frontiers in
NEUROROBOTICS

ORIGINAL RESEARCH ARTICLE
published: 02 October 2014
doi: 10.3389/fnbot.2014.00023

A novel approach to locomotion learning: Actor-Critic architecture using central pattern generators and dynamic motor primitives

***Cai Li*[1]**, *Robert Lowe*[1] *and Tom Ziemke*[1,2]**

[1] *Interaction Lab, School of Informatics, University of Skövde, Skövde, Sweden*
[2] *Department of Computer and Information Science, Linköping University, Linköping, Sweden*

Edited by:
Marc-Oliver Gewaltig, Ecole Polytechnique Fédérale de Lausanne, Switzerland

Reviewed by:
Tomohiro Shibata, Nara Institute of Science and Technology, Japan
Berat Denizdurduran, Ecole Polytechnique Fédérale de Lausanne, Switzerland

****Correspondence:***
Cai Li, School of Informatics, University of Skövde, Högskolevägen, 54128 Skövde, Sweden
e-mail: gauss.lee@his.se

In this article, we propose an architecture of a bio-inspired controller that addresses the problem of learning different locomotion gaits for different robot morphologies. The modeling objective is split into two: baseline motion modeling and dynamics adaptation. Baseline motion modeling aims to achieve fundamental functions of a certain type of locomotion and dynamics adaptation provides a "reshaping" function for adapting the baseline motion to desired motion. Based on this assumption, a three-layer architecture is developed using central pattern generators (CPGs, a bio-inspired locomotor center for the baseline motion) and dynamic motor primitives (DMPs, a model with universal "reshaping" functions). In this article, we use this architecture with the actor-critic algorithms for finding a good "reshaping" function. In order to demonstrate the learning power of the actor-critic based architecture, we tested it on two experiments: (1) learning to crawl on a humanoid and, (2) learning to gallop on a puppy robot. Two types of actor-critic algorithms (policy search and policy gradient) are compared in order to evaluate the advantages and disadvantages of different actor-critic based learning algorithms for different morphologies. Finally, based on the analysis of the experimental results, a generic view/architecture for locomotion learning is discussed in the conclusion.

Keywords: actor-critic, central pattern generators (CPG), reinforcement learning, locomotion control, NAO robot

1. INTRODUCTION

Locomotion modeling for robotics aims to endow a robot with the ability to propel itself in an environment. Traditional engineering approaches can model locomotion on a rigid-body robot with detailed modeling of a particular environment and body, such as zero moment point model and inverse kinematics model (for a review see Siciliano and Khatib, 2008). So traditional engineering approaches based locomotion models can work quite well in a constrained context but might have difficulties in adapting to different environments. However, a lot of modern robots are built on the basis of different animals, especially with distinct morphologies, e.g., fish robot (Marchese et al., 2013), worm robot (Ueno et al., 2014) and roboy humanoid (Pfeifer et al., 2014). None of their morphologies can be easily modeled. Therefore, bio-inspired approaches have been widely applied to model locomotion capabilities for such kind of robots (for a review see Ijspeert, 2008, Li, 2014), providing more flexibility focusing on the interaction with the environment and the emergence of different gaits. The dynamic systems theory was proposed by Thelen (1996) for emphasizing the importance of environmental interaction on the development of locomotor systems. The salient role of morphology (body) in the process of gait emergence was highlighted by Pfeifer and Bongard (2006). In order to model a locomotor system on a flexible body, we need to design (a) an interaction process involving the body and the environment; (b) a neural controller which can be adapted into a particular body following a design methodology and find a proper gait with this body in a particular environment. Therefore, in this article, key components (a body, an environment and a neural controller) of a locomotor system are highlighted and then a method for designing a robotic locomotion system adaptable to these components (based on learning) is proposed.

Figure 1 shows a schema in which three key components of a locomotor system interact. The neural controller is a structure that assimilates perceptual information from the environment and the state space of the body. It can be highly complex with brain-like functions such as memorization, perception (e.g., vision), learnability and so forth. The body is a physical medium through which neural systems contact the environment. In most robotics scenarios, it refers to a high DOF mechanical structure. The environment is not specifically modeled but considered to have a significant role affecting locomotion. In the mechanism described by **Figure 1**, the neural controller emits control signals to the physical body and receives the perceived changes in body states. The body receives the control signals and acts on the environment. Between the body and the environment, the contact force (e.g., supporting force, tangential force.) determines the quality of locomotion. Then the neural controller can also perceive information from the environment to evaluate its own behavior in order to send out better control signals. However,

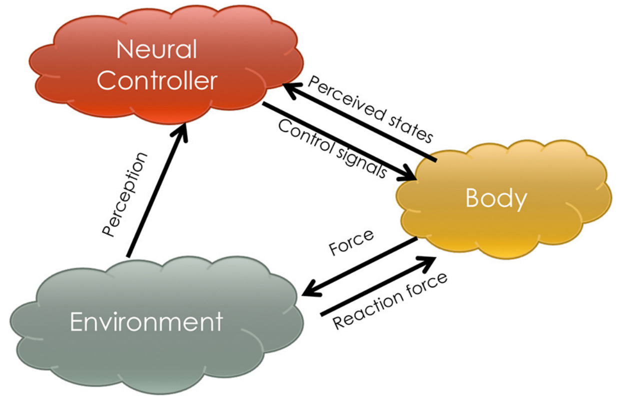

FIGURE 1 | This figure sketches the mechanism of three-component locomotion. The arrows indicate the functional information flow amongst the components, the neural controller, the body and the environment.

without the complexity of modeling an environment and a body like traditional engineering approaches, a methodology is developed in this article for designing a neural controller that (a) has the ability to learn in the above-mentioned three component interaction process and (b) can be utilized on different morphologies. We provide an example instantiation and demonstrate its generalizability by evaluating it on two robot morphologies.

In terms of neural controllers, choosing central pattern generators (CPGs) is one mainstream bio-inspired solution to modeling quadrupedal locomotion (Degallier et al., 2008; Harischandra et al., 2011; Zhao et al., 2012). CPGs are neural circuits which are located in the spinal cord of vertebrates and able to generate rhythmic movement without sensory feedback (Orlovsky et al., 1999; Latash, 2008). According to Grillner et al. (2005) and Rybak et al. (2006), CPGs receive input from a lot of brain parts (e.g., basal ganglia and brainstem) and muscles. This means CPGs not only provide strong adaptation capabilities to a certain type of locomotion but also are useful to explore fundamental locomotion principles for transferring animal locomotion capabilities to robots. In most work, CPGs are used as sensory-input-dependent neural networks of which the output is considered as a force or trajectory generator. According to Ijspeert et al. (2013), there are two modeling objectives for locomotion capabilities: One is a baseline behavior which contains core foundational patterns for a type of motor ability, for example the coordination of joints. After this is accomplished, the second concerns dynamic adaptation: how the baseline patterns can adapt to complex and dynamical changes pertaining to the environment or the physical body. On the basis of these two objectives, Section 2.1 will introduce the design of *baseline behavior* and *dynamic adaptation*.

The concept of *motor primitives* has been defined by researchers from biology as sets of force-fields generated by muscle synergies (Mussa-Ivaldi, 1999). It is also coined as "building blocks of movement generation" by Schaal et al. (2004) from the perspective of motor control. A very important function of CPGs for adaptation is the "reshaping" function which reshapes the output of neural circuits into the required one (Rybak et al., 2006; Ijspeert et al., 2013). Dynamic motor primitives (DMPs) can be used as a universal morphed oscillator which can turn rhythmic output into desired ones by constraining search space to a period of rhythmic input (for details, please refer to Section 2). Therefore, in our work, DMPs are used to model the function of dynamics adaptation, a representation of "reshaping" function. The model of DMPs is broadly used for motor learning (Peters, 2007; Kober et al., 2012) in a supervised learning algorithms since it has a good capability of reshaping the output to different dynamics with linear regression techniques. Therefore, in this article, DMPs are chosen as an interface because of its learnability with RL algorithms.

As for the mechanism of interaction, reinforcement learning (RL) is a particularly effective mechanism for searching proper "reshaping" functions in locomotion learning, especially for robotic applications (Nakamura et al., 2007; Endo et al., 2008; Li et al., 2013b). Nakamura et al. (2007) and Li et al. (2013b) demonstrated CPG architectures without a general "reshaping" function. The former developed a "reshaping" function based on a predefined sum of several variables and the latter proposed a "reshaping" function based on the limited sum of sensory input. Proper joint dynamics cannot be properly found with limited "reshaping" function. In our work, periodic DMPs are used to avoid this problem. Endo et al. (2008) aimed to learn leg trajectories of a biped based on a detailed model of the body (inverse kinematics). However, in our work, the body does not need to be modeled, which makes our model able to be used on different morphologies. The novelty of our method on the implementation level focus on the emergence of a certain type of gait in an interactive learning process provided by RL. From the perspective of neuroscience, RL also sketches a bio-inspired function for integrating different perceptual information, especially the actor-critic mechanism regarding basal ganglia (Wiering and van Otterlo, 2012) emphasized by Grillner et al. (2008) in their biological CPG structure. On this basis, we consider using CPGs in an actor-critic RL schema (CPG-Actor-Critic) a suitable approach which we adopt after accomplishing the design of the CPG architecture. Section 2.2 will introduce the use of CPG-Actor-Critic.

Based on the above-mentioned perspective, in order to test the neural controller, two experiments were conducted on two different-morphology robots, the NAO robot (rigid body) and ghost dog (soft body), for learning crawling and running, respectively (Section 3). After that, the process of locomotion capabilities emerging from baseline behaviors which serve as "prior knowledge" is analyzed and the detailed analysis of dynamics adaptation is shown in terms of joint dynamics. Finally in Section 4, the conclusion regarding a generic neural structure of locomotion learning is drawn for the purpose of implementing locomotion learning in a robot.

2. METHODS AND THEORIES

2.1. DESIGN OF THE CPG ARCHITECTURE

CPGs have been investigated to model locomotion in many robotic applications and there are also many existing CPG models inspired by the biological underpinnings of various levels and species (Ijspeert, 2008). In this respect, oscillator models are the ones commonly used and with a lot of advantages. Firstly, oscillator models can be easily modeled by ordinary differential equations (ODEs). Secondly, dynamics of oscillators come from the topology of couplings in the oscillator network, which is based on the well-established dynamic systems theory, especially for the symmetric topology (Golubitsky and Stewart, 2003). So oscillator models have a strong mathematical background. Moreover, the focus of oscillator models is on how phase difference and synchronization of different oscillators can be determined by the topology couplings or frequencies of the oscillator populations, rather than rhythm generation. So an oscillator in the model is not a model of a neuron but rather works like a complete oscillation center. If each DOF of a robotic body is controlled by at least one oscillator, in terms of Grillner et al's (Grillner, 1985) assumptions, oscillator models can handle the problem of how each DOF is coordinated with others within a high-DOF body. According to Grillner et al's research (Grillner et al., 2008), CPGs biologically are able to assimilate two functions: dynamics adaptation and posture control. If each DOF of the robotic joint controlled by an oscillator is considered as an adaptive limit cycle, dynamics adaptation is the function of reshaping the limit cycle and the posture control points to the ability of shifting the oscillation center. There are also many oscillators with such functionalities (Righetti and Ijspeert, 2006; Pouya et al., 2010).

Rybak et al. (2006) uncovered a possible biological anatomy of CPGs (**Figure 2** left). In this structure, the rhythm generator (RG) layer provides a primitive source of oscillatory signals. The pattern formation (PF) layer is a level on which all the RGs are mutually connected to form the phase-separated output. The dynamics adaptation (DA) represents the functions of motoneurons of which the output is sent directly to muscle fibers. In this layer, the output of PF layer is adapted into distinct dynamics in order to adapt to different environments or interactions. After reshaping the output of PF layer in DA layer, the RG itself turns out to be a "clocking" driver for CPGs. This three-layer architecture has been implemented to model walking behaviors (Li et al., 2013b; Nassour et al., 2013). This also matches the two objectives (Section 1) of locomotion modeling: from *baseline behaviors* to *dynamics adaptation*. RG and PF layers represent the architecture of baseline behavior. These two layers encode fundamental characteristics of one type of locomotion. For example, crawling is experimentally observed to be one type of locomotion featured by anti-phase movement of the ipsilateral limbs and in-phase movement of diagonal limbs, as coupling information (Righetti et al., 2008). The final layer is a layer of adapting baseline dynamics into desired locomotion.

2.1.1. The method of designing the baseline behavior

Golubitsky and Stewart (2003) propose an approach to designing the symmetric CPG topology based on the dynamic systems theory. A four-cell architecture (**Figure 3**) is widely used to coordinate the main joints (the joints attached to the main body, usually hip and shoulder joints) of a locomotor system (Degallier et al., 2008; Li et al., 2011). It is mathematically proved that this architecture can simulate the synchrony of different quadrupedal gaits, such as trot, walk, pace and gallop. If the couplings are changed, the transition amongst those gaits can be simulated too. On the other hand, as a minimal topology of CPGs, it can be extended to an eight-cell architecture by using zig-zag or cross coupling to generate all kinds of gaits for quadrupeds according to Golubitsky and Stewart (2003). Since all the quadrupedal animals have similar gaits and the four-cell architecture accounts for most such gaits (Righetti, 2008), it can be used to model the basic dynamics of each gait, including preliminary coupling, trajectory and frequency.

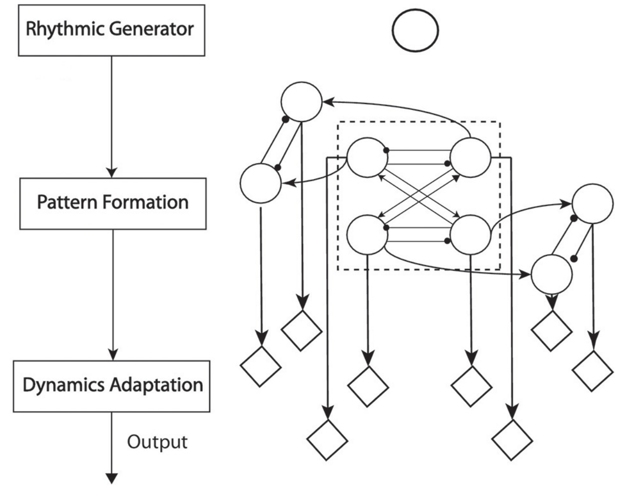

FIGURE 2 | Left: The functional structure of CPG anatomy; each block represents one-layer functionality. **Right**: The neural structure of CPG employed for crawling. The single circle above most represents the RG layer as an oscillator. The recurrent neural network composed of connected circles represents the function of the PF layer. The diamonds represent functions of the DA layer. Within the PF-layer network, the four-cell network in the dash-line frame controls the rhythms of pitch motion for Shoulders and Hips. The other four outside the dash-line frame control the roll motion of Shoulders and Hips. The arrow-head lines represent in-phase oscillation (the phase difference between two oscillators is 0 or 2π). The dot-head lines represent anti-phase oscillation (the phase difference between two oscillators is π).

Before using this architecture, the existence of stable periodic solutions has to be determined according to the H/K theorem (Golubitsky and Stewart, 2003; Righetti, 2008) (for details, please refer to Supplementary Material). As examples of the H/K theorem, regarding the two gaits we are concerned with in this article, we can mathematically prove that the stable periodic solutions exist. Assume the nodes in **Figure 3** control different joints respectively (1: left shoulder, 2: right shoulder, 3: left hip, 4: right hip. Knee and ankle joints are not controlled since it is not convenient for them to oscillate for a humanoid like NAO with big feet. All the other joints are synchronized with the correspondent joints controlled by the four-cell architectures, including roll and elbow joints), we can start to permute the symmetric characteristics of the architecture. In terms of crawling, a characteristic of crawling gaits is the anti-phase and in-phase relation of the ipsilateral limbs and diagonal limbs, respectively (for the detailed mathematical meanings of H and K group, please refer to

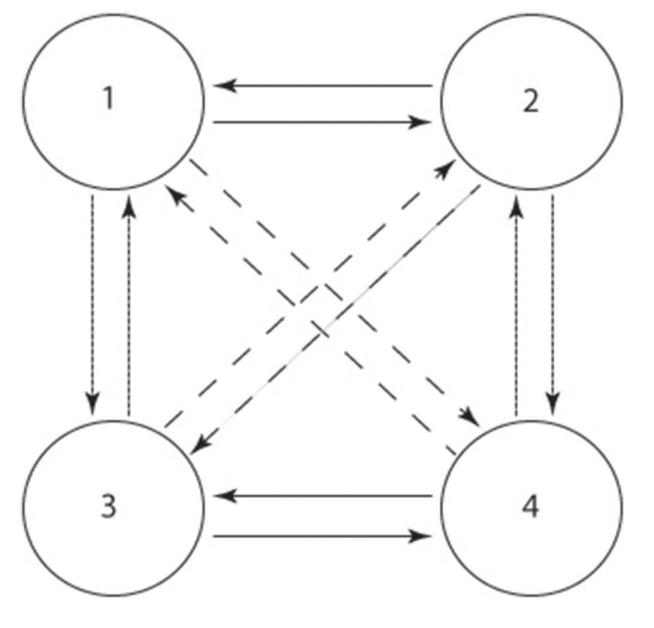

FIGURE 3 | The architecture of the four-cell CPG network. The condense dashed, dashed, and straight arrow indicate ipsilateral, diagonal and contralateral couplings.

Supplementary Material). So the spatial-temporal group H_c and spatial group K_c for crawling are:

$$H_c : \left\{\left((12), (34), \frac{T}{2}\right), \left((13), (24), \frac{T}{2}\right), ((14), (23), 0)\right\}$$
$$K_c : \left\{\left((14), (23), 0\right)\right\}$$

where T is the period of one signal and $\frac{T}{2}$ is the phase shift in each group. Obviously $H_c/K_c \cong Z_2$ which is cyclic (Righetti, 2008) (also see Supplementary Material) and K_c is an isotropy group. In terms of double-suspension gallop (front and rear feet are in phase respectively), the spatial-temporal group H_g and spatial group K_g of this gait are:

$$H_g : \left\{\left((12), (34), 0\right), \left((13), (24), a\right), \left((14), (23), a\right)\right\}$$
$$K_g : \left\{\left((12), (34), 0\right)\right\}$$

where a is the phase shift and $a \in [0, T]$. In the work described in this article, a is equal to 2. $H_g/K_g \cong Z_m$ ($m \gg 2$) which is cyclic too (For the proof please refer to Supplementary Material) and K_g is an isotropy group.

Within the four-cell CPG network, each node can be modeled by an oscillator (e.g., numerical oscillators Li et al., 2013b or phase oscillators Pouya et al., 2010). The advantage of using phase oscillators is that the phase shift can be explicitly specified. Therefore, in this article, a standard phase oscillator is chosen:

$$\dot{r}_i = a_i(R_i - r_i)$$
$$\dot{W}_i = 2\pi\omega_i + K_i$$
$$K_i = \sum_j w_{ji} \cdot r_j sin\left(W_j - W_i - P_{ji}\right)$$
$$\omega_i = \frac{\omega_{1i}}{e^{-100 \cdot Aex_i} + 1} + \frac{\omega_{2i}}{e^{100 \cdot Aex_i} + 1}$$
$$Aex_i = r_i \cdot sin\left(W_i + \frac{\pi}{2}\right)$$
$$A_i = r_i \cdot sin(W_i)$$

where A_i is the output of this phase oscillator and Aex_i is the frequency control output. r_i and W_i are the amplitude and phase variables respectively. ω_i is the frequency of the oscillator with ω_{1i} and ω_{2i} controlling the ascending and descending frequency. K_i is the connection term from the other oscillators to oscillator i. w_{ji} is the connection weight of from oscillator j to i. W_j is the phase of oscillator j and P_{ji} is the phase difference from oscillator j to i (For example, in the four-cell network formed based on the H/K theorem, the groups $((13), (24), \theta)$ (θ is the phase shift) are represented by setting $P_{31} = \theta$ (cell 1), $P_{42} = \theta$ (cell 2), $P_{13} = -\theta$ (cell 3) and $P_{24} = -\theta$ (cell 4), θ can be $\frac{T}{2} = \pi$ for crawling or $a = 2$ for galloping). a_i and R_i are the convergence rate and converged value of amplitude respectively. In our work, the parameter settings are as follows: $a_i = 50$, $R_i = 1.0$, $w_{1i} = w_{2i} = 1.0$.

In summary, corresponding to the RG and PF layer, a four-cell CPG network is utilized as a baseline motion generator to drive the motion of each joint (details are in **Figure 2**). This baseline motion generator has the capabilities to maintain structural stability according to group theory (Golubitsky and Stewart, 2003) and has been verified to generate basic patterns of both crawling and bipedal walking by adapting parameters w_{1i} and w_{2i} (Degallier et al., 2008; Li et al., 2011, 2012). Moreover, the prior knowledge about a specific gait is encoded in this baseline behavior generator to reduce the workload of gait learning. For example, taking advantage of the symmetric topology can reduce the number of DOFs which are to be learned/optimized so that the dimensions of further learning are lowered.

2.1.2. Design of the dynamics adaptation function

With a baseline behavior, in this section we will discuss how to adapt it into a mature gait. This needs an architecture which can reshape/shift the baseline dynamics to achieve DA. There are two situations for DA in practical implementation: the body without (proper) sensors and with *useful* sensors. In a lot of cases, because of the mechanical design, some robots do not have useful sensors for certain types of locomotion gait. For example, the NAO robot does not have pressure sensors for crawling. So a general approach is required for the situation whether there are proper sensors or not. In fact, DA is a trivial function which involves how different perceptual/sensory information (e.g., sensory feedback) contributes to reshaping the dynamics of each DOF. Since there is no systematic approach to finding a reshaping mechanism regarding to different sensory feedback and robots also have different sensor configurations, in this article, DA only focuses on an abstract basic mathematical framework to achieve the same function of reshaping the dynamics based on the baseline behavior in order to alter the trajectories/dynamics of each joint. Therefore, an architecture which can modify the joint movement is required. According to Ijspeert et al. (2013)[1], DMPs have two types of format: discrete and periodic. They differ in the forcing term f. In our work, since the task is to learn a rhythmic movement, the periodic DMPs are selected as a dynamics modifier and the mathematical model of periodic DMPs is:

$$\begin{aligned}
\tau \dot{z}_i &= \alpha\left(\beta(g_i - y_i) - z_i\right) + amp \cdot A_i + f \\
\tau \dot{y}_i &= z_i \\
f(W_i, p) &= \frac{\sum_{j=1}^{N} \psi_j v_j}{\sum_{j=1}^{N} \psi_j} p_i \qquad (1) \\
\psi_j &= exp\left(h_j(cos(W_i - c_j) - 1)\right) \qquad (2) \\
\tau \dot{g}_i &= \alpha_g(g_0 - g_i) \\
\alpha &= 8.0, \beta = \frac{\alpha}{4}, \alpha_g = \frac{\alpha}{2} \qquad (3) \\
c_j &\sim (0, 2\pi)
\end{aligned}$$

where z_i, y_i and g_i are the variables of the motor primitive. amp is the weight of correspondent input from PF layer and set to 27 which makes the output of motor primitives oscillate between −1 and 1. A_i is the "clocking" input from the baseline behavior (the PF layer). τ is the time constant which is equal to the period ($\frac{1}{\omega_i}$) of input A_i. f is the forcing term in which ψ_j are fixed basis functions, v_j are the weights and p_i is the amplitude which is equal to amp. $N = 50$ represents the number of basis functions. Using nonlinear arbitrary functions in f is a well-defined approach in machine learning (Bishop, 2006) for nonlinear regression and analogous to population coding in computational neuroscience (Dayan, 2005). In Equation (2), h_j is a constant equal to $2 \cdot N$ and W_i is the phase input from baseline behaviors. c_j is a vector containing N separations of the scope in $(0, 2\pi)$. g_0 is the anchor point ($g_0 = 0$). Equation group (3) guarantees the damping convergence of the DMPs.

Theoretically, the forcing term f above is used as a universal signal modifier. Assume $F = amp \cdot A_i + f$ is equal to the second-order equation based on the optimal signal (y_{op}):$\tau^2 \ddot{y}_{op} + \tau\alpha\dot{y}_{op} + \alpha\beta y_{op} - \alpha\beta g_i$, then substitute this equation to replace the F term, then we can rewrite the DMPs equation:

$$\dot{z}_i = \tau \ddot{y}_{op} + \frac{\alpha}{\tau}\left(\tau \dot{y}_{op} - z_i\right) + \frac{\alpha\beta}{\tau}(y_{op} - y_i), \tau \dot{y}_i = z_i \qquad (4)$$

According to the theory of morphed oscillation (Ajallooeian et al., 2013), Equation 4 is a second order morphed oscillator which can adapt the baseline behavior into any limit cycle shape. Then the task left is to figure out a mechanism for changing the parameters of DMPs to converge to a desired gait. Therefore, the DMP model is used as an interface for learning/adaptation.

In fact, DMPs are widely used to model discrete motor learning (Peters, 2007; Kober et al., 2012) and rhythmic movement (Ijspeert et al., 2013). In terms of periodic movement learning, Nakanishi et al. (2004) and Gams et al. (2009) employed demonstrated signals to learn motor primitives of rhythmic motion with local weighted regression. However, supervised learning might not always be the case for locomotion learning. Infants learn to crawl by interacting with the environment rather than being demonstrated how each joint moves dynamically (Clearfield, 2004; Kail and Cavanaugh, 2012). Locomotion learning based on reinforcement learning (RL) without demonstrated signals and motor primitives is also popular (Morimoto et al., 2005; Endo et al., 2008). However, as the motor primitives model has a good learnability, in this article, an approach of using motor primitives and RL for locomotion learning without demonstrated signals is proposed.

2.2. THE MECHANISM: CPG-ACTOR-CRITIC

The CPG-Actor-Citic architecture has been used for exploring and learning complex locomotion patterns for both bipeds (Nakamura et al., 2007; Endo et al., 2008; Li et al., 2013b) and quadrupeds (Kohl and Stone, 2004; Li et al., 2013a). Inspired by Grillner et al. (2008), the functions of CPG-Actor-Critic connects the layered architecture to an behavioral selection learning (RL) process in which the optimal parameters of CPGs are determined. The actor, by generating actions, explores the state space of the body and the critic evaluates the actions taken by observing rewards in order to send improved control signals to the body. In this article, two modern policy-focused RL techniques are used for this continuous-space learning problem of DMPs. One is episodic natural actor critic (eNAC) (Peters, 2007) and the other is policy learning by weighting exploration with returns (PoWER) (Kober et al., 2012). eNAC has been empirically demonstrated to be a "winner" algorithm for policy gradient approaches, outperforming FDG (Finite Definite Gradient) and VPG ("Vanilla" Policy Gradient) (Peters, 2007). Also, PoWER is a faster RL learning algorithm better than the other "family" members, such as eRWR (episodic Reward-Weighted Regression) (Kober et al., 2012). In this article, we also intend to implement and compare these two state-of-the-art algorithms on locomotion learning tasks using DMPs.

Generally speaking, in the parameter space for using DMPs, a policy-based actor is used to sample/explore the continuous action space. Equation 5 shows a time-variant actor used for DMPs.

$$\begin{aligned}
\boldsymbol{a} &= \boldsymbol{\theta}^T \boldsymbol{\psi_f}(\boldsymbol{x}, \boldsymbol{t}) + \boldsymbol{\epsilon_t} \qquad (5) \\
\epsilon_t^n &\sim N\left(0, (\sigma^n)^2\right)
\end{aligned}$$

where a is the output vector of an actor and $\boldsymbol{\theta}$ is the policy parameter vector reflecting the weights v_j in motor primitives. $\boldsymbol{\psi_f}$ is the vector of normalized basis functions of motor primitives. $\boldsymbol{\epsilon_t}$ is the gaussian exploration vector with standard deviation $\boldsymbol{\sigma}$ which contains the standard deviation for each basis function at time t and ϵ_t^n is the exploration for the nth basis function in $\boldsymbol{\epsilon_t}$. In most of cases, the $\boldsymbol{\sigma}$ is kept constant for all the parameters. It has been mathematically proved that both eNAC and PoWER using DMPs can also use an actor exploring only in the state-dependent parameter space (Shown in Equation 6) (Kober et al., 2012).

[1] In Ijspeert et al's work, they refer to "dynamic movement primitives" which is "dynamic motor primitives" in this article.

$$\boldsymbol{a} = \left(\boldsymbol{\theta}^T + \boldsymbol{\epsilon}_{\boldsymbol{t}}^T\right) \boldsymbol{\psi}_f(\boldsymbol{x}, \boldsymbol{t}),\ t = 1, 2, 3...T \tag{6}$$

$$\epsilon_t^n \sim N(0, (\sigma^n)^2)$$

According to Kober et al. (2012), the disadvantages of using state-independent exploration is: (1) Large variance in parameter updates. (2) The effect of perturbation could be washed out if exploration is too frequent as the system works like a low-pass filter. (3) It could possibly damage the completeness of system execution. On the other hand, the advantage of using state-dependent exploration in the actor is able to reduce the computational load by not executing a matrix multiplication $\boldsymbol{\theta}_{\boldsymbol{t}}^{\boldsymbol{T}} \boldsymbol{\psi}_f(\boldsymbol{s}, \boldsymbol{t})$. Moreover, when the DMPs are used in the case that each basis function is activated only once in one period, the exploration can be further simplified to be executed at the beginning by using a time-invariant exploration $\boldsymbol{\epsilon}_{\boldsymbol{0}}^T$. In the following section part, the update mechanism of the critic for each algorithm will be introduced.

2.2.1. Policy gradient approach

Policy gradient is a well-established method used to update the parameterized action space. Since the normal "vanilla" gradient suffers the slow learning rate, natural policy gradient is formed by adding a regularized term in normal gradient approach to force the update path to follow the steepest direction (Kober et al., 2012).

Since learning locomotion might be a repetitive task (Adolph et al., 2012), episodic natural actor critic (eNAC) using natural gradient is selected. NAC is proposed by Kakade (2001) and further developed and used in motor learning by Peters (2007); Peters and Schaal (2008). eNAC is mathematically constructed on the NAC approach and uses episodic exploration for each rollout. In the practical work, eNAC generates a sufficient number of rollouts in order to get the realization of gradient information around the current state. It transforms the traditional RL problem of solving the Bellman equation to an explorative process of linear regression using DMP basis functions. Assume there are H rollouts the eNAC algorithm generates for each update, then the update rule for the critic can be summarized as below (for a mathematical introduction to eNAC, please refer to Supplementary Material):

$$\begin{bmatrix} \boldsymbol{w} \\ J \end{bmatrix} = \left(\boldsymbol{\phi}\boldsymbol{\phi}^T\right)^{-1} \boldsymbol{\phi}\boldsymbol{R}.$$

$$\boldsymbol{\phi} = \left[\sum_{t=1}^{s} \alpha_t \bigtriangledown log^T(\pi^{\boldsymbol{\theta}}(\boldsymbol{u}_t|\boldsymbol{x}_t))\boldsymbol{w}, 1\right]_{1:H}^T$$

$$\boldsymbol{R} = \left[\sum_{t=1}^{T} \alpha_t r(\boldsymbol{x}_t, \boldsymbol{u}_t)\right]_{1:H}^T$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \boldsymbol{w}$$

where $1 : H$ represents H samplings within one trial (refer to details in the Algorithm). $\boldsymbol{\phi}$ is the basis matrix containing H basis vectors for H rollouts and constant 1 in it is used to determine the baseline J avoiding large-variance updates. α_t is the theoretical discounting factor. $\boldsymbol{R}$ is the average reward vector in which r is the instant reward (for the detailed eNAC proof, please refer to Peters, 2007). $\boldsymbol{w}$ is the estimated steepest gradient according to your sampling rollouts and used to update the parameter $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$ with α learning rate ($\alpha = 0.1$).

2.2.2. Expectation Maximization based policy search

Expectation maximization (EM) is a useful tool as a machine learning technique to find out the optimal solutions based on increasing the value of the lower bound for the cost function (Bishop, 2006). According to Kober et al. (2012), a critical drawback of policy-gradient is its difficulty in determining the learning rate and the unsteadiness to reward values. This is why usually EM-based policy approaches can converge faster than policy-gradient approaches. A normal EM-based policy search algorithm works to find out the $\boldsymbol{\theta}'$ which maximizes the lower bound $L(\boldsymbol{\theta}')$ on a cost function. Likewise, in order to find the maximum value of $L(\boldsymbol{\theta}')$, the derivative of it is set to zero (Equation 7, for the details please refer to Supplementary Material).

$$\partial_{\boldsymbol{\theta}'} L(\boldsymbol{\theta}') = 0 \tag{7}$$

With the actor in Equation 5, it can generate an algorithm called episodic reward weighted regression (eRWR). But in our work, authors are more interested in a more efficient algorithm derived from the actor using the Equation 6, the policy learning based on weighted exploration with returns (PoWER). The derivation of PoWER is given in Supplementary Material. The update rule of PoWER is:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \left(\sum_{t=1}^{T} \boldsymbol{W}(\boldsymbol{s}, \boldsymbol{t})\boldsymbol{Q}^{\pi}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{t})\right)^{-1} \left(\sum_{t=1}^{T} \boldsymbol{W}(\boldsymbol{s}, \boldsymbol{t})\boldsymbol{\epsilon}_{\boldsymbol{t}}\boldsymbol{Q}^{\pi}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{t})\right)$$

Where $\boldsymbol{W}(\boldsymbol{s}, \boldsymbol{t}) = \boldsymbol{\psi}(s, t)\boldsymbol{\psi}(s, t)^T(\boldsymbol{\psi}(s, t)^T\boldsymbol{\Sigma}\boldsymbol{\psi}(s, t))^{-1}$. $\boldsymbol{\epsilon}_{\boldsymbol{t}}$ is the exploration in Equation 6 and $\boldsymbol{Q}(s, \boldsymbol{a}, t)$ is the action-state function based on the policy π with exploration $\boldsymbol{\Sigma}$ which is a diagonal deviation matrix.

2.2.3. Continuous action space learning logic

In the above-mentioned algorithms, the convergence condition is set to be $\boldsymbol{\theta}' \approx \boldsymbol{\theta}$ which might be difficult to achieve when the reward function is not explicitly bounded as in the case of supervised RL with clear targets (Peters, 2007). Therefore, a general continuous action space learning logic is required to help us intuitively judge if the algorithm converges or not. Cacla (continuous action learning automaton) proposed by van Hasselt and Wiering (2007) and proved to outperform some typical RL methods (like SARSA, Q(λ), NAC and so on) (Hasselt, 2007), offers an update logic by exploring around the current state. If the value of the state increases after the action is taken, the update is executed. The schema of Cacla is shown in Algorithm 2.2.3. In the case of episodic learning, Monte-Carlo difference is used instead of typical temporal difference since the prediction value function $V(\boldsymbol{s'})$ for next future step $\boldsymbol{s'}$ is not explicitly observable (Jaakkola et al., 1995).

```
Algorithm 2.2.3

• Initialize the parameter $\boldsymbol{\theta}$ and state space $\boldsymbol{s}$
• Repeat:
    ○ Perform the exploration on policy $\pi(\boldsymbol{a}|\boldsymbol{\theta}, \boldsymbol{s})$ and generate actions for rollouts (the number of rollouts $H > 1$).
    ○ Calculate the value function $V$ difference between current state $\boldsymbol{s}$ and the future state $\boldsymbol{s'}$:
    ○ Value function estimation/approximation:
        1. Temporal difference: $\delta = r - \lambda V(\boldsymbol{s'}) + V(\boldsymbol{s})$, estimate $V(\boldsymbol{s'}) = V(\boldsymbol{s}) + \beta\delta$, where $r$ is the immediate reward and $\beta$ is the learning rate.
        2. Monte-Carlo difference for episodic learning: $\delta = R - V(\boldsymbol{s})$, estimate $V(\boldsymbol{s'}) = V(\boldsymbol{s}) + \beta\delta$, where $R$ is the reward for one episode.
    ○ Update judge: if $\delta > 0$: Update the policy toward the good actions with gradients.
• Until no update is executed as the algorithm cannot find any better solution any more.
```

Using Cacla as an update logic is very useful for determining the condition of update ($\delta > 0$) the termination of the algorithm for unbounded reward functions. In order to adapt eNAC and PoWER into the Cacla logic, we can modify the algorithm as follows (only the detailed modification is shown below neglecting the unchanged part):

```
• ....
• Repeat:
    ○ M trials each of which includes 10 rollouts (H=10), In each rollout, action is generated by $\boldsymbol{a} = \left(\boldsymbol{\theta}^T + \boldsymbol{\epsilon}_t^T\right)\boldsymbol{\psi}_f$ where $\boldsymbol{\epsilon_t} \sim N\left(0, \sigma^2 * \mathbf{I}\right)$ $(\sigma = 0.1)$ for t = 1, 2, 3. . . .s
    ○ Calculate the value function $V$ difference between current state $\boldsymbol{s}$ . . .
    ○ . . . .
    ○ Update judge: if $\delta > 0$: *eNAC* —- Calculate the gradient $g$:
          $$\begin{bmatrix} \mathbf{w} \\ J \end{bmatrix} = \left(\boldsymbol{\phi}\boldsymbol{\phi}^T\right)^{-1}\boldsymbol{\phi}\boldsymbol{R}.$$
      where $\boldsymbol{R} = [r_1, r_2, \cdots r_H]^T$ and $\boldsymbol{\phi} = [\Psi_1, \Psi_2, \cdots, \Psi_H]^T$. $r_i$ represents the accumulated reward in this episode $r_i = \sum_{t=1}^{s} reward_t$ and $\Psi_i = \sum_{t=1}^{s} \sigma^2(\boldsymbol{\epsilon_t}\boldsymbol{\psi})\boldsymbol{\psi}^T$. Then update $\boldsymbol{\theta'} = \boldsymbol{\theta} + \alpha\boldsymbol{w}$
      *PoWER* —- Calculate the update for each parameter $\theta_i$:
          $\theta_i' = \theta_i + \frac{E\left\{\sum_{t=1}^{T} \sigma_i^2 \epsilon_{i,t} Q^\pi\right\}}{\sum_{t=1}^{T} \sigma_i^2 Q^\pi}$
      where $Q^\pi$ is the action-state function and $\epsilon_{i,t}$ represents the output of an actor for parameter i at time t. $\sigma$ is the variance of the policy.
• ....
```

Generally speaking, in this section, a locomotion system with the theme "two systems (the baseline system and DMPs) and one mechanism" is introduced as a design methodology for CPG-Actor-Critic. The design of each system also might be flexibly replaceable by more sophisticated systems. For example, the baseline behavior can be a complicated coupled network of which each node stands for one advanced baseline motion generator (Buchli et al., 2006; Gams et al., 2009). In the next section, we will test this method on two different robotic platforms.

3. EXPERIMENTS AND ANALYSIS

In this section, two experiments are reported for two different purposes. Firstly, the experiment 1 is to test and verify the learnability of the DMPs based CPG-Actor-Critic by using eNAC only since DMPs are always used in the regression approach. In this experiment, standard crawling (Wikipedia, 2013) is learned on the NAO robot which does not have pressure sensors for crawling. Secondly, the whole system is transferred to the ghostdog robot, whose body is softer and more flexible than the NAO robot, to test the transferability of the system to distinct morphologies and compare the two algorithms, eNAC and PoWER, on the more challenging locomotion task (more likely to find differences in performance this way).

3.1. EXPERIMENT 1: LEARNING TO CRAWL

The objective of experiment 1 is to verify the capability of DMPs based CPG-Actor-Critic on limit-cycle reshaping and postural control. There are two sub-experiments: One is to test the learnability of the CPG architecture by using a generic "reshaping" mechanism [Equation (1) with the same targeted posture-the same spineline angle]. The other is to test if the generic motor primitives can also adjust the joint posture (shifting centers of limit cycles) under the condition that the posture control reward is set to two different targets.

With the eNAC algorithm above-mentioned, the robot is able to explore the dynamics of each joint on its own according to the specific reward function. **Figure 4** shows the standard crawling (crawling on knees and hands Wikipedia, 2013). The main joints controlled by CPGs are the ones located at the hip and shoulder. The elbow oscillates with the rhythms of the shoulder pitch. Since crawling is left-and-right symmetric (Righetti et al., 2008), the number of degrees of freedom (DOFs) can be reduced from 8 (left and right joints) to 4 (left or right joints only). Therefore, the parameters for standard-crawling learning are $4 \cdot 50 = 200$ (where 4 is the number of DOFs and 50 is that of basis functions). From our previous work Li et al. (2013a), the move distance and spineline angle (**Figure 4**) are two significant factors to evaluate the quality of crawling behaviors. Accordingly, in the CPG-Actor-Critic architecture, the reward function is composed of two terms ($r_{distance}$ and r_{angle}) as two evaluation landmarks for the above-mentioned two variables:

$$\begin{aligned} r_{reward} &= r_{distance} + r_{angle} \\ r_{distance} &= exp(\frac{D}{2}) - 1 \\ r_{angle} &= exp(e) - 1 \\ with\ e &= N(\mathbf{x_0}, \sigma = 0.02) \end{aligned} \tag{8}$$

where D is the distance the robot crawls every episode. e is a gaussian distribution with the center x_0 and variance σ. Using e enables maintaining the posture of standard crawling without learning some extreme postures (Li et al., 2013a). In the case of infants learning to crawl, this function works similar to parents' hands adjusting or holding up the infant's body when (s)he is crawling.

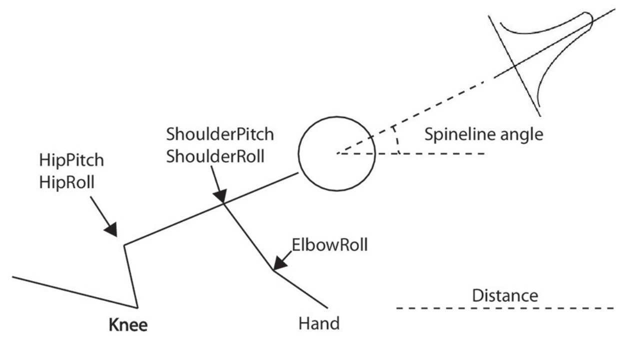

FIGURE 4 | The standard crawling posture on knees and hands and the main joints controlled by CPGs. The distance and spineline angle indicate the quality of crawling. The spineline angle is controlled by a gaussian function.

3.1.1. Sub-experiment: Reshaping-mechanism test

In this Experiment, with the average spineline angle fixed at 30° ($x_0 = 1.05$), the robot learns to crawl in three independent runs and finally converges to three different results by balancing the distance maximization and posture maintenance. Every experiment starts with the same initial posture with ($x_0 = 1.08$, approximately 28°) and performs a pre-learning non-crawling behavior with no crawling distance (**Figure 5**). However, after learning, the standard crawling emerges through the interaction amongst the CPG-Actor-Critic architecture, the humanoid body and the environment (For the detailed performance, please refer to the video Li, 2013a). Interestingly, the three learning trials converge with similar smooth reward curves (**Figure 6**) but different results (**Figure 7**).

In order to clearly investigate the reasons of the formation of crawling, the joint dynamics are shown separately in **Figure 7**. Since the standard crawling is a whole-body motion, the CPG-Actor-Critic autonomously decides how to adapt the motion of each joint. The adaptive changes of pitch joints for shoulders and hips focus on the adjustment of their amplitudes (**Figures 7A,B**). Especially, the HipPitch joint tends to swing more backward so that robot can crawl forward with more force. Interestingly, extracting from the results in our experiments (**Figures 7C,D**), the significant factor determining whether the robot can crawl forward properly is the roll motion. Not only are the amplitudes of roll joints in shoulders and hips statistically adapted, but

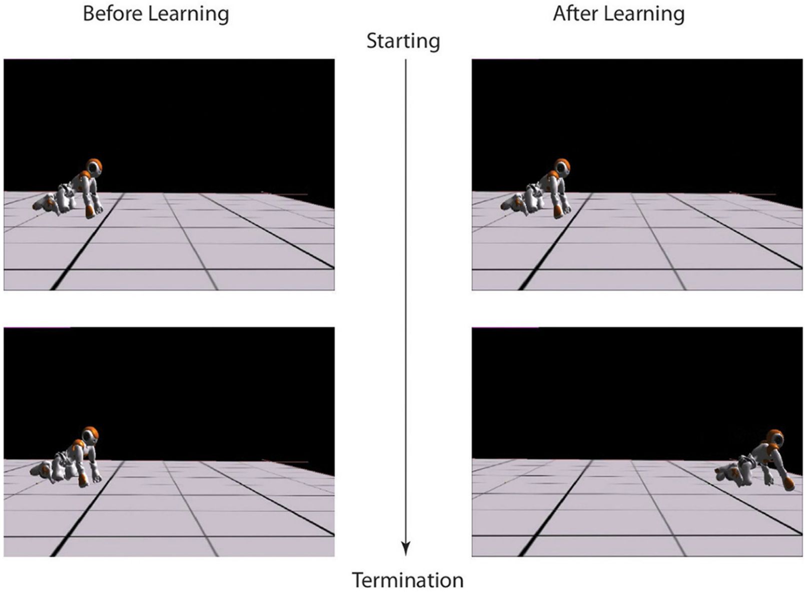

FIGURE 5 | Left: the beginning and termination snapshots of crawling before learning. **Right**: the beginning and termination snapshots of crawling after learning.

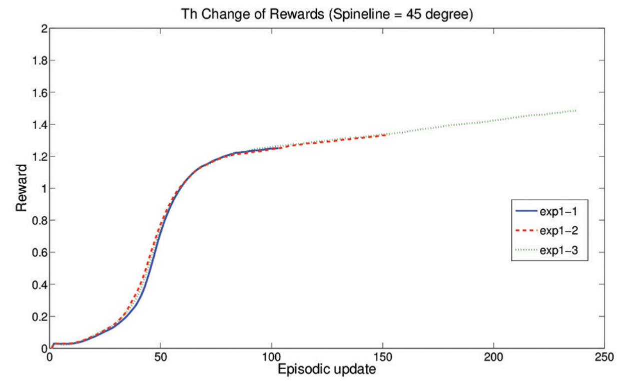

FIGURE 6 | Learning curves of three experiments. exp1-n represents the result of the nth run in the experiment 1.

also are the phases of CPGs controlling roll joints shifted compared to the original CPG output without learning. It seems the DA layer modeled as motor primitives has the capability to deal with the reshaping of different limit cycle output and even to adjust the phase difference which is set inappropriately in the PF layer. Apart from these, it can also tune the posture. From the joint dynamics of roll motion, it is clearly observed that the limit cycles of roll joints are shifted, in which case the oscillation centers of roll joints are adaptively adjusted. Compared to the explicit posture-control terms in our previous work Li et al. (2013a,b), the implicit terms grounded in the motor primitives can integrate two functions: DA and posture control. To verify the functionality of posture control, we conduct a second experiment.

3.1.2. *Sub-experiment: Posture adjustment*

In the experiment 2, the objective is to verify the capability of the proposed CPG-Actor-Critic architecture on the adjustment of joint posture. Actually, the spineline angle reward proffers a control signal of limiting the whole-body posture. With a loose control coefficient [e.g., $\sigma > 0.02$ in Equation (10)] or without the spineline restriction, the robot will only consider the maximization of crawling distance, ignoring the maintenance of the posture. This causes a convergence to an extreme crawling behavior. In human reality, parents always need to guide a right posture by holding up or lifting the infant's body when they are crawling. Therefore, the posture limitation is necessary.

For testing the posture control abilities, two spinelines are chosen ($x_0 = 1.03$ and $x_0 = 1.08$, approximately 31° and 28°). Two independent learning experiments are performed respectively for each of these two spineline-angle controlled postures. With the results obtained, the comparison of limit cycles of joints in 4 runs are given in **Figure 8**. For each group of the results (black and red curves), the crawling joint dynamics converge to similar limit cycles. In terms of the motion of pitch joints (shoulders and hips), from **Figures 8A,B**, the deviation between two limit-cycle centers is blurry. However, it is conspicuous for the roll joints, especially hip roll joints (**Figures 8C,D**). The limit-cycle centers are both shifted rightwards for shoulder and hip roll motion from posture 2 (28° spineline angle) to posture 1(31° spineline angle). This limit-cycle-center shifts correspond to the closing-inward and opening-outward posture changes of shoulder and hip joints. This is a typical whole-body motion of lifting the center of gravity of the body and increasing the spineline angle. Compared to the explicitly allocated posture-change terms in previous work Li et al. (2013a,b), using motor primitives can interactively rule out the unnecessary joints for posture control. For example, in the experiment 2, to change from posture 1 to posture 2, the system determines to fixating on altering the posture of roll joints other than pitch joints based on the whole-body motion logic.

3.2. TRANSFERRED TEST ON THE PHYSICAL ROBOT

In this article, with 7 learned sets of parameters, the learned CPGs are transferred to the physical NAO robot for testing. In all the experiments, the popular Webots simulator (Michel, 2004) based on ODE (open dynamics engine) is used. In order to successfully test the learned motion from the simulated robot to the physical one, some preconditions have to be realized. As discussed in previous work Li et al. (2013a), the possible failures of transferred results on physical robots could be caused by the disparity in physics engines and difference between simulation time and real time. In our work, the frequency of the CPG is doubled while being transferred. 5 out of 7 results can be successfully transferred except the results for the posture ($x_0 = 1.03$). After the CPG amplitudes of pitch joints are reduced to 70%, the failed transferring becomes successful. **Figure 9** shows the snapshots of one-step crawling on the physical robot (for details, please refer to the video Li, 2013b). Compared to the previous implementation with only optimized postures, the left-right curvy motion of the spineline, a typical characteristic of crawling behaviors (Righetti et al., 2008), emerges after learning.

3.3. EXPERIMENT 2: LEARNING DOUBLE-SUSPENSION GALLOP

In this experiment, the CPG-Actor-Critic architecture is transferred to the puppy robot (ghostdog) with a flexible body for learning the double-suspension gallop, a typical dog running Gait (dog). The rear limbs are synchronized in phase and so are the front limbs. The main joints controlled by the CPGs are the rear and front hip joints. The knee part of this robot is fully passive with a spring-damper system. In order to maximize the running distance, a reward based on the distance only is used, in which case the robot is required to run as fast as possible in a time-fixed episode. Similar to the reward function of learning to crawl, we adopt the distance-related part of that reward function as the only learning landmark. With both eNAC and PoWER, the objectives of experiment 2 include: 1. Test the morphological transferability of the CPG-Actor-Critic. 2. Compare the learning efficiency of eNAC and PoWER.

3.3.1. *Morphological Transferability*

In the experiment 1, the learning architecture has already been demonstrated to be workable on a rigid-body system, the NAO robot. However, a complete architecture should also be able to work on different morphologies, especially in the case that a robot's body cannot be accurately modeled in terms of its morphological flexibility. Then the learning/adaptation system can help the robot to find out solutions by itself with some prior

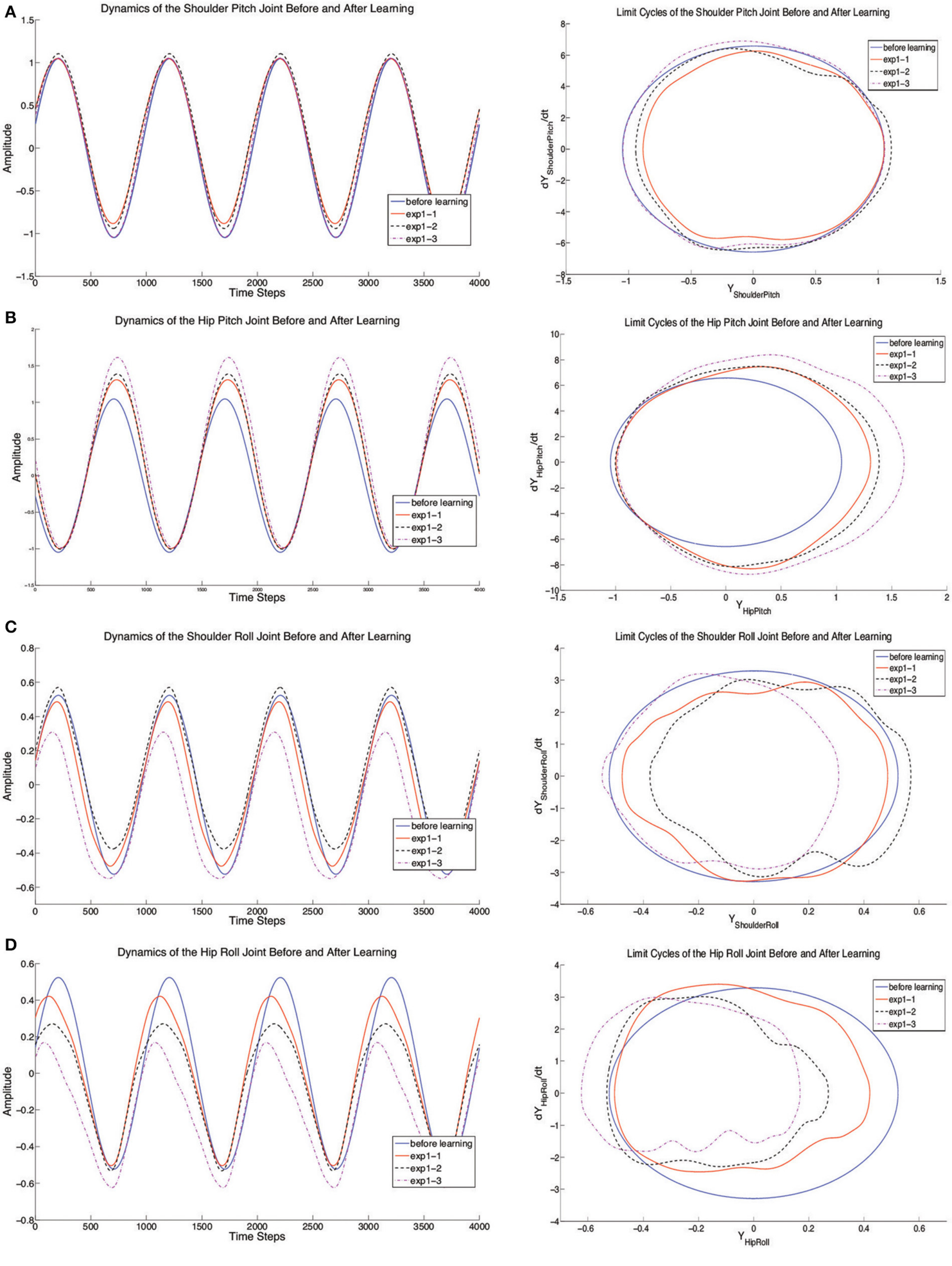

FIGURE 7 | The dynamics of joints (Shoulder Pitch and Roll, Hip Pitch, and Roll) before and after learning. The left side of **(A–D)** represent the trajectory change for each joint and the right side of **(A–D)** represent limit cycles of each joint. In each figure, the blue line indicates the original joint motion without learning and the red solid lines, black dashed lines and purple dashed lines show the results of experiment 1–3 after learning.

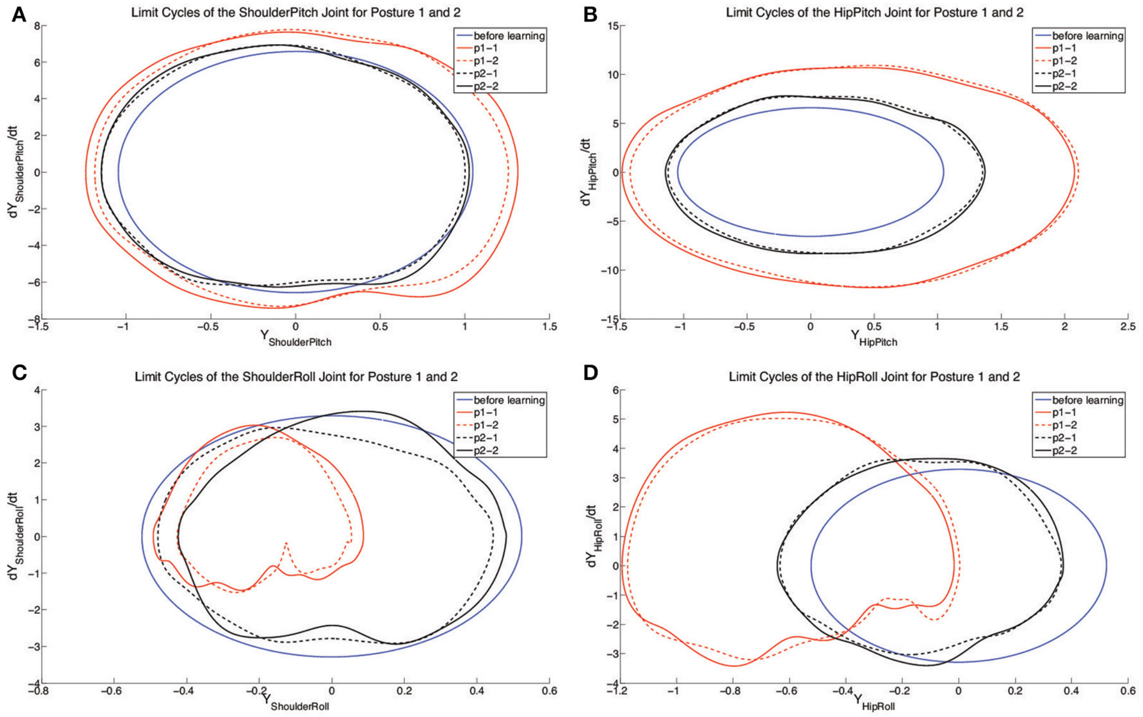

FIGURE 8 | The figure shows the limit cycles of each controlled joint with 2 independent learning experiments for two different body postures. (A–D) represent the limit cycles of each joint before and after learning. The blue solid lines represent the limit cycles for each joint before learning. Red solid and dashed lines indicate the limit cycles for tested posture 1 ($x_0 = 1.03$, ~31°). Black solid and dashed lines indicate the limit cycles for tested posture 2 ($x_0 = 1.08$, ~28°). pn-n is the abbreviation of posture n-experiment n.

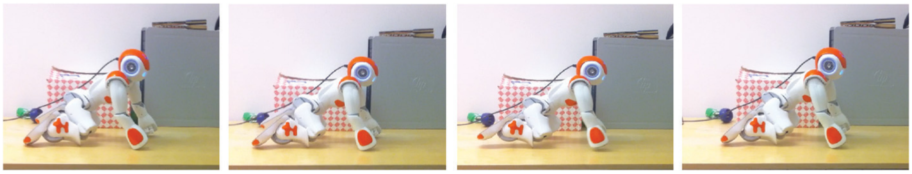

FIGURE 9 | The implementation on the physical robot. This figure shows the video snapshot of one-crawling-step NAO robot on a wooden flat table (One crawling step means one time alternation of the supporting leg and arm).

knowledge. In this article, the baseline behaviors serve as prior knowledge and the emergence of a particular gait based on this prior knowledge becomes so intriguing. **Figure 10** presents the snapshots of double-suspension gallop gait before and after learning. The difference before and after learning is so conspicuous (For the details, please refer to the video Li, 2013c). Before learning, the puppy robot can move very hard by scratching the ground. But after learning, a new gait emerges from the previous ground-scratching behavior. In the new gait, it seems the robot can take advantage of its own characteristics of the physical body to move as fast as possible. In **Figure 10**, the third snapshot of two after-learning gaits both demonstrate that the robot uses the spring of rear legs to bounce up the whole body in the air so that it can move much further and the first snapshot of two after-learning gaits also both demonstrate that the spring of front legs are used to reduce the impact when the body hits the ground from the air. These phenomena possibly indicate that the CPG-Actor-Critic might have the ability to realize the morphological advantages of a certain body.

In details, **Figure 11** presents the learning results of converged joint dynamics. For each RL technique of eNAC and PoWER, there are three learning trials conducted with the same initial conditions (posture and position). The ghostdog robot learns repeatedly in the simulator and is automatically reset by supervisor functions according to Michel (2004). It is shown in **Figure 11** that each batch of three trials by eNAC and PoWER qualitatively converge to similar joint dynamics respectively. There are some common features of the converged dynamics captured by both eNAC and PoWER. Firstly, as for the front legs, the changes focus on increasing the amplitude of joint oscillation. Secondly, as for the rear legs, both eNAC and PoWER change the originally equal stance and swing phase to the dynamics in which the stance phase is much longer than swing phase. In terms of double-suspension gallop, a longer stance phase ($T1$ and $T1'$) on the rear legs drive

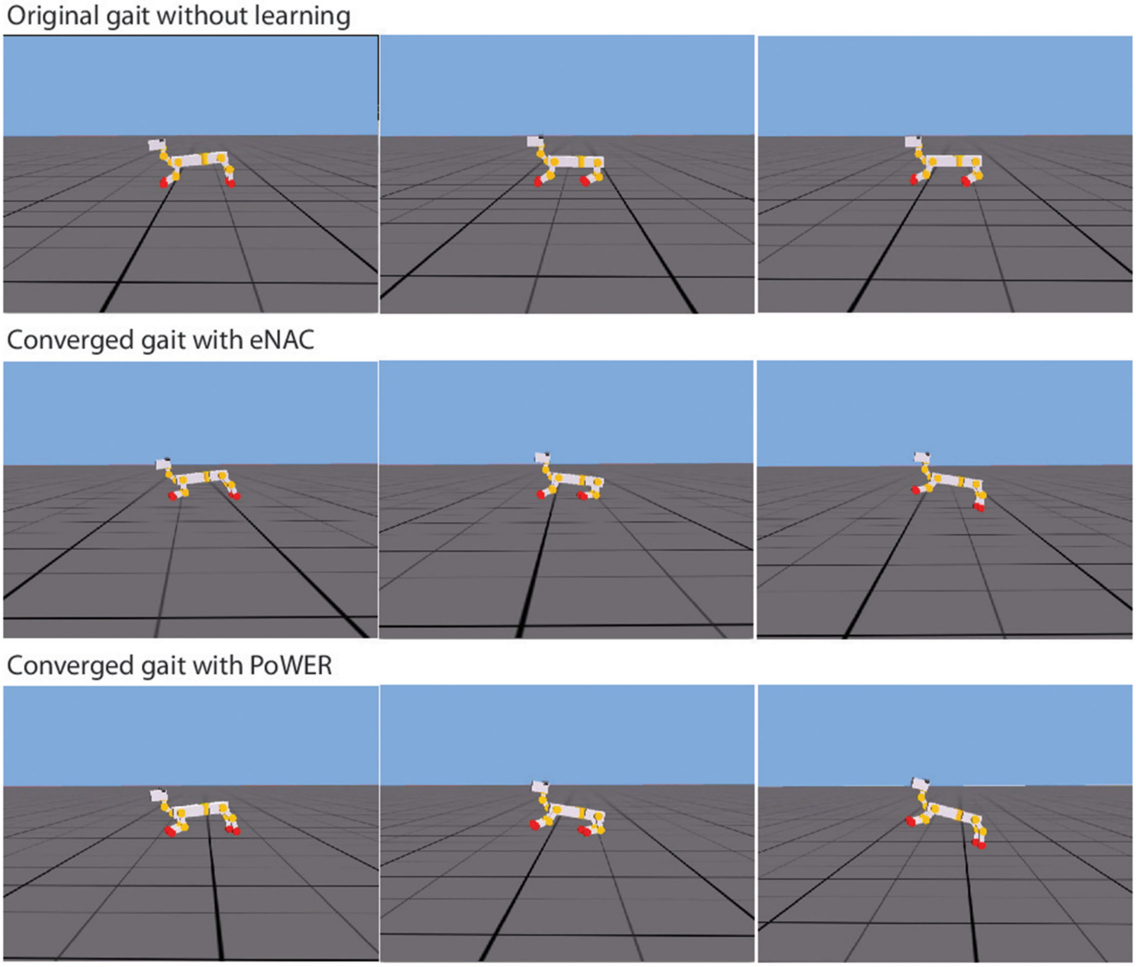

FIGURE 10 | The snapshots of gallop gait under three conditions. The first row shows the original gait without learning, the middle row shows the converged gait after eNAC learning and the final row shows the converged gait after PoWER learning.

the front legs off the ground and a shorter swing phase ($T2$ and $T2'$) of the rear legs makes them follow the move direction and finish the phase transition from swing to stance as fast as possible when the front legs hit the ground.

3.3.2. Comparison between eNAC and PoWER

According to Kober et al. (2012), as a typical gradient based approach, eNAC suffers the problem of finding out a proper learning rate in the supervised learning. It is an open problem for policy gradient approaches in terms of good learning performance but EM-based policy search can avoid this problem. In order to compare eNAC and PoWER for RL cases, a stable learning rate is chosen to be used in eNAC ($\alpha = 0.1$, when $\alpha > 0.5$, five trials are conducted but none of them can successfully and stably learn with more than 20 updates (less than 20 updates is considered as failure). When $\alpha = 0.2, 0.3, 0.4$, the failure rate is about 20%, 20%, 40% for five trials respectively). **Figure 12** shows that PoWER outperforms eNAC in learning speed by boosting the reward in three trials respectively. eNAC is able to optimize the moving distance but it gets stuck in some local optima while PoWER can find better solution (further distance in the same period). This result is similar to the cases presented in Kober et al. (2012) supervised learning experiments (i.e., PoWER outperforms eNAC in terms of learning rate and results). Nevertheless, eNAC is very sensitive to the reward value. When the reward value increases, the update variance starts to increase and the stability of learning starts to deteriorate. But PoWER is quite stable to the reward value change. In terms of convergence, the convergence of PoWER and eNAC is determined by δ with Cacla logic. Because of the different learning speed, Cacla converged as no further expected return of samples can make δ positive so that the final converged expected reward oscillates around the average optima (**Figure 12**). However, eNAC converged as δ is almost zero ($|\delta| < 10^{-4}$).

3.4. SUMMARY

In this section, the DMPs based CPG-Actor-Critic architecture is tested for its learnability on a rigid body robot, the NAO humanoid, for learning to crawl. Then it is transferred to a "soft"-bodied robot, the ghostdog robot, for learning double-suspension

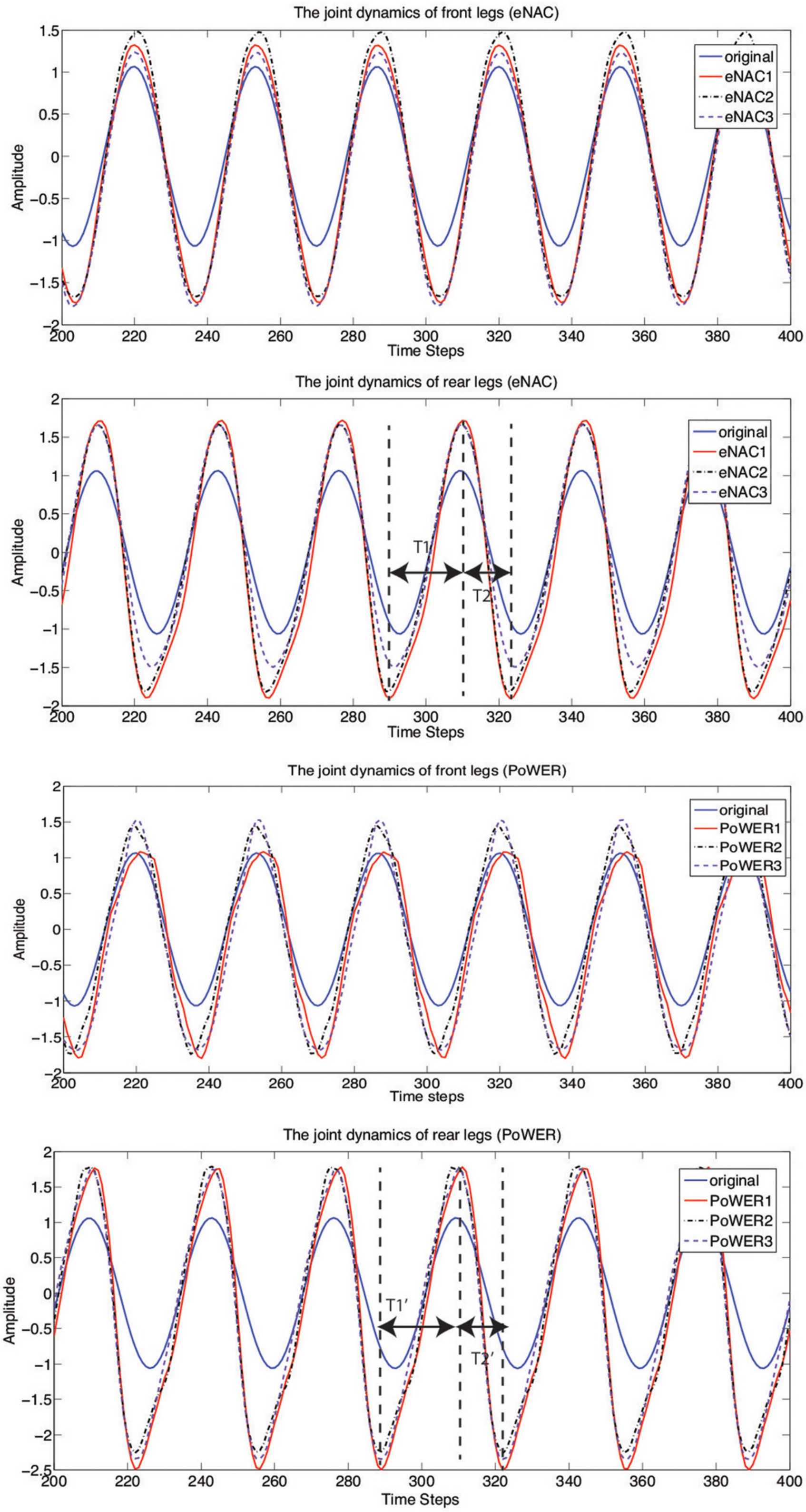

FIGURE 11 | The front and rear joint dynamics of the ghostdog robot before and after learning with eNAC and PoWER in three trials (marked by eNAC1 ~ 3 and PoWER1 ~ 3). $T1$, $T2$, $T1'$, and $T2'$ represent the ascending time and descending time of the joint dynamics tuned by eNAC and PoWER after learning. Before learning, $T1 = T2$ and $T1' = T2'$.

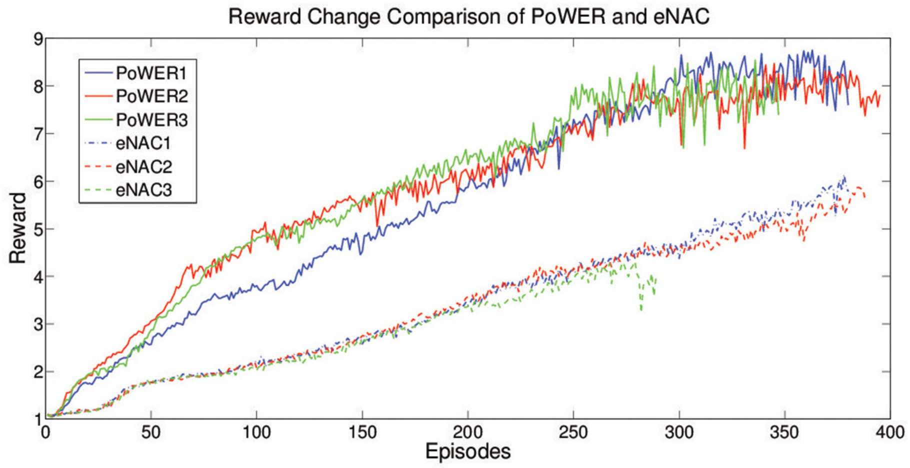

FIGURE 12 | The learning curve of eNAC and PoWER over 400 episodes which comprise three trials. The reward reflects the distance the robot moves in the same period.

gallop. Two advanced RL algorithms (eNAC and PoWER) are utilized and the results are compared. With two state-of-the-art RL techniques, qualitatively similar gaits emerge from the interaction amongst the CPG controller, the environment and the body in an actor-critic mechanism. In both experiments, DMPs work as an universal limit cycle modifier to reshape the existing baseline behavior into optimal gait dynamics. The functionality of DMPs is a mathematical framework for optimizing/learning locomotion gaits without explicit sensory feedback but serves as a mechanism functionally similar to sensory-feedback-reshaping (Grillner et al., 2008; Ijspeert et al., 2013).

4. CONCLUSION

4.1. eNAC AND PoWER

eNAC and PoWER, as two different RL algorithms, are based on the policy gradient and the EM-based policy search, respectively. They are compared the first time for learning a periodic motion. According to the results in Experiment 2, we can summarize the difference between them as follows:

(1) Learning speed. PoWER searches a better policy faster than eNAC as the small learning rate α of policy gradient slows down the learning speed of policy gradient approaches.
(2) Converged results. Both PoWER and eNAC possibly get stuck in local optima according to Kober et al. (2012). However, PoWER can converge to better results than eNAC.

4.2. DMPs BASED CPG-ACTOR-CRITIC

In this article, the proposed CPG-Actor-Critic based on DMPs seems to be able to optimize/learn a gait given an initially rough baseline behavior and a body. The forcing term $f(W_i, p)$ in Equation (1) works like sampling sensors which “perceive” a sufficiently large number of proprioceptive points of the CPGs so that it can adapt them flexibly into distinct dynamic patterns on the basis of actor-critic interaction. Even though this implementation of motor primitives with RL approaches instead of supervised learning opens a new page for locomotion learning, this approach still has some disadvantages: Firstly, learning might be slow. In the experiments, each one takes about 6–7 h to complete. Therefore, the learning process might not be transferred to the physical robot. On the other hand, a fast learning mechanism might be needed for a fast adaptation to the dynamical environmental changes in our architecture. The potential solution is either to develop a higher level of cognitive interpretation of environmental needs for switching different learned gaits (e.g., in Aoi and Tsuchiya, 2005) or use a faster learning algorithm in a fast adaptation mechanism (e.g., in Manoonpong et al., 2013). Secondly, the frequency is not adaptive. In the work presented above, all the CPG frequencies are fixed. Even though the motor primitives can innately preserve the learned dynamics when the frequency is changed, it still cannot guarantee that the new frequency patterns still can work when the whole-body dynamics change with the oscillation frequency. As a matter of fact, after reducing the frequency from 1.0 to 0.5, the robot’s body dynamics change and the crawling in experiment 1 cannot be properly presented. The solution to this problem might be to use hybrid learning based on eNAC by counting in the frequency parameters (Kober, 2012). Thirdly, the implementation of CPGs is not energy efficient. In our work, the CPGs are used as trajectory generators. The layered architecture still lacks an adaptive approach to altering the stiffness of joints. On the other hand, low energy efficiency might be a natural flaw of rigid body robots. Even though force control might be able to improve the energy issues on a rigid body, the inflexibility of the body is still a stumbling block preventing a robot from being energy-efficient for locomotion. Finally, the lack of a memory architecture. Even though DMPs can optimize/learn gaits from the prior knowledge, it cannot memorize the relation between the environment and parameter space. This is an open problem for locomotion modeling.

In conclusion, DMPs based CPGs are able to not only learn demonstrated/supervised signals (Gams et al., 2009) but also adapt to flexible patterns based RL approaches in our work. Even though it is not a complete solution for the integration of sensory feedback (e.g., moving distance, spineline angle and muscle reflex. Muscle reflex is not used in our work since the NAO robot does not have pressure sensors on the hand), it offers a mathematical solution to mimic the same function of sensory feedback in reshaping and shifting limit cycles. The advantage of using DMPs as a dynamic adaptation tool can be summarized as follows:

(1) *an optimizer for an existing gait*. Since there is no context-free locomotion capability and the environment is not possible to model, using a well-designed gait and optimize it in the environment in which the original gait cannot work well.
(2) *a gait searcher*. Given prior knowledge about a certain gait and a body, locomotion modelers might have difficulty in determining the detailed motion of each DOF/joint. Using DMPs with RL mechanism enables seeking out the optimal solution. In this article, two experiments show the process of the emergence of a particular gait based on its baseline motion (prior knowledge).

4.3. A GENERIC VIEW: TWO SYSTEMS AND ONE MECHANISM

A lot of inspiration related to locomotion learning/development can be extracted from cognitive science (e.g., Thelen, 1996), neuroscience (e.g., Schore, 1994; Grillner et al., 2008), psychology (e.g., Clearfield, 2004; Adolph et al., 2012) and robotics (e.g., Pfeifer and Bongard, 2006). From the perspective of Thelen (1996), locomotion development/learning is focused on the formation and adaptation of the so-called "attractors" in a dynamic system. This assumption indicates that locomotor system design should not be focused on how a static system can be modeled but how a dynamic system might develop in the interaction with environments. The stagnation is only one "special" attractor of the system. In this sense, DMPs have been assumed to represent locomotion attractors in Ijspeert et al's work (Ijspeert et al., 2013). Both Schore (1994) and Grillner et al. (2008) imply that locomotion learning might be RL-related from the perspective of neuropsychology and neural structures. From the psychological point of view, Clearfield (2004) indicates the developmental relation of locomotion to spatial memory including the distance. Adolph et al. (2012) recently explained why the repetitiveness is important for infants to learn locomotion. Finally, Pfeifer and Bongard (2006) rethink the locomotion and emphasize the interaction between the body and the environment. Based on the above view of locomotion from different angles, locomotion learning is an affective-related, interactive and repetitive process with cognitive cues.

Therefore, in order to have a sketch of a dynamic locomotor system composed of the three components mentioned in Section 1, in this article, we propose the "two systems and one mechanism" architecture. Two systems cover one baseline motion system and one adaptation system. The former includes a general model which is able to handle basic locomotion functions (e.g., the coordination and synchronization of DOFs, gait transition). The latter comprises basic adaptation function/interface to adapt basic locomotion dynamics into context-specific dynamics. One mechanism is an affective-modulated process of organizing how the DA can happen with more complex perceptual information (e.g., visual signals) and determining what context-specific dynamics the locomotor system should adapt into.

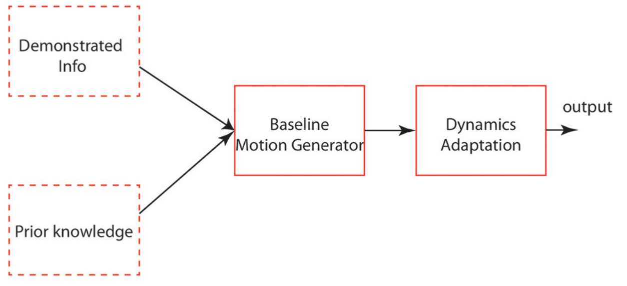

FIGURE 13 | The generic architecture of locomotion learning. The dashed-line blocks represent the choices of input. The other blocks represent the functions of each layer.

The general architecture based on the points above-mentioned in **Figure 13** explains an applicable schema for locomotion learning. The baseline motion generator can be modeled based on demonstrated signals (Nakanishi et al., 2004; Gams et al., 2009) or prior knowledge (Righetti, 2008). After this is accomplished, the basic motion dynamics is to be adapted into context-related dynamics by an interactive mechanism. In our modeling approach, the four-cell architecture encodes the basic patterns of different kinds of gaits. Each cell in this architecture can also be modeled as a complicated neural system and more neural systems are coupled in this network. DMPs, as a mathematical dynamics modifier, work with actor-critic RL mechanism (Kober and Peters, 2010) to optimize/learn a locomotor system. In the future work, a sophisticated memory system, which includes short-term memory and long-term memory, is required in our system to map the contextual factors into parameter space. Also, a sensory feedback integration will be considered to be used together with DMPs based CPG-Actor-Critic.

ACKNOWLEDGMENT

The authors would like to thank the European RobotDoC project (www.robotdoc.org) for funding and supporting this research.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fnbot.2014.00023/abstract

REFERENCES

Adolph, K. E., Cole, W. G., Komati, M., Garciaguirre, J. S., Badaly, D., Lingeman, J. M., et al. (2012). How do you learn to walk? Thousands of steps and dozens of falls per day. *Psychol. Sci.* 23, 1387–1394. doi: 10.1177/0956797612446346

Ajallooeian, M., van den Kieboom, J., Mukovskiy, A., Giese, M., and Ijspeert, A. (2013). A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape. *Physica D*. 263, 41–56. doi: 10.1016/j.physd.2013.07.016

Aoi, S., and Tsuchiya, K. (2005). "Transition from quadrupedal to bipedal locomotion," in *Intelligent Robots and Systems, 2005* (Alberta).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Buchli, J., Righetti, L., and Ijspeert, A. J. (2006). "Adaptive frequency oscillators applied to dynamic walking i. Programmable central pattern generators," in *Proceedings of Dynamic Walking 2006 (this volume)* (Ann Arbor, MI).

Clearfield, M. W. (2004). The role of crawling and walking experience in infant spatial memory. *J. Exp. Child Psychol.* 89, 214–241. doi: 10.1016/j.jecp.2004.07.003

Dayan, P. (2005). *Theoretical Neuroscience: Computational And Mathematical Modeling of Neural Systems*. Cambrige; Boston, MA: Massachusetts Institute of Technology Press.

Dayan, P., and Hinton, G. E. (1997). Using expectation-maximization for reinforcement learning. *Neural Comput.* 9, 271–278. doi: 10.1162/neco.1997.9.2.271

Degallier, S., Righetti, L., Natale, L., Nori, F., Metta, G., and Ijspeert, A. (2008). "A modular bio-inspired architecture for movement generation for the infant-like robot iCub," in *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics* (Scottsdale, AZ), 795–800. doi: 10.1109/BIOROB.2008.4762847

Endo, G., Morimoto, J., Matsubara, T., Nakanishi, J., and Cheng, G. (2008). Learning CPG-based biped locomotion with a policy gradient method: application to a humanoid robot. *Int. J. Robot. Res.* 27, 213–228. doi: 10.1177/0278364907084980

Gait (dog). (2009). Available online at: http://www.en.wikipedia.org/wiki/Gait_(dog)

Gams, A., Ijspeert, A. J., Schaal, S., and Lenarčič, J. (2009). On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Auton. Robots* 27, 3–23. doi: 10.1007/s10514-009-9118-y

Golubitsky, M., and Stewart, I. (2003). *The Symmetry Perspective: From Equilibrium to Chaos in Phase Space and Physical Space, Progress in Mathematics Series*. Basel: Birkhäuser.

Grillner, S. (1985). "Neural control of vertebrate locomotion - central mechanisms and reflex interaction with special reference to the cat," in *Feedback and Motor Control in Invertebrates and Vertebrates*, eds W. Barnes and M. Gladden (Glasgow, UK: Springer), 35–56.

Grillner, S., Hellgren, J., Ménard, A., Saitoh, K., and Wikström, M. A. (2005). Mechanisms for selection of basic motor programs–roles for the striatum and pallidum. *Trends Neurosci.* 28, 364–370. doi: 10.1016/j.tins.2005.05.004

Grillner, S., Wallén, P., Saitoh, K., Kozlov, A., and Robertson, B. (2008). Neural bases of goal-directed locomotion in vertebrates–an overview. *Brain Res. Rev.* 57, 2–12. doi: 10.1016/j.brainresrev.2007.06.027

Harischandra, N., Knuesel, J., Kozlov, A., Bicanski, A., Cabelguen, J.-M., Ijspeert, A., et al. (2011). Sensory feedback plays a significant role in generating walking gait and in gait transition in salamanders: a simulation study. *Front. Neurorobot.* 5:3. doi: 10.3389/fnbot.2011.00003

Hasselt, H. V. (2007). *Insights in the Reinforcement Learning*. Ph.D. thesis, Utrecht University, Utrecht, Netherlands.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review Neurobiology of CPGs. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput.* 25, 328–373. doi: 10.1162/NECO-a-00393

Jaakkola, T., Singh, S. P., and Jordan, M. I. (1995). "Reinforcement learning algorithm for partially observable markov decision problems," in *Advances in Neural Information Processing Systems 7*, eds G. Tesauro, D. S. Touretzky, and T. K. Leen (Denver, CO: MIT Press), 345–352.

Kail, R. V., and Cavanaugh, J. C. (2012). *Human Development: A Life-span View*. Wadsworth: Wadsworth Cengage Learning.

Kakade, S. (2001). A Natural policy gradient. *Adv. Neural Inf. Process. Syst*. 238, 1531–1538.

Kober, J. (2012). *Learning Motor Skills: from Algorithms to Robot Experiments, Springer Tracts in Advanced Robotics*. Cham: Springer International Publishing AG.

Kober, J., and Peters, J. (2010). Policy search for motor primitives in robotics. *Mach. Learn.* 84, 171–203. doi: 10.1007/s10994-010-5223-6

Kober, J., Wilhelm, A., Oztop, E., and Peters, J. (2012). Reinforcement learning to adjust parametrized motor primitives to new situations. *Auton. Robots* 33, 361–379. doi: 10.1007/s10514-012-9290-3

Kohl, N., and Stone, P. (2004). "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Barcelona).

Latash, M. (2008). *Neurophysiological Basis of Movement, 2nd Edn*. Champaign: Human Kinetics.

Li, C. (2013a). *Crawling Before Learning and After Learning.* Available online at: http://www.youtube.com/watch?v=sb0brwcajcw

Li, C. (2013b). *Crawling Nao Robot Learned by cpg-actor-critic*. Available online at: http://www.youtube.com/watch?v=6CtKSDQV9sw

Li, C. (2013c). *Galloping Ghostdog Learned by cpg-actor-critic.* Available online at: http://www.youtube.com/watch?v=5z7zisenphg

Li, C. (2014). *Reinforcement Learning of Locomotion based on Central Pattern Generators*. Ph.D. thesis, Department of Computer and Information Science, The Institute of Technology, Linköping University, Linköping, Sweden. Available online at: http://liu.diva-portal.org/smash/record.jsf?pid=diva2:712601

Li, C., Lowe, R., Duran, B., and Ziemke, T. (2011). "Humanoids that crawl: comparing gait performance of iCub and NAO using a CPG architecture," in *2011 IEEE International Conference on Computer Science and Automation Engineering* (Shanghai), 577–582.

Li, C., Lowe, R., and Ziemke, T. (2012). "Modelling Walking Behaviors Based on CPGs: a Simplified Bio-inspired Architecture," in *From Animals to Animats 12 Lecture Notes in Computer Science*, Vol. 7426 (Odense), 156–166. doi: 10.1007/978-3-642-33093-3-16

Li, C., Lowe, R., and Ziemke, T. (2013a). "Crawling Posture Learning in Humanoid Robots using a Natural-Actor-Critic CPG Architecture," in *Proceedings of European Conference on Artificial Life* (Taormina, ME).

Li, C., Lowe, R., and Ziemke, T. (2013b). Humanoids learning to walk: a natural CPG-Actor-Critic architecture. *Front. Neurorobot.* 7:5. doi: 10.3389/fnbot.2013.00005

Manoonpong, P., Parlitz, U., and Wörgötter, F. (2013). Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Front. Neural Circuits* 7:12. doi: 10.3389/fncir.2013.00012

Marchese, A. D., Onal, C. D., and Rus, D. (2013). *Experimental Towards a Self-contained Soft Robotic Fish: On-Board Pressure Generation and Embedded Electro-permanent Magnet Valves, Volume 88 of Springer Tracts in Advanced Robotics*. Heidelberg: Springer International Publishing.

Michel, O. (2004). Webots TM : professional mobile robot simulation. *Adv. Robot. Syst.* 1, 40–43.

Morimoto, J., Nakanishi, J., Endo, G., Cheng, G., Atkeson, C. G., and Zeglin, G. (2005). "Poincaré-map-based reinforcement learning for biped walking," in *ICRA* (Seattle, WA), 2381–2386.

Mussa-Ivaldi, F. A. (1999). "Motor primitives, force-fields and the equilibrium point theory," in *From Basic Motor Control to Functional Recovery*, eds N. Gantchev and G. N. Gantchev (Sofia: Academic Oublishing House), 392–398.

Nakamura, Y., Mori, T., Sato, M.-A., and Ishii, S. (2007). Reinforcement learning for a biped robot based on a CPG-actor-critic method. *Neural Netw.* 20, 723–735. doi: 10.1016/j.neunet.2007.01.002

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robot. Auton. Syst.* 47, 79–91. doi: 10.1016/j.robot.2004.03.003

Nassour, J., Hugel, V., Ouezdou, F. B., Cheng, G., and Member, S. (2013). Failure maps: applied to humanoid robot walking. *IEEE Trans. Neural Netw.* 24, 81–93. doi: 10.1109/TNNLS.2012.2224370

Orlovsky, G., Deliagina, T. G., and Grillner, S. (1999). *Neuronal Control of Locomotion From Mollusc to Man*. New York, NY: Oxford University Press. doi: 10.1093/acprof:oso/9780198524052.001.0001

Peters, J. (2007). *Machine Learning for Motor Skills in Robotics*. Ph.D. thesis, University of Southern California, Los Angeles, CA.

Peters, J., and Schaal, S. (2008). Natural actor-critic. *Neurocomputing* 71, 1180–1190. doi: 10.1016/j.neucom.2007.11.026

Pfeifer, R., and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. Cambridge: The MIT Press.

Pfeifer, R., Iida, F., and Lungarella, M. (2014). Cognition from the bottom up: on biological inspiration, body morphology, and soft materials. *Trends Cogn. Sci.* 18, 404–413. doi: 10.1016/j.tics.2014.04.004

Pouya, S., van den Kieboom, J., Sprwitz, A., and Ijspeert, A. J. (2010). "Automatic gait generation in modular robots: "to oscillate or to rotate; that is the question," in *IROS* (Taipei: IEEE), 514–520.

Righetti, L. (2008). *Control of Legged Locomotion using Dynamical Systems*. Ph.D. thesis, Lausanne, EPFL, Lausanne, Switzerland.

Righetti, L., and Ijspeert, A. J. (2006). "Design methodologies for central pattern generators: an application to crawling humanoids," in *Proceedings of Robotics: Science and Systems* (Pisa), 191–198.

Righetti, L., Nylén, K. A., and Ijspeert, A. (2008). *Is the Locomotion of Crawling Human Infants Different from Other Quadruped Mammals?* Technical report, Swiss Federal Institute of Technology Lausanne.

Rybak, I. A., Shevtsova, N. A., Lafreniere-Roula, M., and McCrea, D. A. (2006). Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *J. Physiol.* 577(Pt 2), 617–639. doi: 10.1113/jphysiol.2006.118703

Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2004). "Learning movement primitives," in *International Symposium on Robotics Research (ISRR2003)* (Siena: Springer).

Schore, A. N. (1994). *Affect Regulation and the Origin of the Self: The Neurobiology of Emotional Development*. New York, NY: Psychology Press, Taylor & Francis Group.

Siciliano, B., and Khatib, O. (2008). *Springer Handbook of Robotics. Gale Virtual Reference Library*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-540-30301-5

Thelen, E. S. (1996). *A Dynamic Systems Approach to the Development of Cognition and Action*. Cambrige, MA: MIT Press.

Ueno, S., Takemura, K., Yokota, S., and Edamura, K. (2014). Micro inchworm robot using electro-conjugate fluid. *Sens. Actuat. A Phys.* 216, 36–42. doi: 10.1016/j.sna.2014.04.032

van Hasselt, H., and Wiering, M. A. (2007). "Reinforcement learning in continuous action spaces," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (Honolulu, HI: Adprl), 272–279. doi: 10.1109/ADPRL.2007.368199

Wiering, M., and van Otterlo, M. (2012). *Reinforcement Learning: State-of-the-Art, Adaptation, Learning and Optimization*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-642-27645-3

Wikipedia. (2013). *Crawling (Human)*. Available online at: http://en.wikipedia.org/wiki/Crawling

Zhao, X., Zhang, J., and Qi, C. (2012). "CPG and reflexes combined adaptive walking control for AIBO," in *2012 11th International Conference on Machine Learning and Applications* (Boca Raton, FL), 448–453. doi: 10.1109/ICMLA.2012.81

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 20 March 2014; accepted: 14 August 2014; published online: 02 October 2014.

Citation: Li C, Lowe R and Ziemke T (2014) A novel approach to locomotion learning: Actor-Critic architecture using central pattern generators and dynamic motor primitives. Front. Neurorobot. ***8****:23. doi: 10.3389/fnbot.2014.00023*

This article was submitted to the journal Frontiers in Neurorobotics.