

Abstract

The field of bipedal robots has advanced significantly over the last few decades. Yet even the most advanced robots cannot match the efficiency and gracefulness of human movement, which relies on a coordination between neural mechanisms and the dynamics of the skeletal structure. As an alternative to model based methods, Central Pattern Generators such as the Matsuoka oscillator have emerged as a viable biologically inspired approach to achieve robust bipedal locomotion in robots. The use of the Matsuoka oscillator on position controlled robots is limited by the fact that this oscillator generates torques and is more suitable for torque controlled robots. In this paper, a novel way to overcome this problem is presented. This technique can be extended in the future into a framework that can lead to robust walking and is not tightly coupled to a particular robot model.

Contents

1	Introduction	2
2	Related Work	3
3	Biological Motivation	4
4	Choice of Oscillator	6
5	Matsuoka Oscillator	6
6	Robot Joint Control	12
7	Future Work	15
8	Conclusion	16

1 Introduction

Walking on two feet is an activity that all of us take for granted. Yet this apparently simple action involves a host of complex mechanisms which are tied together to produce the bipedal movements of human beings. The ability to move around on two legs has allowed humans to effectively deal with a wide variety of terrains.

| The efficiency and versatility of biped movement is undeniable, and so it is natural for us to desire our robots to have the same capabilities.

| However, the benefits of legged robots comes at a price. Bipedal locomotion is inherently unstable, and a considerable amount of effort goes into maintaining balance. In addition, it involves a complicated mix of movements of different limbs in a synchronous fashion. Human walking involves multiple degrees of freedom and is achieved with the combined effort of muscles, bones, ligaments and tendons. Replicating this movement in robots using the conventional servo motors we have at our disposal today, is also a very difficult task.

In the literature, perhaps the most popular approach for solving the problem of bipedal movement is that of using the Zero Moment Point (ZMP) [35, 10]. ZMP is defined as that point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes [34]. By ensuring that the ZMP always lies within the support polygon of the robot, dynamical stability of the system can be ensured. The approach then is to find a suitable trajectory of the ZMP and then to use inverse kinematics to figure out what the individual joint angle trajectories should be which satisfy the required ZMP trajectory.

| However, despite the popularity of ZMP, it has some limitations. Firstly, the computation of ZMP and the subsequent computation of joint angle trajectories is computationally intensive and depends on an accurate model of the robot. Secondly, such robots move in a very conservative manner with bent knees and slow carefully placed steps. In this sense, the movement of ZMP based robots is very different from the movement of humans, which is characterized by almost straight knees and fast steps. Animals are adept at utilizing the passive dynamics of their bodies for walking. This leads to an energy efficient mode of transport. ZMP based robots, on the other hand, cancel out the natural dynamics of the mechanical system in an effort to maintain control over the system at all times, and this leads to high energy consumption.

As with so many other areas of robotics, for walking too, we can draw inspiration from nature and build systems that try to emulate the mechanisms present in animals. This is the approach followed in this paper, where a neural mechanism known as Central Pattern Generators (CPG) is utilized for controlling robot joints. This technique can be extended into a framework for bipedal locomotion. The remainder of this paper is organized as follows. In section 2, some relevant work is discussed. This is followed in section 3 by a discussion of the neural mechanisms of walking. The advantages of the Matsuoka oscillator over other oscillators is discussed in section 4. Section 5 contains an in-depth discussion of the Matsuoka oscillator and its properties. This is followed by section 6, where a novel process of

using an oscillator to control a position controlled robot joint is described. Section 7 describes how this technique of joint control can be extended into a framework of bipedal locomotion. Finally section 8 contains concluding remarks about the work presented in this paper.

2 Related Work

Research on bipedal locomotion has yielded a number of ways in which humanoid robots can be made to walk in a stable manner. In [11] Kajita et al. have presented a method to generate ZMP and Center of Mass (CoM) trajectories of the robot using a technique known as preview control, where the reference ZMP is known for a few seconds into the future. In [13], Kajita et al. achieved stabilization using the Linear Inverted Pendulum (LIP) model on a 42-degree-of-freedom (DoF) adult sized humanoid robot. In [12] the authors have presented a simplified model (3D-LIP) which can be used to accurately model the dynamics of a high DoF robot.

An argument against pure ZMP based methods, put forward by Missura et al. in [20, 21, 22], is that such methods are not suitable in situations where the robot may be subject to strong disturbances. The authors have presented a hierarchical gait control architecture where balance control is separated from motion generation. This architecture relies on the LIP model and is able to resist strong disturbances using a combination of different strategies including ZMP. In [32], simple models for humanoids have been investigated with respect to push recovery.

However, despite the popularity of ZMP based techniques, the gait of even highly sophisticated humanoid robots such as ASIMO is very different from the gait of humans. An interesting alternative to the computationally involved ZMP technique is that of passive walking. McGeer's work [19] has shown that highly efficient human-like gait can be achieved even by an unactuated humanoid walking down a ramp. McGeer's passive walkers have highlighted the fact that if the physical constraints of the robot can be utilized efficiently, then human-like motion is achievable without relying on excessively complicated techniques.

While passive walking relies on exploiting the mechanical constraints of the robot's structure, on its own it cannot yield walkers which can tackle different terrains, slopes or situations. To do this a controller is required which can utilize information from the robot's external and proprioceptive sensors to adapt the gait as required. In [6] a neural network controlled biped walker has been presented. A reflexive hierarchical controller controls biped walking without any kind of trajectory control. Reinforcement learning has been used to tune the parameters of the controller. In [26] sensorimotor neural networks (one for each leg) control the motion of the biped. The network is optimized using a genetic algorithm.

Taking the biological inspiration factor a step further, Central Pattern Generators have been used to design the low level neural circuitry that controls bipedal locomotion. CPGs are biological neural networks which produce rhythmic output on their own without the need of any sensory feedback. The output of the individ-

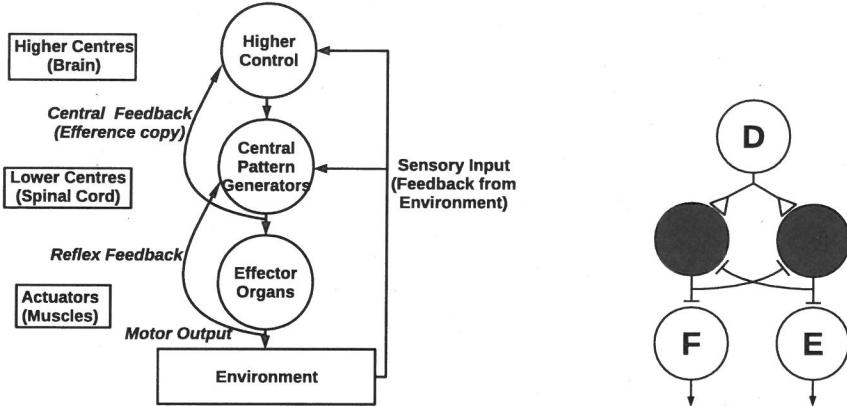
ual networks control the movement of the individual limbs and can be modulated using sensory feedback to produce different kinds of motion. Typically the parameters of the CPG are tuned using some kind of evolutionary algorithm. Various mathematical models, known as oscillators, are available which encapsulate the behavior of the CPG in the form of coupled non-linear differential equations. The most used oscillators are the Van der Pol oscillator [14], the Hopf oscillator [27] and the Matsuoka oscillator [17, 18]. In [27] and [16], Hopf oscillators have been used for pattern generation, along with some feedback mechanisms for stabilizations. Taga [33] produced the seminal work in which Matsuoka oscillators were used for gait generation of a two dimensional planar biped robot. Ishiguro et al. [9] have presented an approach in which a CPG circuit is modulated using simulated neuromodulators to achieve stable bipedal locomotion. In [8] Ijspeert has provided a comprehensive review of the use of CPGs for locomotion control.

3 Biological Motivation

In both higher level vertebrates such as humans as well as in invertebrates, neural circuits known as Central Pattern Generators are responsible for producing the rhythmic movements which are responsible for locomotion [8]. These CPGs can produce rhythmic patterns of neural activity without the need of rhythmic or sensory inputs, and they are the basis of fundamental activities such as chewing, breathing and locomotion. Several studies have proven the existence of CPGs in animals and their responsibility for the movements of the animals [3, 7, 31]. Although sensory feedback is not required to generate the rhythmic patterns by CPGs, it plays an important role in modulating the rhythmic patterns [8]. Vertebrate locomotor systems are organized such that the CPGs produce basic rhythmic patterns, and the higher brain centres modulate these patterns based on external sensory inputs [8]. Such an organization has important consequences, even from a robotics perspective, since it drastically reduces the dimensionality of higher level control signals. The higher centres do not need to specify individual muscle (or joint) activity and instead only need to modulate the CPG activity using a small set of parameters.

A general framework for the nervous control of motor systems is illustrated in figure 1(a). The first element in this framework is the CPG, located in the spinal cord [29]. This is responsible for generating coordinated rhythmic outputs for the motor neurons. The CPG in turn is controlled and modulated by the higher motor centres. There also exist feedback circuits which relay information from the muscles (proprioceptive feedback) and from the environment through other sensory pathways [29]. The feedback information is transmitted from lower to higher centres. The proprioceptive feedback informs the higher center about any deviations that exist between the commanded signal and the output [29].

The internal circuitry of the CPG generally belongs to one of three major categories: the half-center model, the closed-loop model and the pacemaker model [29]. Only the half-center model is discussed here as it is relevant to the material



(a) Neural components of motor systems. Feedback information flows from the lower centers to the higher centers. Sensory input to CPG is only used for modulation. The CPGs can produce rhythmic output in the absence of sensory input as well [29]

(b) Half-center model. D denotes driver neurons which supply a constant input, E and F are the extensor and flexor motor neurons. Filled profiles indicate inhibitory neurons, open profiles denote excitatory neurons [29]

Figure 1: Neural components of motor systems and a half-center CPG model [29]

in the subsequent sections. The half-center model, as shown in figure 1(b), was first proposed by Brown [2] to explain the alternating activation of the flexor and extensor muscles of a decerebrated cat's limbs during walking on a treadmill. Each pool of motor neurons for flexor (F) or extensor (E) is activated by a corresponding half-center of interneurons. Another set of neurons (D) provides a constant excitatory drive to the half-centres. Each half-center has inhibitory connections to the other half-center, and this ensures that when one pool is activated, the other one is deactivated. According to Brown, as the first half-center was activated, a process of fatigue would build up in the inhibitory connection between the half-centres which would ultimately lead to a switching of activity between the half-centres. The process of fatigue can be modelled by any process that induces self-inhibition of the active half-centres [29].

The inherent properties of CPGs make them a suitable candidate for the locomotion control of legged robots, as opposed to controllers based on ZMP [8]. First of all, CPG based models display stable limit cycle behavior, which means that although the generated patterns may be modified under the effect of a disturbance, as soon as the disturbance is removed, the CPG returns to its normal behavior. This property maybe beneficial in situations where robots need to tackle uneven terrain. [Secondly, CPG based controllers have very few parameters and hence higher level controllers need not deal with low level motor commands.] Thirdly, CPGs can handle sensory feedback easily, since the feedback signals can be conveniently added as coupling terms in the differential equations. This also leads to the property of entrainment, where the CPG tunes itself to the dynamics of the mechanical system and can effectively utilize the dynamics of the robot [8].

loc, not follow

4 Choice of Oscillator

In bipedal locomotion research, the most used oscillators are the Hopf oscillator and the Matsuoka oscillator. The Hopf oscillator has been used successfully by Righetti and Ijspeert [27] and by Kieboom [16] to design controllers for a bipedal robot. Matsuoka oscillators have also been used for the same purpose by Taga [33], Ishiguro et al. [9] and by Cristiano et al. [4]. The Hopf oscillator is well defined in terms of amplitude and frequency, is easy to control and has strong limit cycle properties [16]. Moreover, different oscillators maybe easily combined to produce complex output patterns [27].

However, one disadvantage is that the Hopf oscillator needs an appropriate teaching signal to which the oscillator entrains to produce a suitable pattern. In [27], the authors have used the joint angle patterns produced by the manufacturer-provided walking algorithm as the teaching signal to train the oscillator network. The problem with this approach is that the CPG based walking algorithm will produce the same gait as the default algorithm, and will replicate ~~any~~ deficiencies of that algorithm. ~~For a new robot, a default walking algorithm may not be available.~~ In [16], the author has used a system of optimized sine generators as the teaching signal for a network of Hopf oscillators. Here additional effort is spent in optimizing the sine equation parameters, and again the teaching signals themselves are very simple and cannot efficiently utilize the robot's dynamics to their advantage.

Further more, The Matsuoka oscillator does not need a teaching signal, can be optimized in a single step to generate waveforms of a desired nature and can be entrained with the mechanics of the biped in an automatic sense [16]. Additionally, among the different oscillators, Matsuoka oscillators are the most plausible biologically and the two-neuron Matsuoka oscillator [18] bears a very close resemblance to the half-center model described in section 3. Keeping in mind the relative advantages of the Matsuoka oscillator over the Hopf oscillator, it has been chosen as the oscillator model for implementing the CPGs in this paper.

5 Matsuoka Oscillator

The Matsuoka oscillator is a mathematical model representing a general class of neural rhythm generators called mutual inhibition networks [18]. In this oscillator a set of neurons inhibit each others activities. For the individual neurons, a continuous-time continuous-variable neuron model is adopted [18]. The autonomous dynamics of the model are based on a piece-wise linear set of differential equations which model the dynamics of neural discharge rates. The oscillator consists of a pair of neurons whose rates of discharge are denoted by ψ_i and ψ_j respectively. The firing rates obey the following equations [28]:

$$t_1 \dot{\psi}_i = -\psi_i - \beta \phi_i - \eta [\psi_j]^+ + u_i \quad (1a)$$

$$t_1 \dot{\psi}_j = -\psi_j - \beta \phi_j - \eta [\psi_i]^+ + u_j \quad (1b)$$

where t_1 is the time constant for the rate of discharge, β is the self-inhibition constant and η is the constant of mutual inhibition. u_i and u_j denote the constant (tonic) input signals that drive the oscillator. ϕ_i and ϕ_j are state variables which control the self-inhibition. $[\bullet]^+ = \max(0, \bullet)$ implies that only the positive part of the discharge rates is considered for mutual inhibition. Equation 1 can be further extended by including terms representing the proprioceptive feedback, which will enable the system to monitor and rectify errors between the commanded and executed trajectories. A way of doing this was suggested in [36], by including the sensory feedback into the firing rate equations (equation 1). On doing so these equations are modified into:

$$t_1 \dot{\psi}_i = -\psi_i - \beta\phi_i - \eta[\psi_j]^+ - \sigma[\theta - \theta^*] + u_i \quad (2a)$$

$$t_1 \dot{\psi}_j = -\psi_j - \beta\phi_j - \eta[\psi_i]^+ - \sigma[-(\theta - \theta^*)] + u_j \quad (2b)$$

Here, σ is the strength of the sensory feedback, θ denotes the proprioceptive feedback and θ^* denotes the desired reference output angle (the mean position about which oscillations should occur). It should be kept in mind that this sensory feedback is not required for producing oscillations but only to make sure that the oscillations occur about a reference position. The state variables ϕ_i and ϕ_j obey the following equations [28]:

$$t_2 \dot{\phi}_i = -\phi_i + [\psi_i]^+ \quad (3a)$$

$$t_2 \dot{\phi}_j = -\phi_j + [\psi_j]^+ \quad (3b)$$

Here, t_2 is the time constant of adaptation or fatigue. The time constants t_1 and t_2 play an important role in the generation of oscillations. The discharge of a neuron with a constant input u will first rise to a maximum value in a time governed by t_1 and then gradually decrease to a lower value in a time governed by t_2 . Robust oscillations emerge due to a balance between self-inhibition and mutual-inhibition. The discharge rates ψ_i and ψ_j are assumed to be proportional to the torque generated in the positive and negative directions respectively [28]. Hence the net torque is given by:

$$\psi_T = h_\psi([\psi_i]^+ - [\psi_j]^+) \quad (4)$$

where h_ψ is the gain factor. The resultant torque can be converted into the desired joint velocity $\dot{\theta}$ and angular position θ by solving the following differential equation of a simple inertial mechanical system [28]:

$$I\ddot{\theta} + \gamma\dot{\theta} - \psi_T = 0 \quad (5)$$

Here, I is the moment of inertia and γ is the damping constant. A pictorial representation of a Matsuoka oscillator driving a single joint (with proprioceptive feedback) is shown in figure 2.

Since the basic equations of the Matsuoka oscillator are now in place, it is worthwhile to investigate how the parameters of the oscillator affect the output

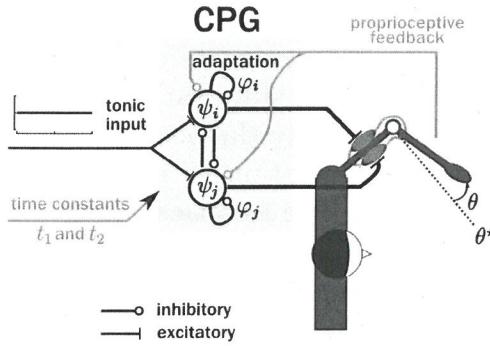


Figure 2: Matsuoka oscillator driving two antagonistic muscles. The difference of the torques produced by the two neurons is the net torque that drives the joint. Proprioceptive feedback is denoted by the gray line. Lines with empty circles at the end denote inhibitory connections [28].

pattern. This will lead to a better understanding of how to achieve desired movements of the robot by changing oscillator parameters. According to equations 1-4, the oscillator parameters are t_1 (time constant for the rate of discharge), t_2 (time constant of fatigue), β (self-inhibition constant), η (mutual-inhibition constant), u_i, u_j (tonic inputs, $u_i = u_j$), σ (strength of sensory feedback), θ^* (desired reference angular position) and h_ψ (torque gain). $\psi_i, \psi_j, \phi_i, \phi_j$ are the state variables. In each of the three examples below, one parameter is modulated while the others are kept constant. The constant parameters are $t_1 = 0.25$, $t_2 = 2.5 \times t_1$ as per [28], $\beta = 2.5$, $\eta = 2.5$, $u_i = u_j = 1$, $\sigma = 1.5$, $\theta^* = 0$ and $h_\psi = 5.0$. The modulated parameter and its value are mentioned for each example.

First the effect of changing the desired reference position on the angular trajectory is demonstrated. On its own, the Matsuoka oscillator produces oscillations about an arbitrary position, and the system itself is dynamically unstable. However, with the introduction of a feedback term (equation 2) and by modulating the parameter θ^* , the mean position of oscillations can be controlled [28] (figure 3). For the first 10 seconds $\theta^* = 0$ radians, from 10-20s, $\theta^* = 1$ radian and from 20-30s $\theta^* = -1$ radian. As illustrated by the joint position graph (figure 3(b)), the mean oscillation position changes and the transition is smooth. This kind of situation maybe encountered when the robot initially moves on a flat surface and then steps upon an inclined surface. At this point the mean oscillation position of some joints would change.

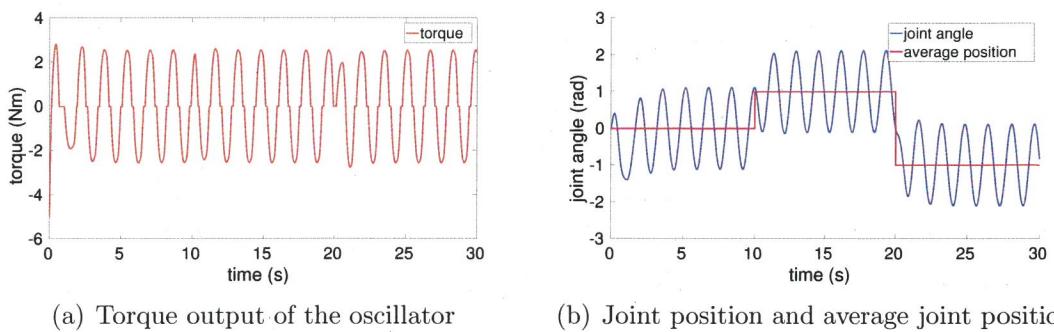


Figure 3: Changing the average position of oscillation

So far, the behavior of only a single oscillator driving a solitary joint has been described. In a robot multiple joints must interact with one another in a way that leads to a stable gait. To do so, the oscillators driving the individual joints must have a way of communicating with each other. This communication is determined by a coupling function which dictates how different oscillators can synchronize their frequencies or lock their phase [28]. To demonstrate the effects of coupling, two oscillators driving two different joints will be considered. For convenience, the two joints may be called the left joint (l subscript) and the right joint (r subscript). The state variable of the left joint are $\psi_{l,i}$, $\psi_{l,j}$, $\phi_{l,i}$, $\phi_{l,j}$, θ_l and $\dot{\theta}_l$ and the tonic inputs are $u_{l,i}$ and $u_{l,j}$. Similarly the variables for the right joint are $\psi_{r,i}$, $\psi_{r,j}$, $\phi_{r,i}$, $\phi_{r,j}$, θ_r and $\dot{\theta}_r$ and the tonic inputs are $u_{r,i}$ and $u_{r,j}$. The time constants for both the joints are assumed to be t_1 and t_2 . With this configuration, the coupling can be of three possible types [28]:

1. **Coupled at the input level:** If $u_{r,i} = u_{l,i}$ and $u_{r,j} = u_{l,j}$, then the coupling is done at the input level. However this kind of coupling does not allow the oscillators to converge to a definite phase and they continue behaving as two independent units [28].
2. **Coupled at the level of internal state variables:** Here the coupling is done by having a term for $\psi_{r,i}$ in the equation of $\psi_{l,i}$ and similarly a term for $\psi_{r,j}$ in the equation of $\psi_{l,j}$ (same for the equations of the right joint). Matsuoka had suggested this kind of coupling for a network of oscillators [18]. The disadvantage of this type of coupling is that in a network of several oscillators, all of them will either converge to in-phase or anti-phase oscillations, and a co-existence of in- and anti-phase relation is not possible [28]. In bipedal locomotion, the left hip joint moves in opposite phase to the right hip joint, but joints within the same leg move in-phase. Hence the coupling described here is not suitable for bipedal locomotion. *between legs only?*
3. **Coupling at the level of output variables:** If the coupling is done at the level of the output variables θ_l and θ_r , both in-phase and anti-phase relationships can be achieved within a network of oscillators [28]. This is the coupling configuration that has been used in this paper. The equations of the oscillator, in this case become (only the equations of the right joint are given below):

$$\begin{aligned} t_1 \dot{\psi}_{r,i} &= -\psi_{r,i} - \beta \phi_{r,i} - \eta [\psi_{r,j}]^+ - \sigma [\theta_r - \theta_r^*] \\ &\quad - \mu [\theta_l - \theta_l^*]^+ - \nu [-\theta_l + \theta_l^*]^+ + u_{r,i} \end{aligned} \tag{6a}$$

$$t_2 \dot{\phi}_{r,i} = -\phi_{r,i} + [\psi_{r,i}]^+ \tag{6b}$$

$$\begin{aligned} t_1 \dot{\psi}_{r,j} &= -\psi_{r,j} - \beta \phi_{r,j} - \eta [\psi_{r,i}]^+ - \sigma [\theta_r - \theta_r^*] \\ &\quad - \mu [\theta_l - \theta_l^*]^+ - \nu [-\theta_l + \theta_l^*]^+ + u_{r,j} \end{aligned} \tag{6c}$$

$$t_2 \dot{\phi}_{r,j} = -\phi_{r,j} + [\psi_{r,j}]^+ \tag{6d}$$

Next, the effect of changing the tonic input is demonstrated. The tonic input to the oscillator is linearly related to the amplitude of the torque output and also approximately linearly to the amplitude of the position oscillations. In this example, the tonic input (u_i, u_j) was linearly increased from 1.0 to 4.0 over 30 seconds, and as shown in figure 4, the amplitude of both the oscillations increases in a linear manner. This is possible in a situation where the robot needs to change the length of its stride to move over an obstacle or to step onto a stair. As is visible from the graphs, the transitions from one amplitude to the next are smooth.

what's interesting?

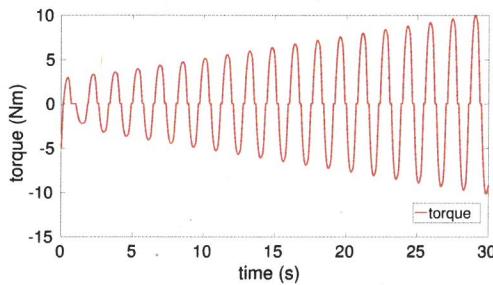
what would

happen if

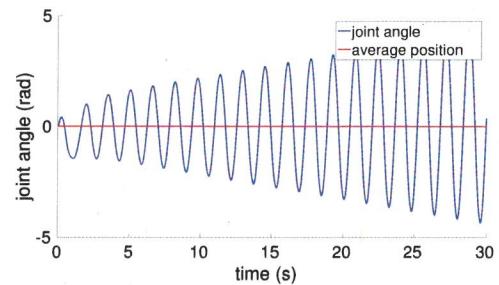
$U_1 \neq U_2$?

unsymmetrical

oscillations?



(a) Torque output of the oscillator

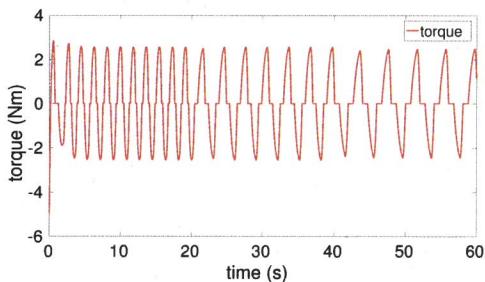


(b) Joint position and average joint position

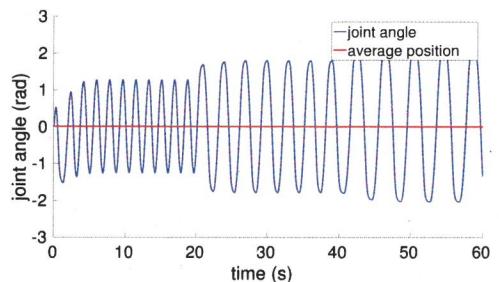
Figure 4: Changing the amplitude of oscillations

Apart from the mean position and amplitude, another key property of a waveform is its frequency. By varying the time constant of the oscillator the frequency of the torque and joint angle oscillations can be varied. In figure 5, the time constant t_1 was set to 0.3 for the first 20 seconds, 0.6 for the next 20 seconds and to 0.9 for the last 20 seconds (only t_1 was modulated, the parameter t_2 was set as $t_2 = 2.5 \times t_1$ as in [28]). As expected the frequency of both the torque and the joint angle oscillations decreases. However for the joint angles, the amplitude also increases as the frequency decreases. This effect maybe countered by gradually decreasing the tonic input as the frequency decreases in order to keep the amplitude constant. This kind of a situation can occur when the robot needs to change its speed of walking. Note that for both the transitions (at $t = 20s$ and $t = 40s$), the changes were smooth.

again



(a) Torque output of the oscillator



(b) Joint position and average joint position

Figure 5: Changing the frequency of oscillation

In this setup equations 4 and 5 remain unchanged. μ represents the constant for homologous coupling (since it is similar to the effect induced by σ but for the other joint) and ν is the constant for non-homologous coupling. All other symbols have the same meaning as in equations 1-3 [28].

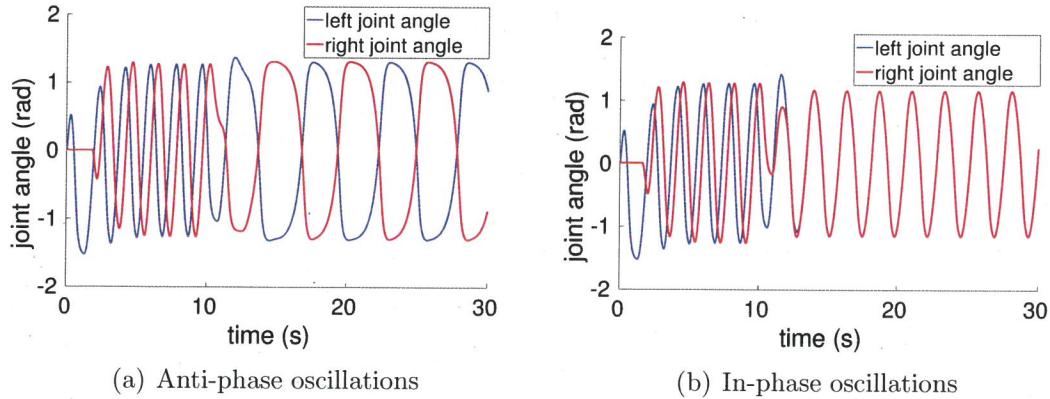


Figure 6: Anti-phase and in-phase oscillations

By using equations 6, 4 and 5, the behavior of a system of coupled oscillators can be demonstrated, as depicted in figure 6. The values of the constants used are $t_1 = 0.3$, $t_2 = 2.5 \times t_1$ as per [28] (same for left and right joints), $\beta = 2.5$, $\eta = 2.5$, $\sigma = 1.5$, $\theta^* = 0.0$, $h_\psi = 5.0$, $\mu = 0.75$, $\kappa = 0.49$ and $u_{l,i} = u_{l,j} = u_{r,i} = u_{r,j} = 1.0$. These values were selected based on [28]. For both the types of oscillations, the oscillator of the right joint is started a little while after the oscillator of the left joint, in order to introduce an initial phase difference. For 10 seconds, the two oscillators behave independently of each other and do not settle into any fixed phase-relationship. However as soon as the coupling is turned on at $t=10s$, the two oscillators converge to either anti-phase or in-phase oscillations. For anti-phase oscillations, the right oscillator is started 1.68s after the left oscillator and for the in-phase oscillation, it is started after 1.67s. Thus it is evident that the initial phase difference between the oscillators also plays a part in determining which kind of phase-relationship the oscillators converge to.

Another important characteristic of this system of coupled oscillators is that the final phase difference between the two oscillators can either be 0 (in-phase) or π radians (anti-phase). No other phase-relationship is possible. In order to demonstrate this characteristic, the system of two oscillators was tested with a range of different initial phase differences (using difference of activation times) and time periods of oscillation. An empirical formula ($T = 1.47t_1 + 2.92\sqrt{t_1} - 0.2304$) for calculating the joint oscillation time period in terms of the time constant t_1 was used [28]. The final phase difference between the two waveforms was calculated by using the Hilbert Transform. Figure 7 confirms that for any initial phase difference and any time period of oscillation, the oscillators converge to either in-phase or anti-phase oscillations. It can also be seen that the basin of attraction of in-phase oscillations is larger than that of anti-phase oscillations.

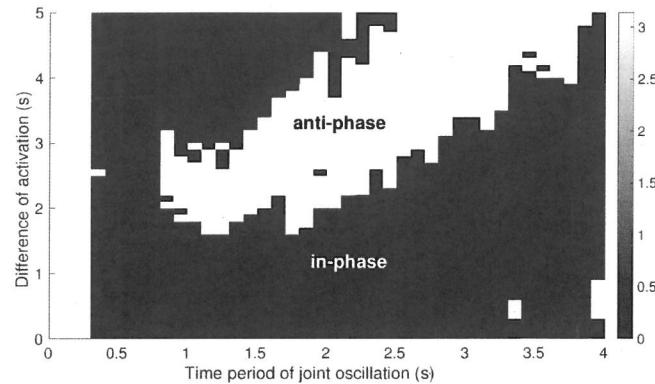


Figure 7: The joint oscillations converge to only 2 modes - either in-phase or anti-phase (based on [28]).

*Some last words summarizing
the current state of the scope*

6 Robot Joint Control

The concept of using CPGs in bipedal robots is not new. Several studies in the past have tried to utilize CPGs, particularly the Matsuoka oscillator, to develop walking algorithms. However, most of these studies such as [33], [23], [9] have used custom built torque driven robot models. Although these papers present useful insights into bipedal walking, it is very difficult to implement the methods on commercially available robots, almost all of which are position controlled (robot control is in terms of the joint angles and not in terms of joint torque). Some studies which do use commercial robots like the Nao, such as [4], have used Matsuoka oscillators in master-slave configurations to drive robot joints. In this case, utilizing the properties of the oscillator (as discussed in section 5) becomes very difficult. Therefore, in this paper, an effort has been made to use the Matsuoka oscillator in a novel configuration which can be implemented on a commercially available position controlled robot, and also utilizes the oscillator's properties.

If the equations of the system of coupled Matsuoka oscillators are investigated (equations 6 and 4), it can be seen that the output ψ_T of the oscillators is the torque that should be exerted at the joints, while the feedback is only in terms of the joint angle at a given time θ_l or θ_r . [In a torque based control system, the system generates target torques and uses the feedback in the form of actual torques to make the necessary corrections, but for the Matsuoka oscillator (as used here), this is not the case.] This is an important property which can be exploited to use the Matsuoka oscillator in a position controlled robot.

May be 1-2 sentences intro before closing in 2
Matsuoka oscillator driving a position controlled joint: As shown in figure 8, first the joint is initialized to 0 radians and then the oscillator parameters are set. The empirical formula for the oscillation time period ($T = 1.47t_1 + 2.92\sqrt{t_1} - 0.2304$ [28]) is used to derive the parameters t_1 and t_2 ; t_1 is calculated as $t_1 = 2.13 + 0.6804T - \sqrt{4.512 + 2.685T}$ and t_2 is calculated as $t_2 = 2.5 \times t_1$ (derivations are as per [28]). The other parameters are set to suitable values, and the time vector is also set (total duration and duration of each time step). After

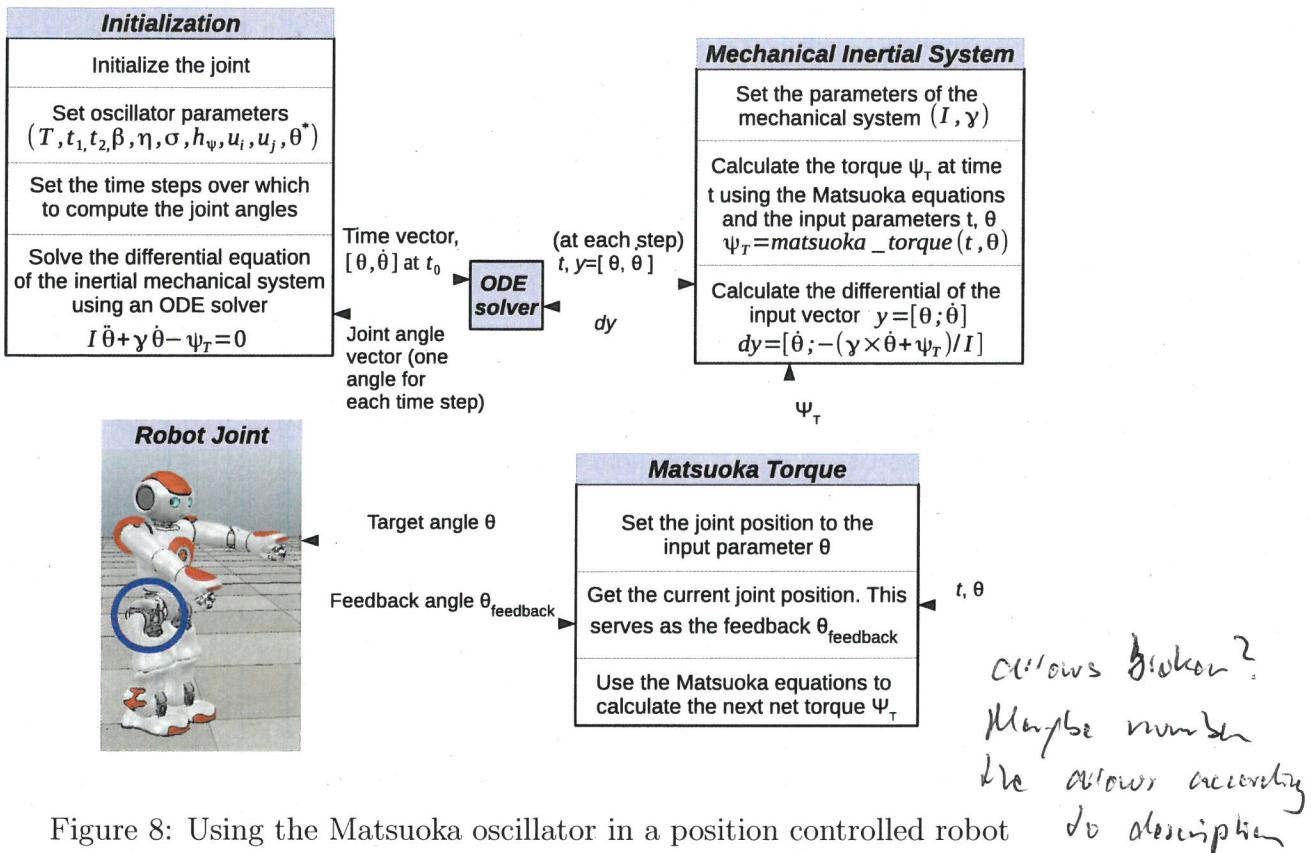


Figure 8: Using the Matsuoka oscillator in a position controlled robot

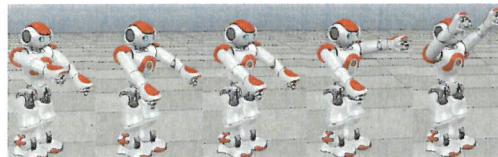
this an Ordinary Differential Equation (ODE) solver is used to solve the differential equation of the mechanical inertial system (equation 5). At each time step, the ODE solver calls a module which computes the derivative of $[\theta, \dot{\theta}]$ (vector made up of the angular position and velocity). To compute the derivative, the module of the mechanical inertial system calculates the net torque first. It passes the target angle to the Matsuoka torque module, which in turn sends it to the robot to be executed on the appropriate joint. The robot senses the actual angle of the joint ($\theta_{feedback}$) and the torque for the next step is computed using the sensed angle. This torque ψ_T is returned to the mechanical inertial system module, and finally the derivative of $[\theta, \dot{\theta}]$ is calculated and sent back to the ODE solver. The same process is repeated for each time step till the previously fixed final time is over.

The important thing to note about the process described above is that the computed torque ψ_T is not applied to the joint at all. It is treated like an intermediate variable that is only required to calculate the target joint angle θ . Use of the feedback angle $\theta_{feedback}$ ensures that the joint moves as desired. Hence the parameters of the inertial system (I and γ) can be set to any value (as long as they do not cause oscillations to disappear) and do not need to accurately model the physical joint properties. It is also worth mentioning that in this implementation, normally available ODE solvers such as the Runge-Kutta solvers cannot be used. The reason is that these solvers use variable time steps to optimize the process

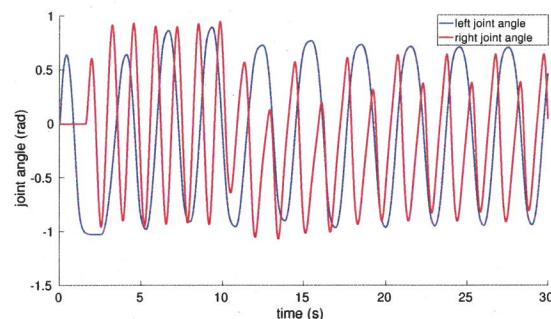
and sometimes traverse back in time to minimize the approximation error. In the current scenario, the ODE solver has to internally use the Matsuoka equations which depend on state variables with persistent values. Going back in time can corrupt these state variables leading to incorrect results. Hence, a custom ODE solver which uses fixed time steps, was used here. *hand-made?*

Using the process described in figure 8, multiple joints can also be controlled (with the help of equation 6). For testing this process, Matsuoka oscillators were used to control the shoulder joints of a Nao robot in simulation. The two shoulder joints were driven first in anti-phase and then in-phase, and the graphs similar to that of figure 6 were reproduced by using the robot. Apart from this, another kind of movement is important in bipedal robots. It is the movement of coupled joints which start in-phase and then one joint moves with a frequency that is a multiple of the other joint's frequency. This relationship exists between the joints of the knees and the hip, with the knees moving at twice the frequency of the hip. In order to show this effect on a stationary robot, one shoulder joint was moved at approximately twice the frequency of the other (figure 9). In this case, some parameters were different for the two oscillators (indicated by the use of an additional subscript - l for left and r for right). The parameters used were $T_l = 2.4$ ($t_{l,1}, t_{l,2}$ are calculated as described above), $u_{l,i} = u_{l,j} = 0.5$, $T_r = 1.2$ ($t_{r,1}, t_{r,2}$ are calculated as described above), $u_{r,i} = u_{r,j} = 1.0$, $\beta = 2.5$, $\eta = 2.5$, $\sigma = 1.5$, $\mu_l = 0.375$, $\mu_r = 0.75$, $\nu_l = 0.245$, $\nu_r = 0.49$, $h_\psi = 5.0$. All these values were arrived at empirically. The resulting motion is depicted in video number 7 in [1]. The other motions generated in the Nao can also be viewed in [1].

Usually I would give
weblinks as footnotes since
they are not strictly speaking
references.



(a) Snapshots from video# 7 in [1], joints moving with different frequencies



(b) Two joints moving in-phase but with different frequencies. Coupling is switched on at $t=10s$

Figure 9: Coupled joint oscillations in the robot

7 Future Work

In section 6 it was demonstrated that it is possible to control a position controlled robot using the Matsuoka oscillator. Using a similar setup a network of oscillators can be used to control the individual joints in a way that leads to a stable gait. Future work in this regard will involve determining the set of parameters that, when implemented on the network of oscillators, produces synchronized joint oscillations and helps the robot to walk on a flat, even surface.

A possible way of connecting several oscillators so as to form a network is shown in figure 10 (bottom-right). Here each oscillator consists of 2 half center neurons as depicted in figure 1(b). An oscillator may be coupled to one or more oscillators by means of the coupling constants μ and ν as used in equation 6. Each oscillator contains a large number of parameters and hence for a network of 10 oscillators together with the associated couplings; the number of parameters is quite large. In order to reduce the parameter space, the symmetry inherent in bipedal locomotion can be utilized. Hence the parameters for an oscillator of the left leg can be made equal to the parameters of the corresponding oscillator from the other leg ($L1$ with $R1$ and so on). For the couplings too, a similar approach can be followed.

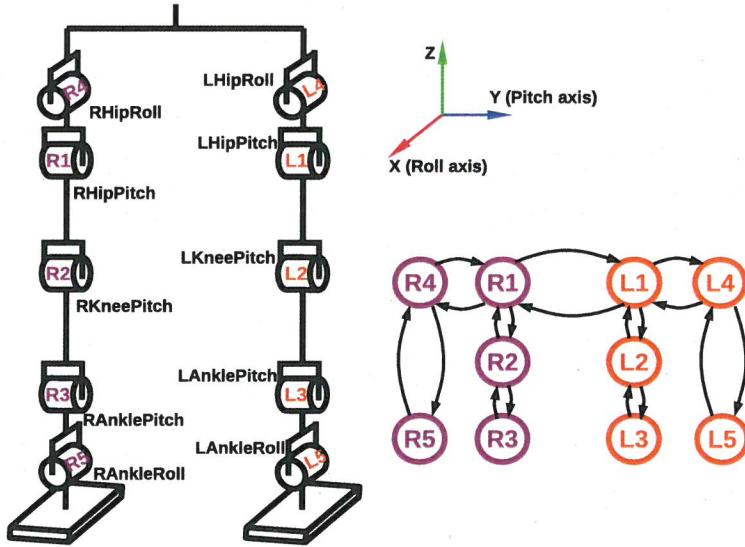


Figure 10: (left) The joints in the legs of the Nao robot which are used for walking. Labels beside the joints denote the joint names and the coloured labels on the joints denote the oscillator which controls the joint. (top-right) Joint rotation axes. (bottom-right) The oscillator network which will control the joints. Arrows denote the coupling between different oscillators.

Separate lines from text

Once a suitable parameter set has been identified, optimization algorithms can be used to find parameter values that result in a stable walk. For this, the oscillator network will be initialized with different parameter values and the robot will be tested in simulation in an attempt to find the ideal set of parameter values.

For the optimization, Particle Swarm Optimization [5, 30, 15] and a few of its variants such as the ATRE-PSO (a type of diversity guided Attraction-Repulsion PSO) [24] and QI-PSO (Quadratic Interpolation PSO) [25] can be used. Since the number of parameters is quite high, ATRE-PSO and QI-PSO are suitable choices, as they have been shown to perform well on high dimensional problems.

If the PSO algorithms are successful in discovering a stable gait, then the next step will be to implement a balance controller that will utilize real time information from the robot's sensors (both proprioceptive sensors and other integrated sensors such as the gyroscope or accelerometer) to make minor modifications to the robot's gait so that it can stabilize itself when dealing with an uneven surface or with random forces. The modifications will be in the form of minor adjustment to the oscillator parameters that have been optimized by the PSO algorithms. A synergy between the oscillator network, balance controller and the robot hardware will hopefully result in a system that is able to tackle both even terrain and moderately uneven terrain.

I would switch 7 and 8 so that you
first conclude the work in this paper and then
go into future work

8 Conclusion

Central Pattern Generators such as the Matsuoka oscillator are potentially an excellent alternative to the widely accepted ZMP based techniques of bipedal walking. CPGs are beneficial in many ways, the most important being their ability of exploiting the natural dynamics of the mechanical structure instead of suppressing it. Using a hierarchical walking controller, where the CPGs are responsible for pattern generation and form the lowest level, has the added advantage of making things very simple for the higher level balance controller. The balance controller only needs to convey control signals for overall posture and walking direction and does not need to bother with low level joint trajectories.

Although the Matsuoka oscillator has been around for a few decades, its use was somewhat restricted by the fact that most commercially available robots are position controlled and the Matsuoka oscillator generates torques which cannot be directly used in such robots. With the development of the technique discussed in this paper this limitation can be overcome. What is also interesting in this respect is that a CPG based framework for walking can be easily adapted to any biped robot and is not tightly coupled with a particular hardware setup. Thus very little additional effort will be required to create a similar setup for different robots.

Biologically inspired techniques have been used to solve many intractable problems in artificial intelligence and robotics. From vision to language understanding, scientists have often replicated mechanisms found in animals to achieve their objectives. For bipedal locomotion too, emulating nature is a promising step forward that we can take so that our robots can one day walk as gracefully and effortlessly as we can.

The last part reads almost like an abstract. First sentence introduction
The ~~last~~ last sentence should be ¹⁶ the take-home message of your
paper. Currently it's too general.

Vin,
general