod gait, as shown in Fig. 5(a). Legs $\{1, 2, 3, 4\}$ form the "right" tetrapod, and legs $\{5, 6, 7, 8\}$ form the "left" tetrapod. The ipsilateral legs of each tetrapod touch the ground at the same instant, and rotate in phase with the same angular velocity. The abstract eight-legged model of Fig. 5(b) illustrates this coupling, based on which we combine ipsilateral legs of each tetrapod into a single "virtual" leg. This combination yields the SFM template (Fig. 5(c)), where contralateral virtual legs (e.g., $\{O_1, O_2\}$) represent the collective effect of the tetrapod they replace (e.g. $\{1, 2, 3, 4\}$) [23].

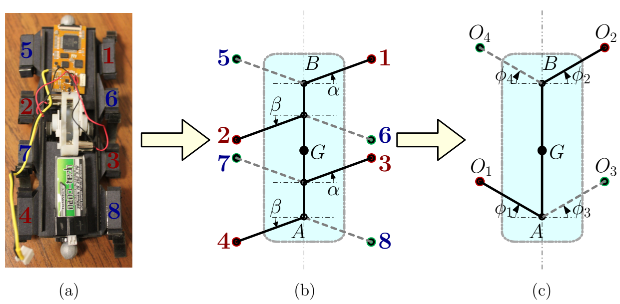


Fig. 5. Relating the SFM template to the OctoRoACH. (a) The foot-fall pattern of the robot, is an alternating tetrapod gait. Legs $\{1, 2, 3, 4\}$ form the right tetrapod, and legs $\{5, 6, 7, 8\}$ form the left tetrapod. (b) An eight-legged kinematic simplification of the gait mechanism used by the robot. The ipsilateral legs of each tetrapod are coupled, forming the angles α and β. This is shown here for the right tetrapod; the left tetrapod operates similarly. (c) The SFM is recovered by grouping coupled legs within a tetrapod into a single virtual leg inducing the same displacement. Legs $\{1, 2, 3, 4\}$ reduce to the pair $\{O_1, O_2\}$, while legs $\{5, 6, 7, 8\}$ reduce to the pair $\{O_3, O_4\}$.

B. A Library of Primitive Behaviors

The robot has two motors, each driving the legs on its own side. The two motor gains, K_L, and K_R, control the leg velocities of the left and right side, respectively. Then, we define three motion primitives: (i) *straight line* (SL), (ii) 90^o *clockwise turn* (CW), and (iii) 90^o *counter-clockwise turn* (CCW); . Table I gives the gains realizing these primitives. Note that one may choose to work with more primitives—see [23]. The particular primitives considered here capture the gross behavior of the robot while keeping the computational complexity of the navigation problem low.

To specify the target curvatures for the primitives we first calculate the curvature of the tightest robot turn.[1] We experimentally measured it at 0.25 cm^{-1}. The range of motion is thus prescribed between this turning curvature and zero, and is partitioned into four sectors. The curvature of the CW and CCW primitives is set at 0.063 cm^{-1}. The SL primitive has zero curvature, in theory; however, measurement noise, robot design errors, and random ground interaction effects, contributed to a nonzero experimentally measured curvature. SL paths are therefore associated with curvatures less than 0.01 cm^{-1} (Table I).

TABLE I

A LIBRARY OF PRIMITIVE BEHAVIORS FOR THE OCTOROACH

Type	Description	Motor Gains (K_L, K_R)	Target Orientation [deg]	Target Curvature [cm^{-1}]
CW	Clockwise 90^o Turn	$(60, 20)$	$\theta \simeq -90^o$	0.063
SL	Straight-Line	$(40, 40)$	$\theta \simeq 0^o$	≤ 0.01
CCW	Counter-Clockwise 90^o Turn	$(20, 60)$	$\theta \simeq 90^o$	0.063

C. Experimental Construction of Motion Primitives

The three primitives of Table I are realized by collecting open-loop planar position and orientation measurement data. The measured states comprise the planar position of the geometric center of the robot (x_G, y_G), and its orientation θ; see Fig. 2(a). By convention, positive changes in the orientation correspond to counter-clockwise angles θ.

We collect data from a total of 250 paths for each primitive. Data is captured with the use of a motion capture system at 100 Hz. All trials last 3 sec, and are conducted on a rubber floor mat surface. The robot is manually set into a designated start area with an initial state set at $(x_G, y_G, \theta) = (0,\ 0,\ 0)$ [cm, cm, deg]. Initial pose errors are bounded by data statistics, and are shown in Table II. Note that these error bounds account for measurement noise as well.

TABLE II

INITIAL POSE ERROR STATISTICS

Type	Mean [cm cm deg]	Standard Deviation [cm cm deg]
CW	$(-0.156,\ -0.041,\ 1.23]$	$[0.177,\ 0.141,\ 1.37)$
SL	$(-0.007,\ 0.027,\ 0.06]$	$[0.234,\ 0.054,\ 1.81)$
CCW	$(-0.322,\ -0.012,\ 2.50]$	$[0.156,\ 0.130,\ 1.23)$

Figure 6 provides insight on the choice of the 3 sec duration for our data collections and the generated motion primitives. The histogram shows how individual experimental paths disperse as time elapses. It can be seen that after the end of the 3 sec period, there is a very high dispersion around

[1]The tightest CW and CCW turns are achieved by setting the motor gains to $(80, 0)$, and $(0, 80)$ respectively.

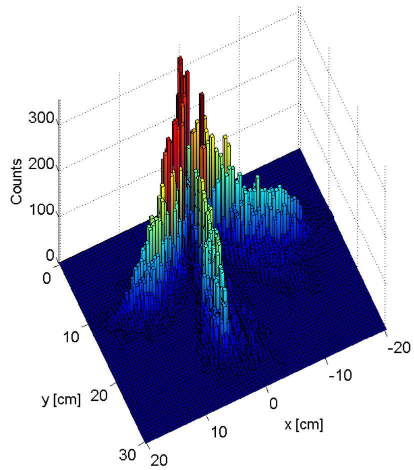


Fig. 6. Path dispersion as time elapses. All primitives start at the origin, and are largely dispersed after the 3 sec trial. The z axis counts how many paths are inside a particular grid square. Due to the selected grid size, some paths may appear more than once inside a square. CW paths curve to the left of the page, while CCW paths curve to the right.

the experimental averages shown with black thick curves in Fig. 7. Thus, constructing motion primitives that last longer is not meaningful; the variance in the experimental data becomes unacceptably high. Capturing the variability due to the stochasticity inherent to leg-ground interaction is beyond the scope of this paper. Preliminary considerations appear in [26], while a detailed account on a method for extending deterministic models to a stochastic regime for capturing the uncertainty within the model is reported in [27].

Table III contains the average final state of the robot for each primitive, while Table IV presents the target and average observed values for path curvatures and final orientations. Both averages are very close to their target values, although individual paths may deviate significantly.

TABLE III
FINAL POSE (AVERAGE VALUES)

Type	x_G [cm]	y_G [cm]	θ [deg]
CW	12.928	11.307	-85.02
SL	3.246	23.044	-10.62
CCW	-14.183	11.960	90.90

TABLE IV
CURVATURE AND FINAL ORIENTATION (AVERAGE VALUES)

Type	Target Curvature [cm^{-1}]	Observed Curvature [cm^{-1}]	Target Orientation [deg]	Observed Orientation [deg]
CW	0.063	0.066	$\simeq -90^o$	-85.02
SL	≤ 0.01	0.009	$\simeq 0^o$	-10.62
CCW	0.063	0.068	$\simeq 90^o$	90.90

Finally, Fig. 7 shows 50 randomly selected paths (in magenta) for each primitive. The average of the whole data set of 250 paths is marked by black thick curves. Note that the platform tends to veer to the right (see Fig. 7(b)).

D. *Parameter Identification*

Let $\mathcal{W}_{\text{SL}}$, $\mathcal{W}_{\text{CW}}$, and $\mathcal{W}_{\text{CCW}}$ denote the collections of all experimental planar trajectories of the robot for the SL, CW, and CCW motion primitives, respectively, and let w denote an element in these sets. The averages of all elements for each set are marked with $w_{\text{SL}}^{\text{ave}}$, $w_{\text{CW}}^{\text{ave}}$, and $w_{\text{CCW}}^{\text{ave}}$.

The model parameters to be identified are included in

$$\zeta = [\bar{\phi}_1^{\text{td}}, \bar{\phi}_2^{\text{td}}, \bar{\phi}_3^{\text{td}}, \bar{\phi}_4^{\text{td}}, \bar{\phi}_1^{\text{lo}}, \bar{\phi}_2^{\text{lo}}, \bar{\phi}_3^{\text{lo}}, \bar{\phi}_4^{\text{lo}}, \dot{\bar{\phi}}_{\mathsf{RL}}] \ .$$

Superscript ($^{\text{n}}$) stands for "nominal," ($^{\text{td}}$) indicates the model's touchdown angles, while ($^{\text{lo}}$) denotes liftoff angles, and $\dot{\bar{\phi}}_{\mathsf{RL}}$ marks the leg angular velocity, assumed to be the same for both pairs. To achieve CW turns we allow only the left pair to be active, while right pair activation only leads to CCW turns. For SL paths, both pairs are active. As before, we set d to 13 cm—equal to the length of the actual platform—and l to 3 cm.

In order to identify the SFM parameters that result in trajectories that remain close to the experimental averages, we formulate a constrained least-squares optimization problem

$$\min_{\zeta \in Z} \|p_i(\zeta) - w_i^{\text{ave}}\|^2, \ i \in \{\text{SL}, \text{CW}, \text{CCW}\} \ , \tag{4}$$

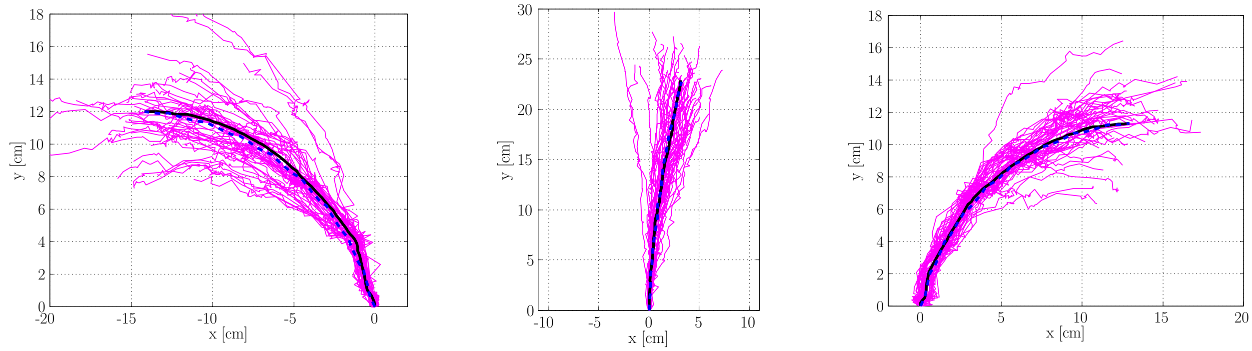
where $|| \cdot ||$ denotes the L_2 norm, and $p_i(\zeta)$ indicates the trajectories generated by the model parameterized according to the value of the vector ζ. Then, the nominal parameter vector ζ for each primitive is selected as the solution of (4).

The solution, $\hat{\zeta}$, enables the SFM to produce the trajectories shown in Fig. 7 with blue thick dashed curves. Table V contains the numerical values of the components of $\hat{\zeta}$, allowing SFM to capture *on average* the behavior of the system, as shown in Fig. 7. Due to their close matching with the experimental averages, the nominal SFM trajectories are used to calculate the values of the observed curvatures shown in Table IV by applying (3).

IV. NAVIGATION WITH THE TEMPLATE

Having recorded the motion primitives, we combine them to navigate in spaces populated with obstacles. To solve this problem, we employ a *rapidly exploring random tree* [24, Section 7.2.2] (RRT) solver that uses the constructed primitives in order to generate new vertices. The choice of RRT is primarily due to its popularity, proven experimental success, and the availability of off-the-shelf software implementing the basic algorithms.

The implementation considered here requires no tuning, but it does not allow for an early termination of a primitive. This means that a generated SL, CW, or CCW path is immediately discarded if it intersects with the obstacle region, and that only the end of each primitive can be used to define a new vertex. Because of this restriction, the algorithm may not always find a solution. This can be rectified by introducing a tuning parameter that shortens execution times. As the number of vertices increases, a path from the initial to the goal state is found more easily. The trade-off, however, is an increase in the computation time for the algorithm to find a solution.



(a) (b) (c)

Fig. 7. Experimental data for the primitives contained in Table I and the respective model output counterparts. The experimental average out of a total of 250 paths for each case is shown in black thick curves, and 50 randomly selected paths of the robot's geometric center are shown in magenta. The blue thick dashed curves depict the output of the model parameterized by the nominal parameter values of Table V.

TABLE V
IDENTIFIED MODEL PARAMETERS

Type	$\phi_1^{\mathrm{td,n}}$ [deg]	$\phi_2^{\mathrm{td,n}}$ [deg]	$\phi_3^{\mathrm{td,n}}$ [deg]	$\phi_4^{\mathrm{td,n}}$ [deg]	$\phi_1^{\mathrm{lo,n}}$ [deg]	$\phi_2^{\mathrm{lo,n}}$ [deg]	$\phi_3^{\mathrm{lo,n}}$ [deg]	$\phi_4^{\mathrm{lo,n}}$ [deg]	$\dot{\phi}_{\mathsf{RL}}$ [deg/sec]
CW	32.72	4.19	0	0	−55.28	−83.81	0	0	7.56
SL	56.72	45.84	68.18	70.47	−51.78	−62.66	−57.70	−55.41	9.00
CCW	0	0	26.28	1.37	0	0	−55.66	−80.57	8.65

V. RESULTS

We implement the solver in four illustrative simulation scenarios. The initial state varies for each case (see Table VI) but the environment (depicted in Fig. 8) stays the same. Similarly, the goal is set at $q_d = (x_d, y_d, \theta_d) = (220, 220, 0)$ [cm, cm, deg] and remains unchanged. Due to the constraints of the problem and the discretization induced by the solver, reaching exactly q_d is unlikely. Thus, we accept as successful all paths that end within a radius of 10 cm around (x_d, y_d), with a final orientation in the range between $-30°$ and $30°$.

TABLE VI
INITIAL STATES FOR CASE STUDY SIMULATIONS

Case	(a)	(b)	(c)	(d)
x_G [cm]	40	120	120	120
y_G [cm]	10	10	10	10
θ [deg]	0	0	−45	45

With respect to Fig. 8, the actual obstacles are marked in blue, while light gray is used to denote the portion of the configuration space where the boundary of the model touches, or crosses the boundary of the obstacle ("grown obstacle"). We use magenta to color the branches of the constructed trees, and we finally highlight in red the sequence of primitives that leads from the initial to a neighborhood of the desired state, marked with a large black circle.

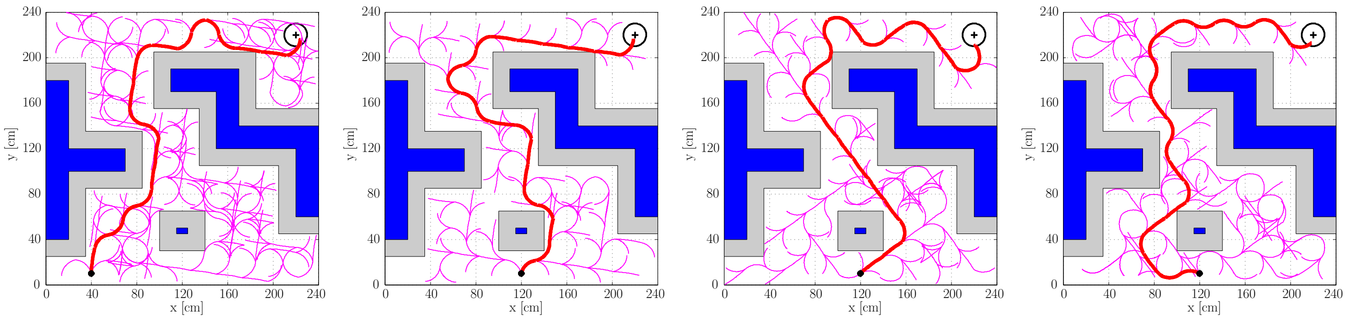
Figure 8 suggests that the solver finds a solution for all cases presented. Note, however, that paths may become very curvy toward the end due to the desired final orientation we have chosen. This behavior is exacerbated by the execution time of primitives being fixed. The solver always chooses the full primitive length, even if this may not be necessary. Allowing the primitives' execution time to vary can improve the quality of paths, but it will increase computational complexity and time required to find a solution. This trade-off is currently being investigated.

Our current work also addresses the experimental implementation and validation of the computed paths shown in Fig. 8 on the robot. This is, however, a challenging task since the high variability observed in the behavior of the robot renders successful completion of the desired paths unlikely; our previous work in [10] illustrates this point for a morphologically similar robot. For the robot considered here, the current low-level hardware does not allow for integration of on-board sensors (e.g. a compass and IMU) for state feedback, hindering the on-board implementation of feedback control policies.

VI. CONCLUSIONS

We demonstrate that the SFM template can be effectively used for navigation of miniature legged robots such as the OctoROACH, when the platform operates in a quasi-static fashion at low crawling speeds. Combining this kinematic template with a generic RRT solver yields reasonably good performance in terms of finding a path from an initial to a final state within cluttered environments. The paths produced respect the kinematic constraints of the robot, with minimal modifications and tuning.

Motion capabilities and mobility constraints are captured in the form of motion primitives, and encode basic motion behaviors: straight line, 90^o clockwise and counter-clockwise turns. The parameters of these primitives are given by the solution of a constrained optimization problem, formulated on a set of experimentally collected data from observed robot



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

Fig. 8. Navigation scenarios for different initial states. The first case differs in the position, while the last three start at the same point, but with different initial orientations. Obstacles are at the same location for all cases. Irrespective of the initial state though, we were always able to find a solution. The curvy final part in cases (a), (c), and (d) is due to the desired final orientation being set in the interval $[-30, 30]^o$; the effect can be remedied by letting the execution time vary.

paths. The returned solution minimizes the deviation of the model output from the average of the observed trajectories for a given type of behavior. Then, the solver uses these primitives as atomic motion behaviors for the system.

Extensions of the work presented here involve low-level hardware modifications, integration of sensors (range sensors, compass and IMU) to provide on-board state feedback for the experimental validation of the generated navigation strategies, investigation of the trade-off introduced by letting the primitives' execution time vary, and experimenting on different terrain types to identify the range of parameters which are robust to terrain changes.

References

[1] A. Yumaryanto, J. An, and S. Lee, "A cockroach-inspired hexapod robot actuated by LIPCA," in *IEEE Conf. on Robotics, Automation and Mechatronics*, 2006, pp. 1 – 6.

[2] J. Morrey, B. Lambrecht, A. Horchler, R. Ritzmann, and R. Quinn, "Highly Mobile and Robust Small Quadruped Robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2003, pp. 82–87.

[3] B. Lambrecht, A. D. Horchler, and R. Quinn, "A small, insect-inspired robot that runs and jumps," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1240 – 1245.

[4] D. Zarrouk, A. Pullin, N. Kohut, and R. S. Fearing, "STAR - a sprawl tuned autonomous robot," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 20 – 25.

[5] A. Stager, K. Karydis, and H. G. Tanner, "An Open-Source Passively Sprawling Miniature Legged Robot," in *IEEE Int. Conf. on Robotics and Automation*, 2015, to appear.

[6] S. Kim, J. E. Clark, and M. R. Cutkosky, "iSprawl: Design and Tuning for High-speed Autonomous Open-loop Running," *The International Journal of Robotics Research*, vol. 25, no. 9, pp. 903–912, Sep. 2006.

[7] R. Wood, S. Avadhanula, R. Sahai, E. Steltz, and R. S. Fearing, "Microrobot Design Using Fiber Reinforced Composites," *Journal of Mechanical Design*, vol. 130, no. 5, pp. 1 – 11, May 2008.

[8] A. M. Hoover, S. Burden, X.-Y. Fu, S. Sastry, and R. S. Fearing, "Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot," in *IEEE Int. Conf. on Biomedical Robotics and Biomechatronics*, 2010, pp. 869–876.

[9] A. Pullin, N. Kohut, D. Zarrouk, and R. S. Fearing, "Dynamic turning of 13 cm robot comparing tail and differential drive," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 5086–5093.

[10] K. Karydis, D. Zarrouk, I. Poulakakis, R. S. Fearing, and H. G. Tanner, "Planning with the STAR(s)," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 3033 – 3038.

[11] A. Mathis, J. Russell, T. Moore, J. Cohen, B. Satterfield, N. Kohut, X.-Y. Fu, and R. S. Fearing, "Autonomous navigation of a 5 gram crawling millirobot in a complex environment," in *Adaptive Mobile Robotics: 15th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2012, pp. 121 – 128.

[12] J. Schmitt and P. Holmes, "Mechanical models for insect locomotion: dynamics and stability in the horizontal plane I. Theory," *Biological Cybernetics*, vol. 83, no. 6, pp. 501–515, Nov. 2000.

[13] ——, "Mechanical models for insect locomotion: dynamics and stability in the horizontal plane - II. Application," *Biological Cybernetics*, vol. 83, no. 6, pp. 517–527, Nov. 2000.

[14] P. Holmes, R. J. Full, D. E. Koditschek, and J. Guckenheimer, "The Dynamics of Legged Locomotion: Models, Analyses, and Challenges," *SIAM Review*, vol. 48, no. 2, pp. 207–304, 2006.

[15] J. E. Seipel, P. J. Holmes, and R. J. Full, "Dynamics and Stability of Insect Locomotion: A Hexapedal Model for Horizontal Plane Motions," *Biological Cybernetics*, vol. 91, no. 2, pp. 76–90, 2004.

[16] D. Jindrich and R. Full, "Many-Legged Maneuverability: Dynamics of Turning in Hexapods," *The Journal of Experimental Biology*, vol. 202, pp. 1603 – 1623, Jun. 1999.

[17] J. Proctor and P. Holmes, "Steering by Transient Destabilization in Piecewise-Holonomic Models of Legged Locomotion," *Regular and Chaotic Dynamics*, vol. 13, no. 4, pp. 267–282, 2008.

[18] D. Zarrouk and R. S. Fearing, "Compliance-based dynamic steering for hexapods," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 3093 – 3098.

[19] A. J. Spence, S. Revzen, J. E. Seipel, C. Mullens, and R. J. Full, "Insects Running on Elastic Surfaces," *Journal of Experimental Biology*, vol. 213, pp. 1907 – 1920, 2010.

[20] F. Qian, T. Zhang, C. Li, P. Masarati, A. Hoover, P. Birkmeyer, A. Pullin, R. S. Fearing, and D. I. Goldman, "Walking and running on yielding and fluidizing ground," in *Robotics: Science and Systems*, Sydney, Australia, Jul. 2012.

[21] C. Li, A. M. Hoover, P. Birkmeyer, P. B. Umbanhowar, R. S. Fearing, and D. I. Goldman, "Systematic study of the performance of small robots on controlled laboratory substrates," in *SPIE Conf. on Micro- and Nanotechnology Sensors, Systems, and Applications II*, vol. 7679, 2010, pp. 76 790Z–13.

[22] K. Karydis, I. Poulakakis, and H. G. Tanner, "A switching kinematic model for an octapedal robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 507–512.

[23] K. Karydis, I. Poulakakis, and H. G. Tanner, "A Template Candidate for Miniature Legged Robots in Quasi-Static Motion," *Autonomous Robots*, vol. 38, no. 2, pp. 193 – 209, 2015.

[24] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.

[25] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[26] K. Karydis, Y. Liu, I. Poulakakis, and H. G. Tanner, "Probabilistic validation of a stochastic kinematic model for an eight-legged robot," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 2562 – 2567.

[27] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically Valid Stochastic Extensions of Deterministic Models for Systems with Uncertainty," *International Journal of Robotics Research*, 2015, accepted.