Stabilizing Task-based Omnidirectional Quadruped Locomotion with Virtual Model Control

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

***Abstract*— Quadruped locomotion offers significant advantages over wheeled locomotion for small mobile robots operating in challenging terrain. Central pattern generators (CPGs), as found in the neural circuitry of many animals, may be used to generate joint trajectories for quadruped robots. However, basic CPG-based trajectories do not explicitly consider ground contact constraints, a particular concern during turning maneuvers when foot slip is most likely to occur. An alternative approach proposed here is to use task-based CPGs such that ground contact constraints are enforced and foot velocities are explicitly controlled, resulting in stable omnidirectional locomotion. Further, incorporating Virtual Model Control with the task-based CPG trajectories improves the stability of the quadruped in hardware experiments.**

I. Introduction

Navigation through rough terrain remains a key challenge for legged robotic systems. Wheeled vehicles may be sufficient for operating on smooth surfaces, but often get stuck when attempting to navigate rough terrain. Legged locomotion offers far greater potential, but presents substantial challenges to path planning, trajectory generation, and control. First, legged platforms must indirectly control the unactuated degrees of freedom of the base using their legs and must maintain balance. On flat even surfaces, in fully modeled environments, and in simulations the contact points between the feet and the ground may be predicted. For example, the Little Dog platform [1] utilizes a high level motion planner to generate paths for the robot base in advance, while foot placement is computed in real time in order to maximize the stability margin based on the zero moment point. To do so either assumes that terrain maps already exist or the legged robot is carrying adequate terrain sensors onboard which is challenging for resource constrained robots.

As noted by Kalakrishnan [1], operation in uncertain environments requires compliance. Reflex behaviors [2] may be used to provide timely stabilization of the platform. Our approach to planning a path across rough terrain involves generating a collision-free path for the base while reactively determining foot placements using a central pattern generator (CPG). Should the vehicle deviate from its intended path, a control twist may be imparted on the base, allowing the platform to start, stop, and move in any direction as needed. We refer to this capability as *omnidirectional locomotion*. Many animals use CPGs to assist in gait generation [3]. CPGs have been used for quadruped robots [4] to generate joint trajectories incorporating sensory feedback to stabilize locomotion [5], [6]. Most such work, however, focuses on the use of CPGs to design steady state forward gaits rather than turning gaits. However, related works by Matos and Santos provides a notable exception [7], [8].

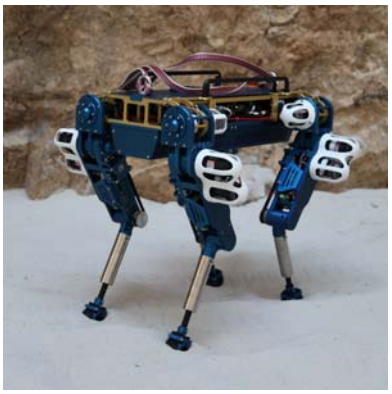


Fig. 1. The Allegro Dog quadruped platform

A common approach to executing turning gaits found in the CPG literature is to change the joint bias of the hip joints. This, however, requires specialized arm geometry, and has the further limitation that it does not explicitly formulate ground contact constraints, resulting in foot slippage. Barasuol et al. [9] demonstrated use of task space CPG-like coupled oscillators to control the HyQ platform, resulting in fewer oscillators. Furthermore, ground contact constraints may be enforced using direct means. In addition, Virtual Model Control (VMC) has been used to stabilize both bipeds [10] and quadrupeds [9], [11]. Further, controlling the attitude angles of a legged robot base with a neural network has been shown to improve stability of quadrupedal locomotion over rough terrain [5], suggesting that VMC will improve the stability of task-based CPG gaits.

II. Hardware Description

The Allegro Dog quadruped robot from SimLab Co., Ltd. (see Figure 1) weighs 20kg and each leg has 3 revolute

[1]M.J. Kuhlman is with the Department of Mechanical Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, 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

joints, each powered by a brushless DC motor with a 86:1 planetary gearhead. The Allegro Dog uses an embedded controller with a CAN bus to receive encoder values and send motor commands at 500 Hz. On the base is a CH-Robotics UM6 inertial measurement unit (IMU) which updates its internal state at 500 Hz and transmits data to the Allegro Dog embedded controller at 70 Hz. Force sensitive resistors (Interlink Electronics FSR 406) were added to the Allegro Dog for binary ground contact sensing. All commands to and state updates from the Allegro Dog embedded computer are communicated over wireless Ethernet and logged using the Lightweight Communications and Marshalling LCM library [12]. For more information about the configuration of the Allegro Dog, see [13].

III. Task-based Central Pattern Generator for Omnidirectional Locomotion

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 $\Delta X = [\Delta\mathbf{x}_1, \cdots, \Delta\mathbf{x}_N] = 0$. We extend the CPG defined in [9] which consists of a canonical system (1) and filtered output (2). 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.III-A.
2) Change the direction of the angular velocity to enable forward/reverse locomotion. See Sec. III-B.
3) The ability to march in place for load testing and turn in place maneuvers. See Sec. III-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. III-C.

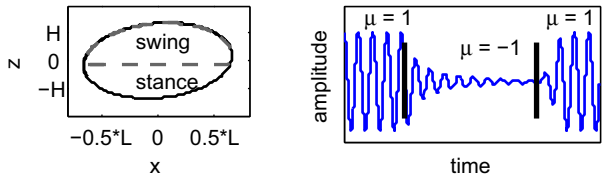
The modified canonical system dynamics are, (1):

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

Where $\mu, \eta \in \{-1, 1\}$ are control inputs to regulate locomotion behavior. 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 $\mathbb{C}_{ij}$. 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. The filtered output dynamics are as follows, (2):

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

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 $V_i = [V_{i,x}, V_{i,y}, V_{i,z}]^{\mathrm{T}} \in \mathbb{R}^3$ which can be computed to enable omnidirectional locomotion. Note that $(\cdot)^{\mathrm{T}}$ denotes the transpose. The step depth parameter $z_{td,i}$ is a controllable input for smoothly clipping the ellipse (See Fig. 2(a)). $z_{td,i}$ can either be set to a constant value or can incorporate contact sensor feedback. However, one must be careful to ensure that the canonical CPG (1) and filtered output (2) trajectories are well matched to minimize aberrant behavior.



(a) typical CPG limit cycle (b) start/stop motion sequence

Fig. 2. (a): 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 $z = z_{td,i} = -0.1H_s$. (b): The x component of a Hopf oscillator changing between start and stop modes by switching control input μ.

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

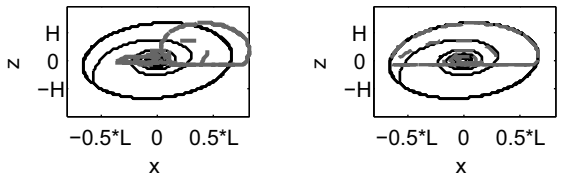
The effect of switching μ on the oscillator's performance can be seen in Fig. 2(b). However, merely modifying the canonical CPG's dynamics is not sufficient to have start/stop gait transitions as tracking issues will occur between the

canonical system and the filtered output. Note that (1) has an equilibrium at $X = X_{p_0}$ regardless of the value of μ. Therefore, if one wishes to successfully start after stopping, one must select $\Delta X \neq \mathbf{0}$ during the stop sequence such that $X_{p_0} + \Delta X$ is within the attractive basin of the limit cycle of the desired gait and will start back up again.

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 (3):

$$V_f = \frac{1}{N}\sum_{i=1}^{N} \frac{V_{f,des}}{L_s}\sqrt{4\bar{x}_i^2 + \bar{z}_i^2\frac{L_s^2}{H_s^2}} \tag{3}$$

Fig. 3 demonstrates how setting $V_i = [-V_{f,des}, 0, 0]^{\mathrm{T}}$ causes tracking issues between the canonical CPG and the filtered output, while $V_i = [-V_f, 0, 0]^{\mathrm{T}}$ improves tracking performance of the filtered output during shutdown.



(a) fixed $V_f = V_{f,des}$ (b) Proper scaling of V_f using (3)

Fig. 3. Tracking issues between the canonical CPG and filtered output arise during startup and shutdown sequences. Using (3) to account for the instantaneous forward speed of the CPG corrects for the mismatch seen in 3(a).

B. Forward/Reverse and March In Place

To switch between forward and reverse locomotion, define angular direction η such that $\eta = 1$ for forward locomotion and $\eta = -1$ for reverse locomotion.

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 $\dot{\mathbf{x}}_{f,i} = [f_1, f_2, f_3]^{\mathrm{T}}$, augment filter dynamics to be (4):

$$\dot{x}_{f,i} = \begin{cases} f_1 & \text{if } d \neq d_{march} \\ (x_{p0,i} - x_{f,i})\sigma_{f1,i}(\bar{z}_{c,i}) & \text{if } d = d_{march} \\ \quad + V_i\sigma_{f2,i}(\bar{z}_{c,i}) & \end{cases} \tag{4}$$

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. III-E. Passing V_i through the modified march dynamics (4) when the leg is in stance mode can allow the quadruped to rotate in place when using body twist control (See Sec. III-C).

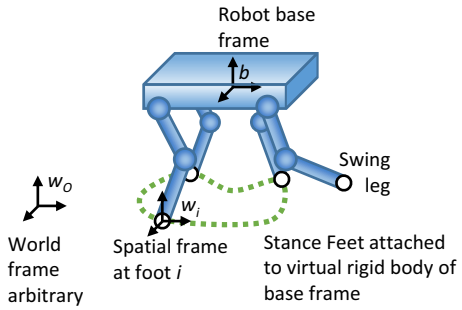


Fig. 4. Schematic representation of the coordinate frames and the virtual rigid body used in the foot velocity controller. Imparting a twist or spatial velocity on the robot base will induce motion of the feet and must be actively canceled by the foot controller to prevent slip.

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 body 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 (5):

$$\hat{\xi} = \hat{V}^b_{w_i b} = \begin{bmatrix} 0 & -\xi_\omega & \xi_x \\ \xi_\omega & 0 & \xi_y \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

Note that $se(2)$ is the Lie algebra of $SE(2)$ so $e^{\hat{\xi}t}$ is a homogeneous transform $\in SE(2)$ [14]. It is assumed that this twist command is given in the robot's base 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 base. One spatial frame of interest is a spatial frame positioned at the foot and oriented as the robot base frame w_i in Fig. 4. 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 base 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 (6). 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 [14]:

$$\hat{V}^s_{w_i b} = \begin{bmatrix} 0 & -\xi_\omega & \xi_x - y_i\xi_\omega \\ \xi_\omega & 0 & \xi_y + x_i\xi_\omega \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

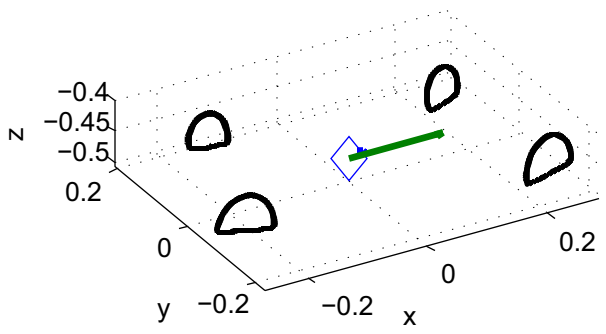


Fig. 5. Example reference foot trajectories generated by the CPG when a left turn command is issued $u = (V_{f,des}, 0, \omega_{left})$. The desired body twist is displayed (linear component as a vector and rotational component as an arc) and foot task trajectories are shown in the base frame and is viewed from a perspective view.

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 (7) holds:

$$(\begin{bmatrix}\dot{x}_i & \dot{y}_i & 0\end{bmatrix}^{\mathrm{T}})^s = \hat{V}^s_{w_i b} p^s_i - \mathbf{v}_i = 0 \tag{7}$$

where $p^s_i = [x_i, y_i, 1]^{\mathrm{T}} = [0, 0, 1]^{\mathrm{T}}$ is the origin of spatial frame w_i placed at the location of foot i and $\mathbf{v}_i = [v_{i,x}, v_{i,y}, 0]^{\mathrm{T}}$. 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 (7) 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 (7) requires that CPG control input V_i be of the following form (8):

$$V_i = \begin{bmatrix} v_{i,x} \\ v_{i,y} \\ 0 \end{bmatrix} = \begin{bmatrix} \xi_x - y_{f,i}\xi_\omega \\ \xi_y + x_{f,i}\xi_\omega \\ 0 \end{bmatrix} \tag{8}$$

D. Integration of Body Twist Controller with CPG

Note that with the CPG due to start and stop condition, the end user or automated planner does not have direct control over the instantaneous body twist. Suppose the user has a desired body twist with twist coordinates $u = (u_x, u_y, u_\omega)$. The CPG then in turn computes instantaneous body twist coordinates ξ considering the previously observed tracking issues in the start/stop motion primitives in Sec. III-A. Setting $\xi_x = V_f$ for forward locomotion will ensure that both the canonical CPG and filtered output are closely matched during startup/shutdown sequences. Fig. 5 highlights the task trajectories of the feet in the base frame when executing a turning maneuver. Injecting additional twist coordinates such as turning and sidestep into the CPG is possible. The described body twist controller will generalize to twist controls in $se(3)$.

E. 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 $\hat{\xi}$. Each hybrid state can reproduce a variety of stable gaits for a particular set of $\hat{\xi} \in se(2)$. This set is dependent on CPG parameters such as X_{p0}, L_s, H_s and system dynamics. 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. Note that η_{prev} is the previously assigned state of η before the hybrid state switched to d_{stop}.

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

TABLE I

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

IV. VIRTUAL MODEL CONTROL

In order to reduce the oscillations in the base pitch and roll to improve gait stability, we use Virtual Model Control (VMC) similar to the one used on the HyQ platform [9]. The VMC will use the quadruped's actuated joints to impose virtual wrenches on the base, stabilizing motion.

Define floating base configuration $\mathbf{q} = \begin{bmatrix} T_b^{\mathrm{T}} & \mathbf{q}_r^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathcal{Q}$ where T_b is the position and orientation of the base using Euler-Angle parameterization with respect to the world frame: $T_b = \begin{bmatrix} \mathbf{x}_b^{\mathrm{T}} & \theta_{roll} & \theta_{pitch} & \theta_{yaw} \end{bmatrix}^{\mathrm{T}}$.

It holds that feet in contact with the ground should not move with respect to the spatial frame. Define the constraint Jacobian of the floating base kinematics map $\mathbf{J}_c$ [15]. The floating base ground contact constraint can be formulated as follows (9):

$$\begin{bmatrix} \dot{\mathbf{x}}^{s'}_{f,s_1}, \ldots, \dot{\mathbf{x}}^{s'}_{f,s_i}, \ldots, \dot{\mathbf{x}}^{s'}_{f,s_{N_c}} \end{bmatrix}^{\mathrm{T}} = \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \tag{9}$$

Where $\dot{\mathbf{x}}^{s'}_{f,s_i}$ is in the horizontal frame (see [15]), $\{s_i\}_{i=1}^{N_c}$ consists of the feet indices which are in contact with the ground. VMC for base attitude control imposes a virtual wrench on the unactuated base of the platform using the null space projection matrix of the constraint Jacobian transpose $\mathbf{J}_c^{\mathrm{T}}$ to compute desired actuator joint torques τ_{VMC} to reproduce the desired wrench on the base $\mathbf{F}_b$ (10) [9]:

$$\tau_{VMC} = S(\mathbf{I} - \mathbf{J}_c^{\mathrm{T}} \mathbf{J}_c^{+'})\mathbf{F}_b \tag{10}$$

$F_b = \begin{bmatrix} \mathbf{f}_b^{\mathrm{T}} & \tau_b^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^6$ is the desired wrench (force and moment) on the base. S is a selection matrix which selects the actuated degrees of freedom of the configuration

space $\mathcal{Q}$. Because the degrees of the base are unactuated, (10) uses the right *weighted* pseudoinverse, where $\mathbf{J}_c^+ = W^{-1}\mathbf{J}_c^{+'}(\mathbf{J}_c W^{-1}\mathbf{J}_c^{+'})^{-1}$. $W \neq \mathbf{I}$ is a weighting matrix biasing the weighted pseudoinverse away from solutions whose base velocities are not reproducible by the joint actuators.

For initial testing, $\mathbf{F}_b$ is computed using a virtual model controller imposing PD setpoint regulation on the pitch and roll angles of the base, $(\theta_{roll,des}, \theta_{pitch,des}) = (0°, 0°)$ (11) (the remaining elements of $\mathbf{F}_b$ are zero):

$$\tau_{b,roll} = K_{p,roll}\theta_{roll} - K_{D,roll}\dot{\theta}_{roll} \quad (11)$$
$$\tau_{b,pitch} = K_{p,pitch}\theta_{pitch} - K_{D,pitch}\dot{\theta}_{pitch}$$

Where $\theta_{roll}, \theta_{pitch}, \dot{\theta}_{roll}, \dot{\theta}_{pitch}$ are sensed by the robot base's IMU.

V. Experimental Setup and Results

We wish to see how VMC improves system performance, both in improving the ability for the Allegro Dog to reproduce a desired body twist, and improving stability by minimizing the base attitude angles. For $t < t_0$, the Allegro Dog starts each trial at the stop position (Fig. 6). System state data is recorded for $T = 5$ seconds after a forward command $V_{f,des}$ is issued for all $t \geq t_0$. The root mean squared error of an arbitrary signal $w(t)$ defined on the interval $t \in [t_0, t_0+T]$ is $w_{RMSE} = \sqrt{\frac{1}{T}\int_{t_0}^{t_0+T}(w(t) - w_0(t))^2 dt}$ where $w(t)$ is the measured signal. $w_0(t)$ is the desired or nominal signal and t_0 is the time that the forward command is issued. A total of $P = 6$ startup sequences with VMC disabled, and similarly $P = 6$ startup sequences with VMC enabled were recorded. The median RMSE values are reported for a given signal.

The following CPG parameters were used for the Allegro Dog. For all hardware experiments, an intermittent trot gait was selected. The following parameters were used: $\mathbb{C}$ as defined for a trot in [9], D_f = 0.6. L_s = 0.08 m, H_s = 0.03 m, and V_f = 0.2 $\frac{\text{m}}{\text{s}}$. The assigned forward velocities and stride lengths results in a nominal angular task frequency of 1.5 Hz. Further, α = 4, β = -20, γ = 4, and b = 225. For VMC, the following gains were selected: $K_{p,roll}$ = 320, $K_{p,pitch}$ = 640, and $K_{D,roll}$ = $K_{D,pitch}$ = 50. For tracking the task trajectories generated by the CPG, the Allegro Dog uses a joint space trajectory tracking PD controller with fixed base (differential) closed loop inverse kinematics [16].

A. Body Twist Generation

Since the Allegro Dog does not have the requisite sensors to be able to measure or estimate the base's motion with respect to the ground plane, the base 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 IMU data, motion commands (forward, reverse, etc.) and the control twist coordinates ξ which is internally generated by the CPG and foot velocity controller. To numerically differentiate the Vicon 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 $\lambda = 0.95$ [17]. The desired body twist coordinates ξ are therefore compared to the observed base twist coordinates $\tilde{\xi}$.

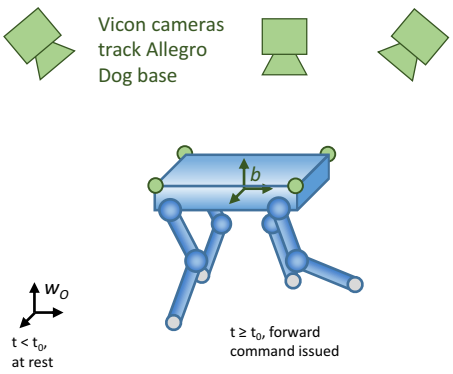


Fig. 6. Experimental setup for the Allegro Dog.

B. Virtual Model Control

To verify that VMC minimizes variations of the attitude angles (roll and pitch), we monitor the reported base attitude angles which are measured by the IMU during a start up motion sequence while trotting. For T seconds, the RMSE of the recorded attitude angles θ_{roll} and θ_{pitch} was computed with VMC both enabled and disabled.

C. Results

Figure 7 compares an example control twist ξ (black) starting from standstill to steady state forward locomotion $V_{f,des} = 0.2\frac{\text{m}}{\text{s}}$, and the measured body twist $\tilde{\xi}$ (dotted blue) from Vicon in the base frame comparing when VMC was disabled and when VMC was enabled. Median RMSE values of the measured twist coordinates are in Table II. 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 which generates ξ is normalized with respect to the current state of the CPG using (3) hence the minor oscillations in ξ_x. In addition, the median RMSE for each attitude angle during a start-up motion sequence are computed and are also summarized in Table II. This disturbance rejection ratio can be increased by increasing the stiffness of the virtual springs $K_{p,roll}$ and $K_{p,pitch}$.

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 a wide range of desired twist on the robot base. Hardware experiments show that a virtual model controller can reject roll disturbances by 55% and pitch disturbances by 30%. Looking forward, we would like to investigate techniques to minimize the effects of lateral disturbances

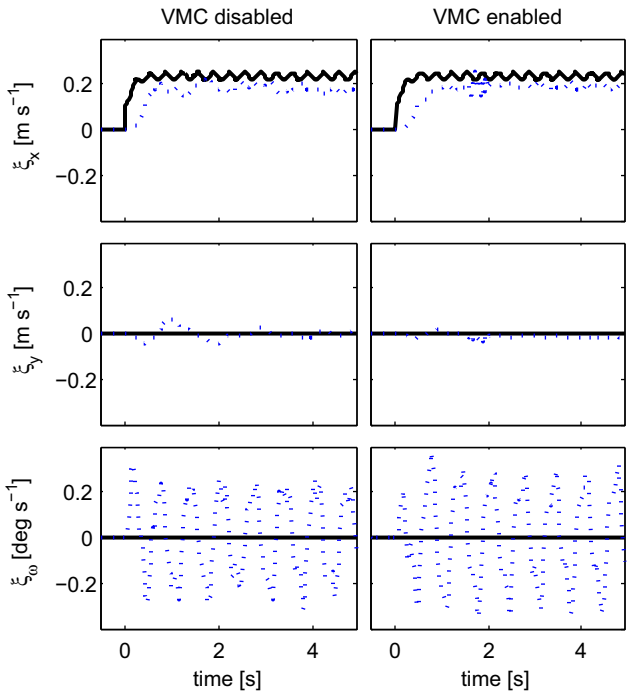


Fig. 7. An example of the body twist controller performance with VMC disabled (left) and VMC enabled (right). In both cases the standstill to forward command $V_{f,des}$ was issued: twist command $\xi(t)$ (solid thick black), and measured twist $\tilde{\xi}(t)$ (dotted blue/gray).

on the base. In addition, the use of reinforcement learning techniques to adapt CPG parameters when negotiating rough terrain is also of great interest. Finally, using a path planner in conjunction with terrain sensors and the body twist controller would enable autonomous navigation around impassable obstacles with reactive control such as VMC for stabilization during foot slip.

ACKNOWLEDGMENTS

This work was performed in part at the Naval Research Laboratory under NRL Work Unit 4843, "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 Victor Barasuol from Istituto Italiano di Tecnologia (IIT) for sharing his insights on VMC, Dan Wood from Praxis for his technical assistance on developing the control software and the rest of the Laboratory for Autonomous Systems Research at NRL.

attribute	without VMC	with VMC	performance improvement
$\tilde{\xi}_{x,RMSE}$	0.079	0.078	3%
$\tilde{\xi}_{y,RMSE}$	0.018	0.016	13%
$\tilde{\xi}_{\omega,RMSE}$	0.18	0.20	-14%
$\theta_{roll,RMSE}$	1.1°	0.5°	55%
$\theta_{pitch,RMSE}$	0.7°	0.5°	30%

TABLE II

MEDIAN RMSE VALUES FOR TWIST COORDINATES AND ATTITUDE ANGLES.

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," *IEEE Transactions onNeural Networks and Learning Systems*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 194–201.

[6] L. Righetti and A. J. Ijspeert, "Pattern Generators with Sensory Feedback for the Control of Quadruped Locomotion," in *IEEE International Conference on Robotics and Automation*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2554–2561.

[10] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual Model Control: An Intuitive Approach for Bipedal Locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[11] I. Havoutis, C. Semini, and D. G. Caldwell, "Virtual Model Control for Quadrupedal Trunk Stabilization," in *Dynamic Walking*, 2013.

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

[13] M. J. Kuhlman, J. Hays, D. Sofge, and S. K. Gupta, "Central Pattern Generator Based Omnidirectional Locomotion for Quadrupedal Robotics," in *Workshop on Real-time Motion Generation & Control Constraint-based Robot Programming, held at IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, September 18, 2014*, 2014.

[14] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC press, 1994.

[15] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse Kinematics with Floating Base and Constraints for Full Body Humanoid Robot Control," in *8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 22–27.

[16] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2008.

[17] J.-A. Ting, A. D'Souza, and S. Schaal, "Automatic Outlier Detection: A Bayesian Approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2489–2494.