Central Pattern Generator Based Omnidirectional Locomotion for Quadrupedal Robotics

Michael J. Kuhlman[1], *Student Member, IEEE*, Joe Hays[2]
Donald Sofge[3], *Member, IEEE*, and Satyandra K. Gupta[4], *Senior Member, IEEE*

***Abstract*— Trajectory generation for quadruped robots is a challenging task since they are underactuated systems which must balance using sensory feedback and satisfy ground contact constraints. There is a substantial body of evidence that many animals use central pattern generators (CPGs) for generating joint trajectories and regulation through sensory feedback. However, CPG models formulated in the joint space do not explicitly formulate or account for ground contact constraints, especially during turning which can introduce foot slip in the gait. Task-based CPGs offer several advantages in that they can explicitly satisfy ground contact constraints, and we suggest an approach to generating foot velocity controls in the body frame to enable omnidirectional locomotion.**

I. Introduction

Legged robots such as quadrupeds have great potential for navigating rough terrain that other autonomous vehicles cannot. Many animals are able to quickly negotiate rough terrain such as steep mountain slopes. However, legged robots are often relegated to the confines of a laboratory setting where either the ground is flat, solid, and devoid of obstacles, or the environment is known beforehand and accurate global pose information is available. Motion planning techniques can be used to generate collision free paths when the environment is known in advance. To reduce the dimensionality of the planning problem, a high level motion planner can generate paths for the base of the robot to follow, while foot placements are planned online which maximize the stability margin from the zero moment point. This approach was used effectively on the Little Dog platform [1]. However, such approaches assume the robot has access to terrain maps, which are constructed offline and require access to global pose information by means of a motion capture system.

To get these robots out of the laboratory and into unknown field environments, planning collision free paths without external sensing is of utmost importance. Large uncertainties in sensing capabilities and especially global pose make it difficult to perfectly negotiate obstacles in a collision free manner. As Kalakrishnan et al. note, compliance is critical for interaction in uncertain environments [1]. Further, reflex behaviors are effective in providing quick stabilizing maneuvers when interacting with unknown terrain [2]. In order to navigate in rough terrain in GPS-denied environments, we propose having the robot base follow a desired collision free path which is planned in local coordinates while footholds are generated reactively using a central pattern generator (CPG). This approach requires *omnidirectional locomotion* (the ability to start, stop, and move in any direction with a desired twist imparted on the body) should the vehicle deviate off course and must return to its intended path.

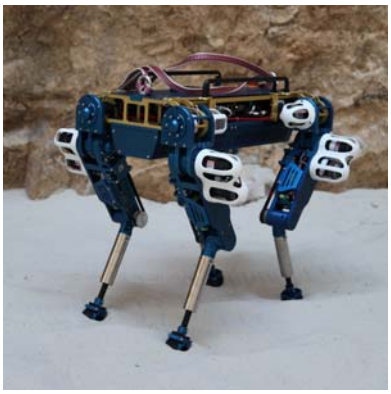


Fig. 1. The Allegro Dog quadruped platform

There is substantial evidence that many animals (mammalian quadrupeds, bipeds, lamprey) have CPGs in the spinal cord that assist in the generation of gaits [3]. CPGs have been used frequently in robotics [4], for quadruped robots in particular, to generate joint trajectories (gait sequences) and incorporate sensory feedback to stabilize locomotion [5], [6] . However (with notable exceptions [7], [8]), the bulk of research has focused on design of steady state forward gaits. The majority of CPG models, when they do specify a turning gait, execute turning maneuvers by changing the joint bias of the hip joints requiring specialized arm geometry. This approach has the further limitation that it does not explicitly formulate ground contact constraints, which can introduce foot slip in the gait. While most CPGs generate joint trajectories directly, Barasuol et al. have demonstrated using task space CPG-like coupled oscillators to control the HyQ platform [9]. The result is that fewer oscillators are required and direct approaches exist for enforcing ground contact constraints.

[1]M.J. Kuhlman is with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA and the Laboratory for Autonomous Systems Research, Naval Research Laboratory, Washington, DC 20375, USA mkuhlman@umd.edu

[2]J. Hays is with the Spacecraft Engineering Department, Naval Research Laboratory, Washington, DC 20375, USA joe.hays@nrl.navy.mil

[3]D. Sofge is with Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375, USA donald.sofge@nrl.navy.mil

[4]S.K. Gupta is with the Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA skgupta@umd.edu

II. Hardware Description

This work used the Allegro Dog robot which is a commercial-off-the-shelf quadruped robot from SimLabs Co., Ltd. (see Figure 1). The system is 450mm in length, 240mm wide, stands at 500mm in height and weighs 20kg. Each leg has actuated revolute joints for the hip-roll, hip-pitch, and knee-pitch degrees-of-freedom and a passive spring-loaded prismatic joint below the knee (totaling to twelve actuated and four passive joints). Each revolute joint is controlled with a 90W Maxon brushless DC motor with an 86:1 ceramic planetary gearhead and is sensed directly at the motor with a 1000 counts per turn (CPT) incremental magnetic encoder. Homing switches are used during initialization to find the mechanical zero position of each joint. Motor controllers close a current loop at 1kHz. Encoder values are sent to and current commands are received from the Allegro Dog embedded controller over a CAN bus at 500 Hz.

The base body has a CH-Robotics UM6 inertial measurement unit (IMU) that provides body orientation, angular velocities, and linear accelerations. This IMU updates internal state at 500 Hz and transmits data to the Allegro Dog embedded controller at about 70 Hz through a TTL-USB converter.

Power is provided externally through a tether, or onboard with 24V/4500mAh batteries. The Allegro Dog embedded controller has both wired and wireless Ethernet capabilities. All commands to and state updates from the Allegro Dog embedded computer are communicated over Ethernet using the Lightweight Communications and Marshalling LCM library [10].

III. Task-based Central Pattern Generator

We use a task-based CPG based which is an extension of the coupled oscillators in [9]. This CPG consists of a network of coupled modified Hopf Oscillators and a filtered system output that can be used to directly modify the foot ground velocities when in stance mode. Define $\mathbf{x}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ to be the task position of the ith foot where the total number of feet is $N = 4$. Combining all the task positions into a single $3 \times N$ matrix yields $X = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ Let $\mathbf{x}_{p_0,i}$ be the center position of the ith foot oscillator's limit cycle. The displacement about the center position is denoted as $\bar{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_{p_0,i} + \Delta\mathbf{x}_i$. In this case $\Delta X = [\Delta\mathbf{x}_1, \cdots, \Delta\mathbf{x}_N] = 0$. The equations of motion of the CPG are in (1):

$$
\begin{aligned}
\dot{x}_{c,i} &= \alpha\Big(1 - \frac{4\bar{x}_i^2}{L_s^2} - \frac{\bar{z}_i^2}{H_s^2}\Big)\bar{x}_i + \frac{\omega_i L_s}{2H_s}\bar{z}_i \\
\dot{y}_{c,i} &= -\beta\bar{y}_i \\
\dot{z}_{c,i} &= \gamma\Big(1 - \frac{4\bar{x}_i^2}{L_s^2} - \frac{\bar{z}_i^2}{H_s^2}\Big)\bar{z}_i - \frac{2H_s}{\omega_i L_s}\bar{x}_i + \sum_j k\mathbb{C}_{ij}\bar{z}_j \\
\omega_i &= \eta\pi\frac{V_{f,des}}{L_s}\big(\frac{D_f}{1-D_f}\sigma_{c1,i}(\bar{z}_{c,i}) + \sigma_{c2,i}(\bar{z}_{c,i})\big)
\end{aligned}
\tag{1}
$$

Define the instantaneous phase dependent angular frequency ω_i depending on the leg swing/stance phase. Also note parameters: gains $\alpha, \beta, \gamma > 0$, stride length L_s, step height H_s, swing/stance duty factor D_f and desired forward velocity $V_{f,des}$. $\mathbb{C}_{ij}$ is the coupling matrix defining foot phase relationships and are defined in [9]. Gaits such as walk, trot, and bound can be characterized by D_f and $\mathbb{C}_{ij}$. k is a scalar parameter to regulate the strength of the coupling. The gating functions $\sigma_{c1,i}(\bar{z}_{c,i}) = (e^{-b\bar{z}_i} + 1)^{-1}$ and $\sigma_{c2,i}(\bar{z}_{c,i}) = (e^{b\bar{z}_i} + 1)^{-1}$ are sigmoids which enable a smooth transition between the swing angular frequency and stance angular frequency. Note that the z component is used to identify whether the foot is in swing or stance mode.

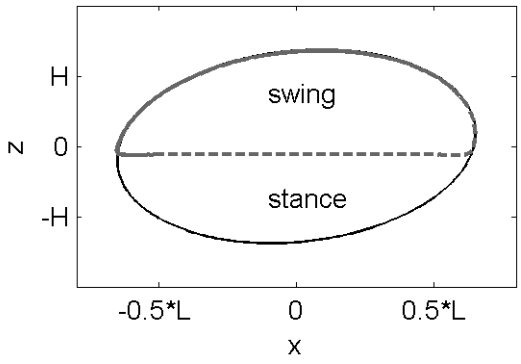


Fig. 2. Limit cycle of an example foot position controlled by the CPG. The solid black line is the limit cycle of the canonical CPG (1) while the gray dotted line the filtered output (2), which is smoothly clipped at $z = z_{td,i} = -0.1H_s$.

The filtered output dynamics are as follows:

$$
\begin{aligned}
\dot{\mathbf{x}}_{f,i} &= \big(\dot{\mathbf{x}}_{c,i} + K_c(\mathbf{x}_{c,i} - \mathbf{x}_{f,i})\big)\sigma_{f1,i}(\bar{z}_{c,i}) \\
&\quad -V_i\sigma_{f2,i}(\bar{z}_{c,i}) \\
\sigma_{f1,i}(\bar{z}_{c,i}) &= (e^{-b(\bar{z}_{c,i} - z_{td,i})} + 1)^{-1} \\
\sigma_{f2,i}(\bar{z}_{c,i}) &= (e^{b(\bar{z}_{c,i} - z_{td,i})} + 1)^{-1}
\end{aligned}
\tag{2}
$$

The use of the filtered output and gating functions $\sigma_{f1,i}$ and $\sigma_{f2,i}$ permit the smooth mixing of different task kinematics for whether the leg is in swing or in stance mode. When in swing mode, the filter output tracks the canonical CPG. When the feet are in stance mode, the task kinematics are defined by $V_i = [V_{i,x}, V_{i,y}, V_{i,x}]' \in \mathbb{R}^3$ which can be computed to enable omnidirectional locomotion. The step depth parameter $z_{td,i}$ is a controllable input for smoothly clipping the ellipse (See Fig. 2). However, one must be careful to ensure that the canonical CPG (1) and filtered output (2) trajectories are well matched to minimize aberrant behavior.

IV. Extensions to the CPG to enable Omnidirectional Locomotion

We extend the CPG defined in (1) and (2) to be capable of omnidirectional locomotion. We modify the dynamics of the CPG to generate smooth trajectories that execute the following actions:

1) The ability to start and stop. This requires adding an additional control input which adds a supercritical bifurcation point in the Hopf Oscillator. See Sec. IV-A.

2) Change the direction of the angular velocity to enable forward/reverse locomotion. See Sec. IV-B.
3) The ability to march in place for load testing, turn in place maneuvers and diagnostic purposes. See Sec. IV-B.
4) Automatically generate turning and crab gaits by generating a body twist controller to generate foot velocities that reproduce the desired locomotion. See Sec. IV-C.

The modified dynamics are:

$$\begin{aligned}
\dot{x}_{c,i} &= \alpha\Big(\mu - \frac{4\bar{x}_i^2}{L_s^2} - \frac{\bar{z}_i^2}{H_s^2}\Big)\bar{x}_i + \eta\frac{\omega_i L_s}{2H_s}\bar{z}_i \qquad (3)\\
\dot{y}_{c,i} &= -\beta\bar{y}_i\\
\dot{z}_{c,i} &= \gamma\Big(\mu - \frac{4\bar{x}_i^2}{L_s^2} - \frac{\bar{z}_i^2}{H_s^2}\Big)\bar{z}_i - \eta\frac{2H_s}{\omega_i L_s}\bar{x}_i + \sum_j k\mathbb{C}_{ij}\bar{z}_j\\
\omega_i &= \eta\pi\tfrac{V_{f,des}}{L_s}\big(\tfrac{D_f}{1-D_f}\sigma_{c1,i}(\bar{z}_{c,i}) + \sigma_{c2,i}(\bar{z}_{c,i})\big)
\end{aligned}$$

Where $\mu, \eta \in \{-1, 1\}$ are control inputs to regulate locomotion behavior. Filter dynamics are only modified for march-in-place.

A. Start and Stop

Previous work [7] has modified the Hopf oscillator to contain a supercritical Hopf bifurcation, which enables the oscillator to enter two different modes:

- $\mu = -1$: discrete motion to a user-selectable stable equilibrium point
- $\mu = 1$: rhythmic locomotion tending to a limit cycle

The effect of switching μ on the oscillator's performance can be seen in Fig. 3. However, merely modifying the canonical CPG's dynamics is not sufficient to have start/stop gait transitions as tracking issues will occur between the canonical system and the filtered output.

In addition, ΔX can be used to change the stable equilibrium point of the oscillator. When in stop mode, ΔX should be selected such that $X_{p_0} + \Delta X$ is within the attractive basin and will quickly converge to the limit cycle of the desired gait. Otherwise, $\Delta X = 0$.

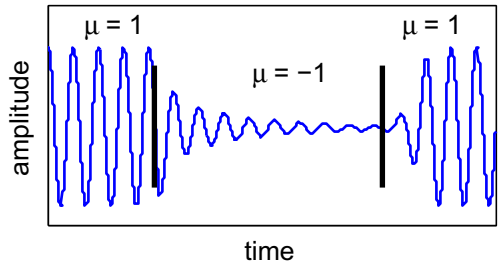
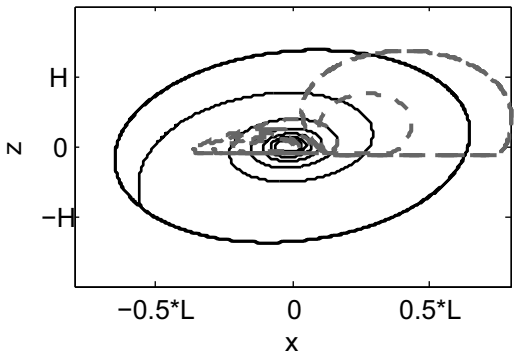


Fig. 3. The x component of a Hopf oscillator changing between start and stop modes by switching control input μ

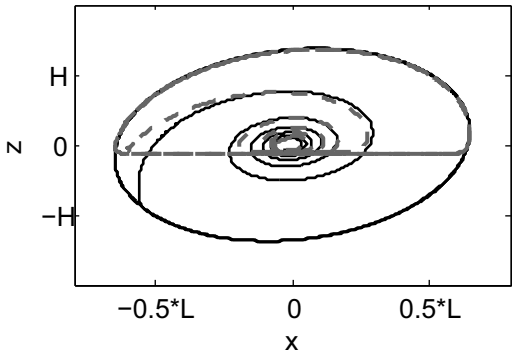
While angular velocity is preserved in the Hopf oscillator during shutdown, the translational velocity with respect to the ground plane is not preserved as the amplitude of the oscillator changes. One can imagine this being the result of shrinking the diameter of a wheel in a differential drive robot. Despite maintaining angular rate, the ground velocity will decrease. By observing that along the limit cycle of the oscillator when $\mu = 1$, $\frac{4\bar{x}^2_{\infty,i}}{L_s^2} + \frac{\bar{z}^2_{\infty,i}}{H_s^2} = 1$, one can scale $V_{f,des}$ and estimate the CPG's instantaneous forward velocity V_f of the canonical CPG given the current state, then average the result of all the legs so that the canonical CPG and filtered output are matched. See (4):

$$V_f = \frac{1}{N}\sum_{i=1}^{N}\frac{V_{f,des}}{L_s}\sqrt{4\bar{x}_i^2 + \bar{z}_i^2\frac{L_s^2}{H_s^2}} \qquad (4)$$

Fig. 4 demonstrates how setting $V_i = [-V_{f,des}, 0, 0]'$ causes tracking issues between the canonical CPG and the filtered output, while $V_i = [-V_f, 0, 0]'$ improves tracking performance of the filtered output.



(a) Having fixed V_f during startup and shutdown sequences causing significant tracking issues in the filtered output.



(b) Proper scaling of V_f using (4) enables the filtered output to properly track the canonical CPG

Fig. 4. Tracking issues between the canonical CPG and filtered output arise during startup and shutdown sequences. Using (4) to account for the instantaneous forward speed of the CPG corrects for this effect.

B. Forward/Reverse and March In Place

To switch between forward and reverse locomotion by modifying the original canonical CPG (1), define angular direction η such that $\eta = 1$ for forward locomotion and $\eta = -1$ for reverse locomotion.

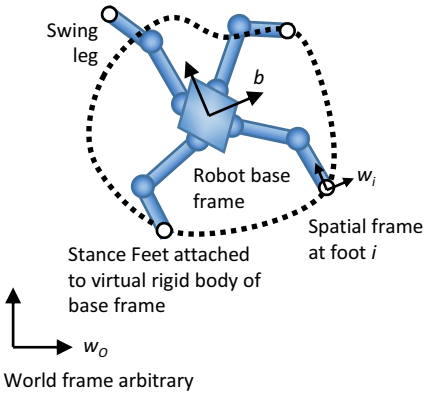


Fig. 5. Schematic representation of the coordinate frames and the virtual rigid body used in the foot velocity controller. Imparting a twist or spatial velocity on the robot base will induce motion of the feet and must be actively canceled by the foot controller to prevent slip.

It is also possible to redefine the filter dynamics to enable a simple march in place. One can replace the $x_{f,i}$ component of the filter dynamics to have stable linear dynamics decaying to the foot center position $x_{p0,i}$ when the leg is in swing mode. Summarizing (2) as $\dot{\mathbf{x}}_{f,i} = [f_1, f_2, f_3]^{'}$, augment filter dynamics to be (5):

$$\dot{x}_{f,i} = \begin{cases} f_1 & \text{if } d \neq d_{march} \\ (x_{p0,i} - x_{f,i})\sigma_{f1,i}(\bar{z}_{c,i}) & \text{if } d = d_{march} \\ \quad + V_i \sigma_{f2,i}(\bar{z}_{c,i}) & \end{cases} \tag{5}$$

Where $d \in \mathcal{D}$ is a new control input specifying the hybrid state of the system (walk, stop, march, etc), which is explained further in Sec. IV-D. Passing V_i through the modified march dynamics (5) when the leg is in stance mode can allow the quadruped to rotate in place when using body twist control (See Sec. IV-C).

C. Foot Velocity Generation for Body Twist Control

Suppose the base of the robot is expected to generate motion constrained to a plane parallel to the ground plane with a given arbitrary twist command $\hat{\xi} \in se(2)$, with twist coordinates $\xi = (\xi_x, \xi_y, \xi_\omega)$, where ξ_x is the desired forward base velocity, ξ_y is the side step velocity (for crab walk), and ξ_ω is the desired turning rate along the z axis (upward). Using these coordinates, the twist is in (6):

$$\hat{\xi} = \hat{V}^b_{w_i b} = \begin{bmatrix} 0 & -\xi_\omega & \xi_x \\ \xi_\omega & 0 & \xi_y \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

Note that $se(2)$ is the Lie algebra of $SE(2)$ so $e^{\hat{\xi}t}$ is a homogeneous transform $\in SE(2)$ [11]. It is assumed that this twist command is given in the robot body's frame instead of any particular spatial frame (or inertial frame). The position of the foot should not move in the world frame or *any* spatial frame so the footholds do not slip by controlling the position/velocity of the feet with respect to the body. One spatial frame of interest is a spatial frame positioned at the foot and oriented as the robot base frame w_i in Fig. 5. However, consider a virtual rigid body where the point contacts of the feet in stance mode are in a fixed position with respect to the body frame b. The movement of the base will induce motion of the point contacts of the stance feet since they are a point on the rigid body. Further, assume the current task positions of the feet are fully inside the workspace of the robot and motions in all directions are admissible.

Define $\hat{V}^s_{w_i b} \in se(2)$ to be the *spatial velocity* of the base moving with respect to the spatial frame placed at foot i which is in stance mode (7). The desired body velocity $\hat{V}^b_{w_i b}$ can be transformed into the spatial velocity $\hat{V}^s_{w_i b}$ using the adjoint transform [11]:

$$\hat{V}^s_{w_i b} = \begin{bmatrix} 0 & -\xi_\omega & \xi_x - y_i \xi_\omega \\ \xi_\omega & 0 & \xi_y + x_i \xi_\omega \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

For there to be no slipping of the stance feet, we must counter the spatial velocity of the feet using control input V_i in (2) such that the no slip constraint (8) holds:

$$(\begin{bmatrix} \dot{x}_i & \dot{y}_i & 0 \end{bmatrix}^{'})^s = \hat{V}^s_{w_i b} p^s_i - \mathbf{v}_i = 0 \tag{8}$$

where $p^s_i = [x_i, y_i, 1]^{'} = [0, 0, 1]^{'}$ is the origin of spatial frame w_i placed at the foot location and $\mathbf{v}_i = [v_{i,x}, v_{i,y}, 0]^{'}$. The quantity $\hat{V}^s_{w_i b} p^s_i$ is the linear velocity of the point p^s_i induced by spatial velocity $\hat{V}^s_{w_i b}$ in the spatial frame. Note that $\mathbf{v}_i$ and (8) are in homogeneous coordinates while $V_i \in \mathbb{R}^3$ where we have the planar assumption $\dot{z}_i = V_{i,z} = 0$. Satisfying (8) requires that CPG control input V_i be of the following form (9):

$$V_i = \begin{bmatrix} v_{i,x} \\ v_{i,y} \\ 0 \end{bmatrix} = \begin{bmatrix} \xi_x - y_{f,i}\xi_\omega \\ \xi_y + x_{f,i}\xi_\omega \\ 0 \end{bmatrix} \tag{9}$$

Figures 6 and 7 highlight the task trajectories of the feet in the body frame when executing a turning maneuver. Considering the previously observed tracking issues in the start/stop motion primitives in Sec. IV-A, the desired twists are normalized with respect to the instantaneous forward velocity $||(\xi_x, \xi_y)|| = V_f$ so that both the canonical CPG and filtered output are closely matched during startup/shutdown sequences. This technique will generalize to twist controls in $se(3)$.

D. Summary: Hybrid State Definitions

We found that in modifying a CPG based gait controller, it is helpful to have the following hybrid state $d \in \mathcal{D}$ to control the rhythmic aspects of the CPG where $\mathcal{D} = \{d_{stop}, d_{forward}, d_{reverse}, d_{march}\}$. This enables the CPG to generate smooth transitions to start and stop walking without explicitly encoding transition sequences. Parameter configurations for each hybrid state are summarized in Table I. Note that the hybrid state is not assigned a particular body twist ξ. Each hybrid state can reproduce a variety of stable

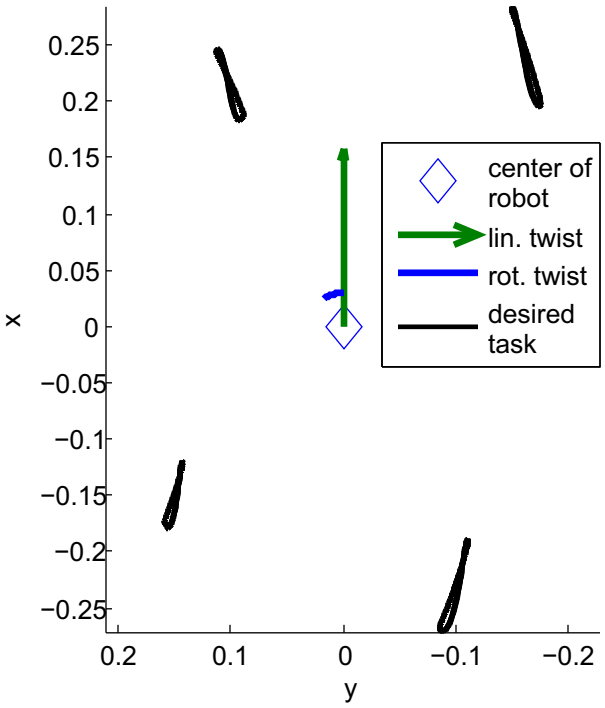


Fig. 6. Example reference foot trajectories generated by the CPG when a turn command is issued $\xi = (V_{f,des}, 0, \omega_{left})$. The desired body twist is displayed (linear component as a vector and rotational component as an arc) and foot task trajectories are shown in the body frame with a top-down view. Note that the Ackermann-like steering geometry emerges automatically from explicitly satisfying ground constraints.

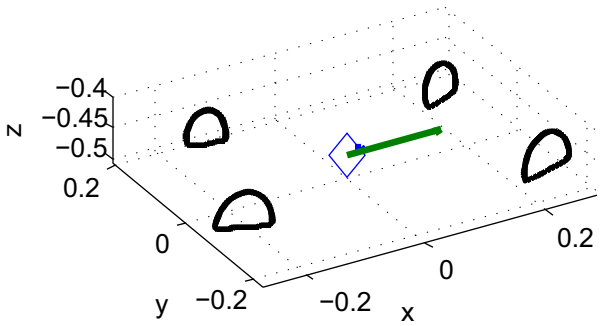


Fig. 7. Example reference foot trajectories generated by the CPG when a left turn command is issued $\xi = (V_{f,des}, 0, \omega_{left})$. This is viewed from a perspective view.

gaits for a particular set of $\xi \in se(2)$. This set is dependent on CPG parameters such as X_{p0}, L_s and H_s. For a quasistatic walk gait, criteria such as zero moment point could be used to verify the feasibility of a body twist for a given hybrid state.

state	d_{stop}	$d_{forward}$	$d_{reverse}$	d_{march}
(μ, η)	$(-1, \eta_{prev.})$	$(1, 1)$	$(1, -1)$	$(1, 1)$
ΔX	See Sec. IV-A	0	0	0

TABLE I

A SUMMARY OF HOW THE HYBRID STATE CONFIGURES CPG CONTROL VARIABLES.

V. EXPERIMENTAL SETUP AND RESULTS

Since the Allegro Dog does not have the requisite sensors to be able to measure or estimate the body's motion with respect to the ground plane, the body frame of the Dog was tracked using an external motion capture system (Vicon) which measures the Dog's pose at about 40 Hz. This data was in turn synchronized to LCM state messages regarding desired and actual joint trajectories, discrete motion commands (forward, reverse, etc) and the control twist which is internally generated by the foot velocity controller.

While the Vicon system returns poses in $SE(3)$, it would be of interest to extract instantaneous twists between poses. However, to numerically differentiate the pose directly would be highly sensitive to the noise present. Recursive Bayesian weighted regression is used for automatic (non-Gaussian) outlier detection with a forgetting factor $\lambda = 0.95$ [12].

The Allegro Dog uses a joint space trajectory tracking PD controller with fixed base (differential) closed loop inverse kinematics [13] and with base attitude compensation (kinematic adjustment in [9]).

Figure 8 compares the control twist starting from standstill to steady state forward locomotion $\xi = (0.11\frac{\text{m}}{\text{s}}, 0, 0)$, and the measured twist (angular rate from IMU, linear velocity from Vicon) in the body frame. Unmodeled dynamics of the foot-ground contacts cause the platform to sway, contributing to errors in the reproduction of the desired twist. Notice the lateral sway in v_y during startup. The oscillations in the yaw rate ω_{yaw} are unbiased so the robot's net motion over time is forward with oscillations in yaw angle being less than 3° peak to peak during the gait cycle. Further, the forward velocity $V_{f,des} = 0.11\frac{\text{m}}{\text{s}}$ is normalized with respect to the current state of the CPG using (4) hence the minor oscillations in v_x.

VI. CONCLUSIONS

A task-based CPG has been proposed that is capable of omnidirectional locomotion, including start/stop behavior and a foot velocity controller that generates stance feet velocities to impart any desired twist on the robot base. Preliminary results show that the foot controller is able to impart the desired twist on the body.

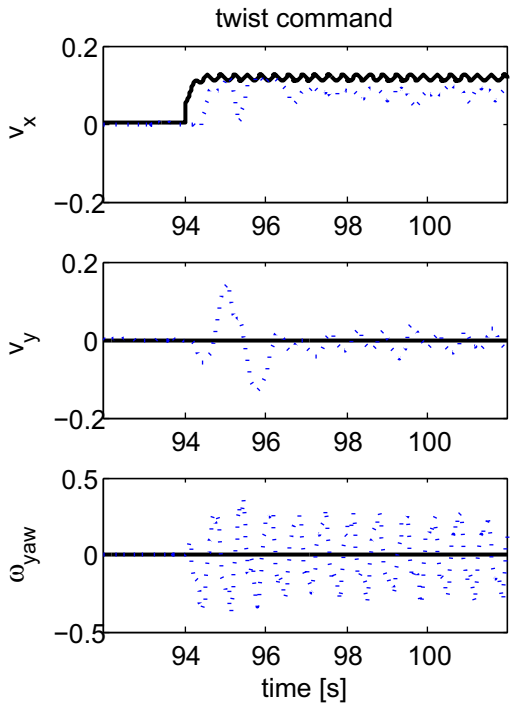


Fig. 8. Results. Twist command: standstill to forward twist command $\xi = (0.11\frac{\text{m}}{\text{s}}, 0, 0))$ (solid thick black), and measured twist (dotted blue/gray).

ACKNOWLEDGMENTS

This work was performed in part at the Naval Research Laboratory under JON 55-4843-A3, MEso-scale Robotic Locomotion InvestigatioN (MERLIN). The views, positions and conclusions expressed herein reflect only the authors' opinions and expressly do not reflect those of the US Department of Defense or the Naval Research Laboratory. S.K. Gupta's participation in this research was supported by National Science Foundation's Independent Research and Development program. The authors would like to thank Dan Wood from NRL for his insights, and the rest of the Laboratory for Autonomous Systems Research at NRL.

REFERENCES

[1] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[2] H. Kimura, Y. Fukuoka, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 475–490, 2007.

[3] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[4] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on cpg-inspired control models and system implementation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 3, pp. 441–456, March 2014.

[5] S. Gay, J. Santos-Victor, and A. Ijspeert, "Learning robot gait stability using neural networks as sensory feedback function for central pattern generators," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 194–201.

[6] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 819–824.

[7] C. P. Santos and V. Matos, "Cpg modulation for navigation and omnidirectional quadruped locomotion," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 912–927, 2012.

[8] V. Matos and C. P. Santos, "Omnidirectional locomotion in a quadruped robot: a cpg-based approach," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3392–3397.

[9] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 2554–2561.

[10] A. Huang, E. Olson, and D. Moore, "Lcm: Lightweight communications and marshalling," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 4057–4062.

[11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC press, 1994. [Online]. Available: http://www.cds.caltech.edu/~murray/mlswiki/

[12] J.-A. Ting, A. D'Souza, and S. Schaal, "Automatic outlier detection: A bayesian approach," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 2489–2494.

[13] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.