
ATTRACTOR SELECTION IN NONLINEAR ENERGY HARVESTING USING DEEP REINFORCEMENT LEARNING

A PREPRINT

Xue-She Wang & Brian P. Mann

Dynamical Systems Research Laboratory
Department of Mechanical Engineering & Materials Science
Duke University
Durham, NC 27708, USA
xueshe.wang@duke.edu

January 11, 2022

ABSTRACT

Recent research efforts demonstrate that the intentional use of nonlinearity enhances the capabilities of energy harvesting systems. One of the primary challenges that arise in nonlinear harvesters is that nonlinearities can often result in multiple attractors with both desirable and undesirable responses that may co-exist. This paper presents a nonlinear energy harvester which is based on translation-to-rotational magnetic transmission and exhibits coexisting attractors with different levels of electric power output. In addition, a control method using deep reinforcement learning was proposed to realize attractor switching between coexisting attractors with constrained actuation.

Keywords Energy Harvesting · Coexisting Attractors · Attractor Switching · Reinforcement Learning · Machine Learning · Nonlinear Dynamical System

1 Introduction

For small and independent devices where replacing a battery or connecting to a power grid is unrealistic, such as environmental monitoring systems [1, 2], medical implants [3, 4] and structural health monitoring sensors [5, 6], vibratory energy harvesters are useful to realize self-powering by converting mechanical energy to electrical energy with electromechanical coupling, e.g. piezoelectric, electromagnetic, electrostatic transduction, etc. Traditional linear energy harvesters which operate based on linear resonance work well only when excitation frequency is close to the natural frequency. When the excitation frequency is known a priori, the geometry and dimensions of a linear harvester can be carefully selected to match its resonant frequency to the excitation frequency. However, when the excitation frequency is unknown or varies in different operational conditions, the harvester with pre-tuned resonant frequency is unable to achieve optimal power output [7].

Many of the control methods applied to energy harvesters have primarily focused on resonant frequency tuning, which actively or passively matches the resonant frequency to environmental forcing frequency to increase harvesting efficiency. The active methods require continuous power input for resonance tuning. While for the passive methods, intermittent power is input for tuning and no power is required when frequency matching is completed, that is until the excitation frequency varies again [8]. Resonance tuning methods can be categorized into mechanical, magnetic, and piezoelectric methods. Mechanical methods are usually developed based on elementary of vibration theory: the resonance of a system can be tuned by changing the stiffness or mass [9, 10]. Magnetic method is an option for resonance tuning by applying magnetic force to alter the effective stiffness of a harvester [11, 12]. Piezoelectric transducers provide another option for resonance tuning based on the fact that the stiffness of the piezoelectric material itself can be varied with various shunt electrical load [13, 14].

Although the aforementioned control systems were implemented for tuning energy harvesters, they were not completely self-powered. The power required to apply control actions is oftentimes greater than the power harvested, thus an

external power supply is still needed and they are only suitable for vibration scenarios where the forcing frequency changes infrequently [15]. An alternate solution to achieving broadband frequency response is using nonlinear energy harvesters. Several recent works have suggested the intentional use of nonlinearity might be beneficial to energy harvesting systems [16, 17, 18, 19]. These studies have explored the use of nonlinearities to broaden the frequency spectrum, extend the bandwidth, engage nonlinear resonances, and/or to facilitate tuning [20, 21, 22, 23, 24, 25, 26].

While these investigations, along with many other recent works, have advanced the current understanding on the beneficial use of nonlinearity, the introduction of nonlinearity can also cause many additional difficulties. Paramount amongst these challenges, and a common issue in nearly all nonlinear harvesting systems, is the presence of coexisting solutions [27]. More specifically, for a certain environmental excitation, there exists multiple responses of a harvesting system and thus various levels of power generation [15]. These responses are stable steady-state oscillations, thus also considered “attractors” in nonlinear dynamics. An energy harvester generally prefers running on a higher-power attractor for faster energy harvesting, but would also need a lower-power attractor for safety reasons or physical restrictions. When one of the attractors is desirable and the other undesirable, it becomes critically important to have a control method to select the desired attractor with minimal energy expenditure.

Unlike conventional linear control problems where the system model is mathematically well-described or weakly nonlinear control problems (e.g. Lyapunov’s function can be easily found), the control method of switching attractors should push the system far away from an equilibrium thus a highly nonlinear behavior is inevitable. Conventional control techniques, such as proportional-integral-derivative (PID) control and linear quadratic regulator (LQR) control, may not provide a robust control. In addition, the attractor switching for an energy harvester needs 1) to apply constraints on control input, and 2) an optimal control method to minimize the energy consumed during the control process. For example, the limitations on the instantaneous power/force and total energy/impulse of a controller need be considered in practice. Another practical consideration is the optimization of total time and energy spent on the control process. Switching attractors using as little energy or time as possible is oftentimes required, especially when attempting to escape detrimental responses or in the case of a finite energy supply.

Fortunately, a technique that is compatible with a broader scope of nonlinear systems, reinforcement learning (RL), can be applied without the aforementioned restrictions. By learning action-decisions while optimizing the long-term consequences of actions, RL can be viewed as an approach to optimal control of nonlinear systems [28]. Various control constraints can also be applied by carefully defining a reward function in RL [29]. In recent years, a large number of advanced RL algorithms have been created to address complex control tasks, including benchmark works using MuJoCo physics simulator [30, 31, 32, 33, 34], motion planning of robotics [35], autonomous driving [36], and active damping [37]. Especially for the investigation of attractors in nonlinear dynamical systems, Ijspeert et al. proposed a generic modeling approach for attractor behaviors of autonomous nonlinear dynamical systems with the help of statistical learning techniques [38]. Wang et al. realized switching between coexisting attractors in a hardening Duffing oscillator using two RL algorithms: the cross-entropy method (CEM) and deep deterministic policy gradient (DDPG) [27]. Several researchers also have explored RL-based optimal control for gene regulatory networks (GRNs), with the goal of driving gene expression towards a desirable attractor while using a minimum number of interventions [39, 40, 41, 42].

The paper proposes a novel nonlinear energy harvesting system and a RL-based control method to switch between the attractors with different energy output levels. The content of this paper is organized as follows. Section 2 presents the mathematical model of an energy harvester which is based on a translational-to-rotational magnetic transmission and electro-magnetic coupling. In Section 3, two types of coexisting attractors were observed with different levels of electric power output: high-power attractor and low-power attractor. Section 4 presents the reinforcement learning framework designed for attractor switching of an energy harvester. Several key terms in RL were defined, including “environment”, “action”, “state & observation” and “reward”. Sections 5 and 6 provide two controller designs based on a linear actuator and motor voltage respectively. Simulation results show time-series response of switching between attractors. Conclusions and application potentials are discussed in Section 7.

2 Energy Harvester Design

2.1 Mechanical System

The energy harvester is based on a non-contact translational-to-rotational magnetic transmission in Ref. [43, 44]. As shown in Fig. 1, the drive magnet is fixed to a vibration source and has a prescribed translational harmonic motion. The driven magnet is pinned and limited to pure rotation about its center of mass. The drive magnet applies a non-contact magnetic force to the driven magnet to achieve translational-to-rotational conversion. In addition to the magnetic force, the driven magnet is also subject to a linear restoring torque from mechanical springs and a damping torque which is assumed to be viscous and proportional to the driven magnet’s angular velocity.

As derived in Ref. [43], the equation of motion can be written as a linear mass-spring-damper system driven by magnetic torque:

$$J\ddot{\theta} + c\dot{\theta} + k(\theta - \theta_0) = \alpha \left[\frac{\sin \theta}{[(b + A \cos(\Omega t))^2 + h^2]^{3/2}} - \frac{3[b + A \cos(\Omega t)][h \cos \theta + (b + A \cos(\Omega t)) \sin \theta]}{[(b + A \cos(\Omega t))^2 + h^2]^{5/2}} \right], \quad (1)$$

where in the left-hand side, J , c , k and θ_0 are the driven magnet's moment of inertia, torsional spring coefficient, torsional damping coefficient, and offset bias angle of spring respectively. The right-hand side is the expression of magnetic torque, which is dependent on the angle of the driven magnet θ , the vertical distance between two magnets h , and the horizontal distance between two magnets $b + A \cos(\Omega t)$. A constant $\alpha = \mu_0 M_0 V_0 M_1 V_1 / 4\pi$ describes the magnetic properties of magnets, where μ_0 is the permeability of free space, M_0 and V_0 are the drive magnet's magnetization and volume respectively, and M_1 and V_1 are the driven magnet's magnetization and volume respectively.

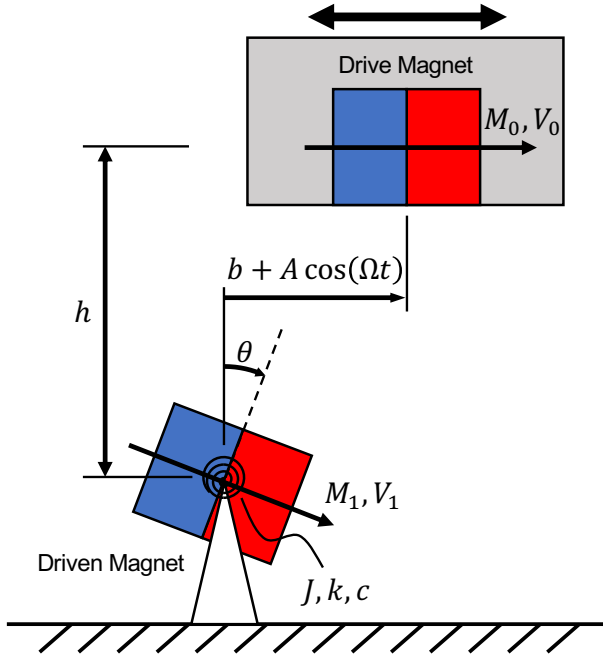


Figure 1: Schematic diagram of the non-contact translational-to-rotational magnetic transmission system.

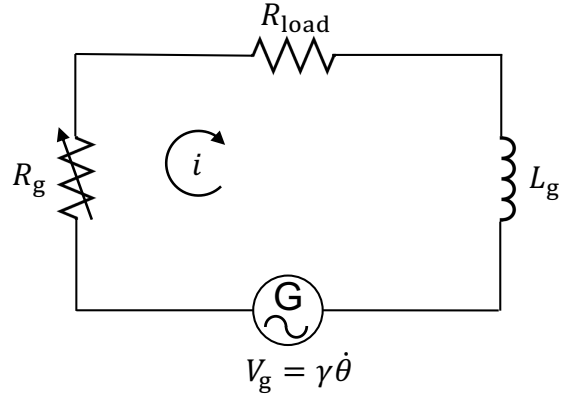


Figure 2: Schematic of the coupled electrical circuit in the generator. R_g and L_g are the resistance and the inductance inside the generator respectively. V_g is the voltage induced by the rotary motion from the mechanical system.

2.2 Electro-Mechanical Coupling

The mechanical system in Fig. 1 and Eq. (1) transmits the vibrational source's translational motion to the driven magnet's rotational motion. This rotation is then conveyed to an electro-magnetic transducer, such as a generator, to produce electric power. For simplicity, as shown in Fig. 2, the voltage induced by the electro-magnetic transducer V_g is assumed to be linearly dependent on the driven magnet's angular velocity $\dot{\theta}$, and the transducer connects to a simple resistor load R_{load} . As derived in Ref. [45], the governing equation for the energy harvester is comprised of 1) the mechanical system's governing equation Eq. (1) where a coupling term γi was introduced, and 2) an additional equation for the electrical circuit:

$$J\ddot{\theta} + c\dot{\theta} + k(\theta - \theta_0) - \boxed{\gamma i} = \alpha \left[\frac{\sin \theta}{[(b + A \cos(\Omega t))^2 + h^2]^{3/2}} - \frac{3[b + A \cos(\Omega t)][h \cos \theta + (b + A \cos(\Omega t)) \sin \theta]}{[(b + A \cos(\Omega t))^2 + h^2]^{5/2}} \right], \quad (2)$$

$$L_g \dot{i} + (R_g + R_{load}) i + \boxed{\gamma \dot{\theta}} = 0 \quad (3)$$

Table 1: Parameters of the energy harvester.

Parameter	Symbol	Value
Driven Magnet		
Driven magnet's moment of inertia	J	$1.11 \times 10^{-6} \text{ kg} \cdot \text{m}^2$
Spring torsion coefficient	k	$5.48 \times 10^{-3} \text{ N} \cdot \text{m/rad}$
Damping torsion coefficient	c	$3.02 \times 10^{-6} \text{ N} \cdot \text{m} \cdot \text{s/rad}$
Natural frequency without damping	ω_n	70.25 rad/s
Offset bias angle	θ_0	0 rad
Excitation		
Excitation amplitude	A	3 mm
Excitation frequency	Ω	50.24 rad/s (8 Hz)
Excitation horizontal bias	b	0 mm
Vertical distance between magnets	h	34 mm
Magnet Properties		
Permeability of free space	μ_0	$4\pi \times 10^{-7} \text{ H/m}$
Residual flux density	B_r	1.32 Tesla
Magnetization	$M_0, M_1 = B_r/\mu_0$	$1.05 \times 10^6 \text{ A/m}$
Magnet radius	r_{mag}	6.35 mm (1/4 in)
Magnet height	h_{mag}	12.7 mm (1/2 in)
Magnet volume	$V_0, V_1 = \pi r_{\text{mag}}^2 h_{\text{mag}}$	1608.8 mm^3
Circuit		
Generator inductance	L_g	1 H
Generator resistance	R_g	0.1Ω
Electric load	R_{load}	5Ω
Coupling coefficient	γ	0.06

where R_g and L_g are the resistance and the inductance inside the generator respectively. γ is the electro-mechanical coupling term and i is the current induced by the rotary motion from the mechanical system. The resistor load R_{load} was used to evaluate the power output. In order to ensure the consistency of the energy harvester throughout this paper, its parameters were measured experimentally and then used for the following numerical and analytical investigations. The value of parameters can be found in Tab. 1.

3 Coexisting Attractors

The presence of coexisting attractors is a challenge for nonlinear energy harvesting. Integrating the governing equations Eq. (2) and (3) with respect to time results in steady-state solutions after initial transient dissipates. These solutions are stable thus they are also considered "attractors" in dynamical systems. As shown in Fig. 3, different initial conditions result in three coexisting steady-state oscillations: one with larger amplitude and two symmetric oscillations with smaller amplitudes. Their response frequencies are all equal to the excitation frequency (the excitation is the motion of vibrational source, $b + A \cos \Omega t$).

The energy harvester has three state variables: the driven magnet's angle θ , angular velocity $\dot{\theta}$ and the induced current i . Thus the phase portraits of steady-state oscillations are closed cycles in 3-dimensional phase domain (see Fig. 4). The phase portraits include two symmetric smaller cycles and one larger cycle. Given that their response frequencies are equal, the cycle with the larger current amplitude corresponds to high-power output (considered **HP** attractor), while the two cycles with the smaller current amplitudes correspond to low-power output (considered **LP** attractor).

As for an energy harvester, running on the HP attractor is generally more desirable, but sometimes LP attractor could also be needed for safety reasons or physical restrictions. The following sections will introduce a control method to switch between coexisting attractors of the energy harvester (see Fig.5).

4 Reinforcement Learning

4.1 Framework

Reinforcement Learning (RL), which can be viewed as an approach to optimal control of nonlinear systems, learns action-decisions while optimizing the long-term consequences of actions. In the RL framework, an *agent* gains

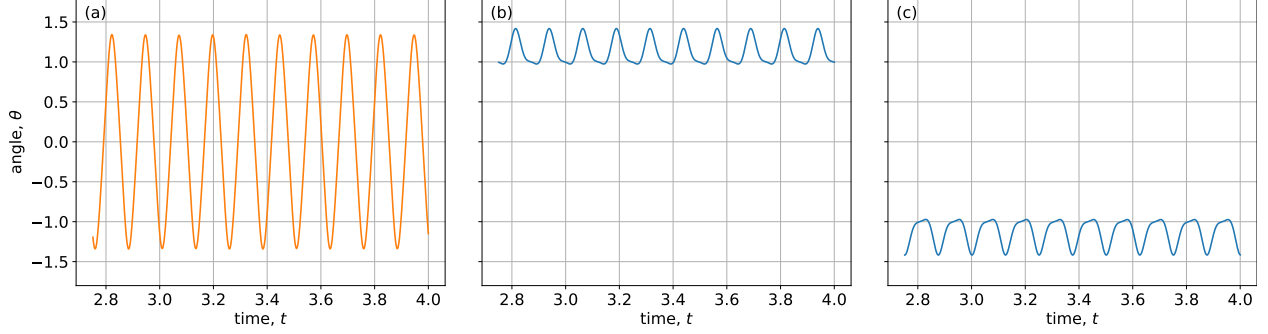


Figure 3: Coexisting steady-state oscillations of the driven magnet's angle θ . (a) has a larger oscillation amplitude while (b) and (c) oscillate around symmetric offset angles with smaller amplitudes. They result from different initial conditions: $[\theta_0, \dot{\theta}_0, i_0] =$ (a) $[-1.15, -38, 0.07]$, (b) $[1.0, -1.4, 0.008]$, (c) $[-1.0, 1.4, -0.008]$.

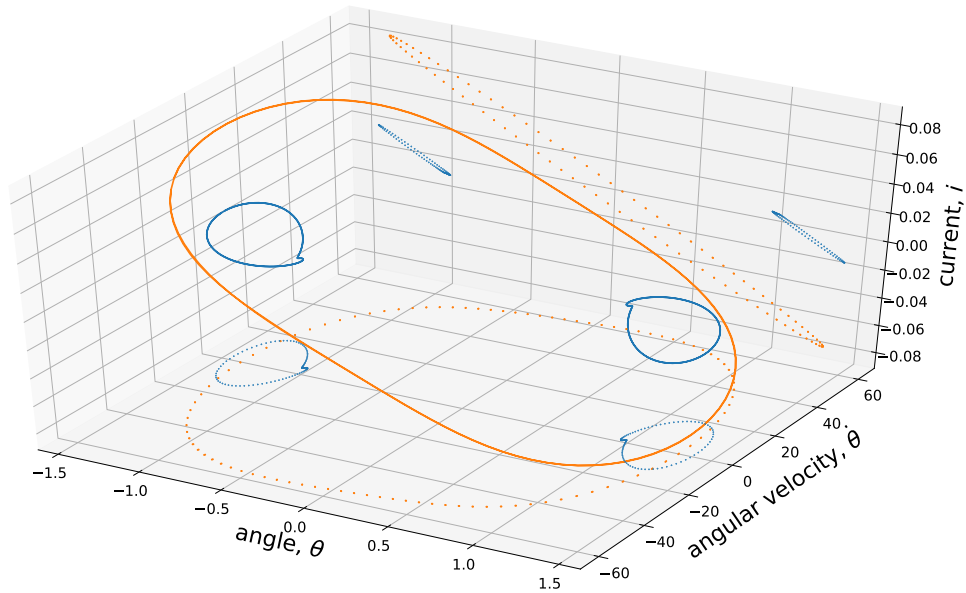


Figure 4: Phase portrait of steady-state oscillations of the energy harvester. The orange cycle with a larger oscillation amplitude represents high-power output (HP attractor) while the two symmetric blue cycles with smaller oscillation amplitudes represent low-power output (LP attractor). The dotted lines are the projections of attractors on 2D planes (top view and side view).

experience by making *observations*, taking *actions* and receiving *rewards* from an *environment*, and then learns a *policy* from past experience to achieve goals (usually maximized cumulative *reward*). This section implements RL algorithms for the energy harvester to switch between its low-power (LP) and high-power (HP) attractors using minimized energy consumed on the controller.

Environment is represented by the energy harvester's governing equation. As shown in Fig. 3 and 4, for certain ranges of the parameters, the energy harvester will always eventually settle into one of the stable attractors. Our objective is to apply control to the system to make it switch between LP and HP attractors using constrained actuation. To provide actuation for attractor switch, the velocity of the linear actuator is controlled by $a(s)$, which depends on the energy harvester's states s .

Action. Aligned with the practical consideration that an actuation is commonly constrained, the action term can be written as $a(s) := F\pi_\eta(s)$, where F is the action bound which denotes the maximum absolute value of the actuation, and $\pi_\eta(s)$ is the control policy. $\pi_\eta(s)$ is designed to be a function parameterized by η , which has an input of the energy harvester's states s , and an output of an actuation scaling value between -1 and 1 . Our objective is achieved by finding qualified parameters η that cause the desired attractor to be reached.

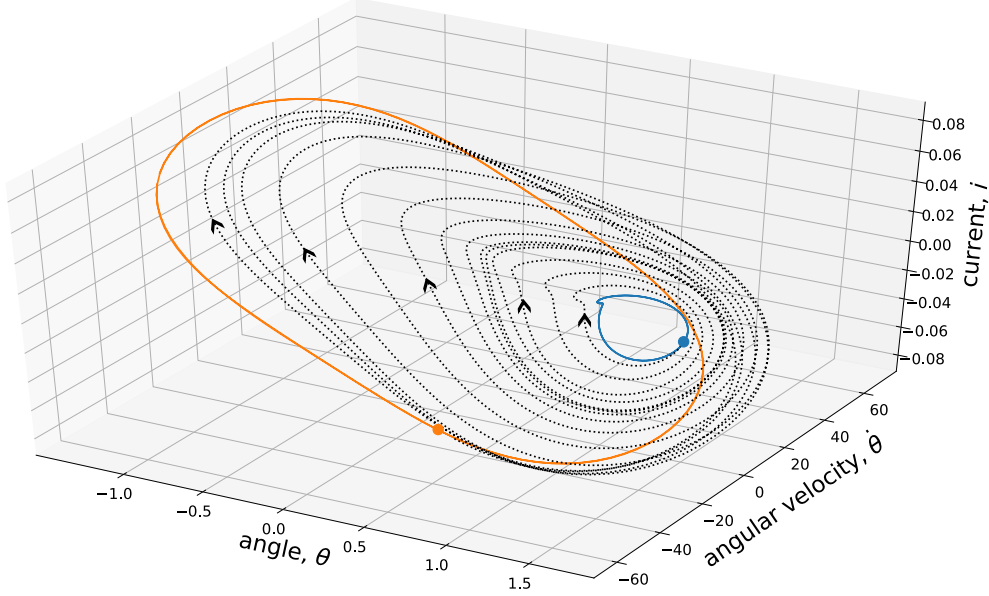


Figure 5: Attractor switching form LP to HP attractor by using the proposed control method.

State & Observation. The energy harvester is driven by a time-varying excitation $A \cos \Omega t$; thus, the state should at least consist of time t , angle θ , angular velocity $\dot{\theta}$, and induced current i . Given that $A \cos \Omega t$ is a sinusoidal function with a phase periodically increasing from 0 to 2π , time can be replaced by phase for a simpler state expression. The system’s state can therefore be written as $s := [\phi, \theta, \dot{\theta}, i]$, where ϕ is equal to ωt modulo 2π . Additional state variable(s) might be needed according to the definition of action a . Multiple definitions of the control input are introduced in the next section, where more details of state variables are discussed. For the sake of simplicity, we have assumed that no observation noise was introduced and the states were fully observable by the agent.

Reward. A well-trained policy should use a minimized control effort (i.e. energy consumed on a controller) to reach the target attractor; thus the reward function, which is determined by state s_t and action a_t , should inform 1) whether the energy harvester reaches the target attractor, and 2) the cost of the action taken. The environment estimates whether the target attractor will be reached by evaluating the next state s_{t+1} . A constant reward of r_{end} is given only if the target attractor will be reached. Similar to the state in RL, the action cost, r_{cost} , differs in the definition of action a , thus the detail of calculating the cost is presented along with the specific controller design in the next section. A generic reward function for attractor selection can be written as:

$$r(s_t, a_t) = -r_{\text{cost}} + \begin{cases} r_{\text{end}}, & \text{if } s_{t+1} \text{ reaches the target attractor (or BoA)} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

For estimating whether the target attractor will be reached, one could observe whether s_{t+1} is in the “basin of attraction” of the target attractor. Basins of attraction (BoA) are the sets of initial states leading to their corresponding attractors as time evolves. Once the target BoA is reached, the system will automatically settle into the desired stable trajectory without further actuation.

4.2 Basin of Attraction (BoA) Prediction

Determining whether a state is in the target BoA is non-trivial. For an instantaneous state $s_t = [\phi_t, \theta_t, \dot{\theta}_t, i_t]$, we could integrate the governing equation with the initial condition $[\phi_0, \theta_0, \dot{\theta}_0, i_0]$. Integrating for a sufficiently long time should give a steady-state response, whose amplitude can be evaluated to determine the attractor where the system will eventually settle down. However, this prediction is needed for each time step of the RL learning process, and the integration time should be sufficiently long to obtain steady-state responses; thus this approach results in expensive computational workload and a slow learning process. As a result, a more efficient method was needed for determining which resting attractor corresponded to the system’s state [46].

From the observation of coexisting attractors, there are two levels of power output for the energy harvester. Since the number of levels is finite, the attractor prediction can be considered a classification problem, where the input is the

system's state and the output is the resting attractor where the system will settle down under no control. Given that the system's state has four dimension and the boundary of BoA tends to be nonlinear (or at least not guaranteed to be linear for the energy harvester), the classification algorithm using neural network was selected for its capability of handling high-dimensional nonlinear classification. As shown in Fig. 6, a 4-layer neural network was trained (or called "fitted") to predict the resting attractor (HP or LP) given an initial condition $[\phi_0, \theta_0, \dot{\theta}_0, i_0]$. The training data was created by sampling states randomly on the domain of four state variables, and the corresponding resting attractor was determined by the method mentioned above: evaluating future responses with long-term integration of governing equation. Generally speaking, this method transfers the recurring cost of integration during the RL learning process to a one-time cost before the learning process begins. Collecting training data can be time consuming (running millions of simulation), but once the classifier is well-trained, the time for predicting the resting attractor can be negligibly small.

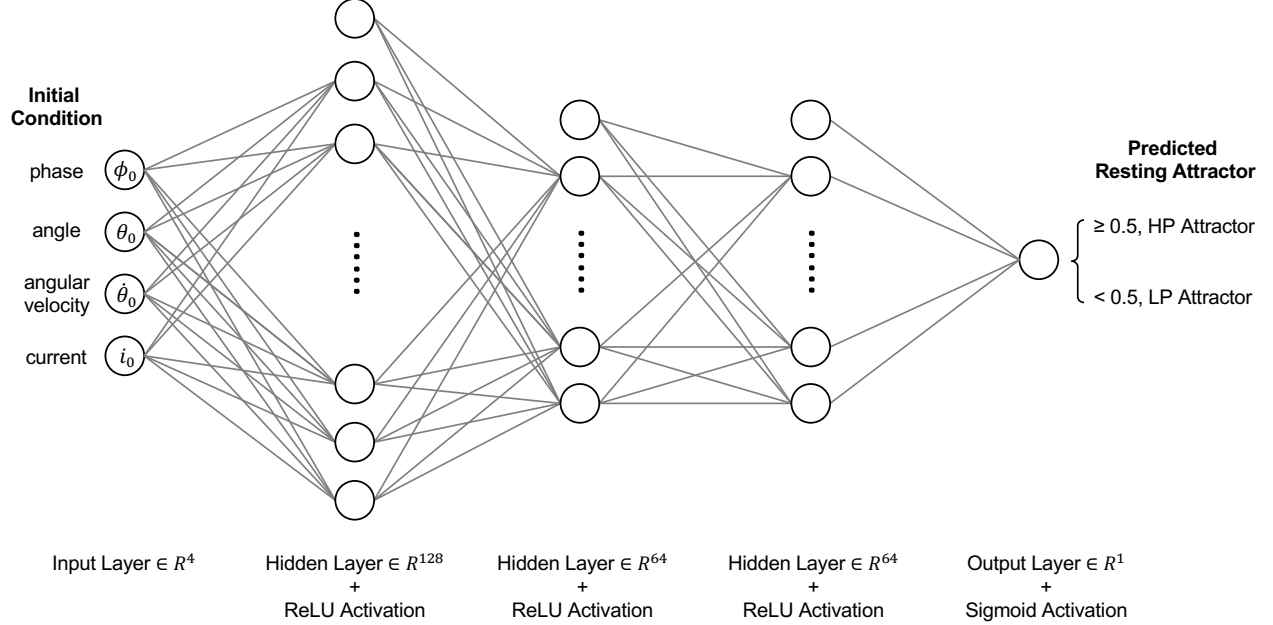


Figure 6: Neural network diagram for predicting the resting attractor (HP or LP) given an initial condition of the energy harvester $[\phi_0, \theta_0, \dot{\theta}_0, i_0]$. The network has four layers: 128 neurons (ReLU) – 64 neurons (ReLU) – 64 neurons (ReLU) – 1 neuron (sigmoid). The ReLU (Rectified Linear Unit) activations introduce nonlinearity to this classifier. The sigmoid activation transforms its input into a value between 0.0 and 1.0, thus predicting the probability of HP attractor as an output. Therefore, when the output value is greater or equal than 0.5, the initial condition is predicted to lead the energy harvester to settle down on the HP attractor, otherwise the LP attractor.

4.3 Algorithm

Amongst various RL algorithms, Deep Deterministic Policy Gradient (DDPG) was selected for its capability of operating over continuous state and action spaces [47]. The goal of DDPG is to learn a policy which maximizes the expected return $J = \mathbb{E}_{r_i, s_i, a_i} [R_{t=1}]$, where the return from a state is defined as the sum of discounted future rewards $R_t = \sum_{i=t}^T \gamma^{i-t} r(s_i, a_i)$ with a discount factor $\gamma \in [0, 1]$.

An action-value function, also known as a “critic” or “Q-value” in DDPG, is used to describe the expected return after taking an action a_t in state s_t :

$$Q(s_t, a_t) = \mathbb{E}_{r_i \geq t, s_i > t, a_i > t} [R_t | s_t, a_t]. \quad (5)$$

DDPG uses a neural network parameterized by ψ as a function approximator of the critic $Q(s, a)$, and updates this critic by minimizing the loss function of the difference between the “true” Q-value $Q(s_t, a_t)$ and the “estimated” Q-value y_t :

$$L(\psi) = \mathbb{E}_{s_t, a_t, r_t} [(Q(s_t, a_t | \psi) - y_t)^2], \quad (6)$$

where

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \pi(s_{t+1}) | \psi). \quad (7)$$

Table 2: Algorithm: Deep Deterministic Policy Gradient (DDPG) for learning a control policy for attractor switching

1	Determine the governing equation: Eq. (12) for the actuator control, or Eq. (14) for the voltage control	
2	Determine the reward function: Eq. (13) for the actuator control, or Eq. (15) for the voltage control	
3	Randomly initialize actor network $\pi_\eta(s)$ and critic network $Q_\psi(s, a)$ with weights η and ψ	
4	Initialize target network $\pi'_{\eta'}(s)$ and $Q'_{\psi'}(s, a)$ with weights $\eta' \leftarrow \eta$, $\psi' \leftarrow \psi$	
5	Set the initial condition of the Duffing equation $s_0 = [x_0, \dot{x}_0, \phi_0]$	
6	Set discount factor γ , and soft update factor τ	
7	Set time of Phase 1 and Phase 2, T_1 and T_2	
8	Initialize replay buffer B	
9	for episode = 1 : M do	
10	Initialize a random process \mathcal{N} for action exploration	
11	Add noise to time of Phase 1, $T_1' = T_1 + \text{random}(0, 2\pi/\omega)$	
12	Integrate the governing equation for $t \in [0, T_1']$ with $a(s) = 0$	Phase 1
13	for $t = T_1' : T_1' + T_2$ do	Phase 2
14	Observe current state, $s_t = [x_t, \dot{x}_t, \phi_t]$	
15	Evaluate action $a_t(s_t) = F\pi_\eta(s_t) + \mathcal{N}_t$, according to the current policy and exploration noise	
16	Step into the next state s_{t+1} , by integrating the governing equation for Δt	
17	Evaluate reward $r_t(s_t, a_t)$ from the reward function	
18	Store transition $[s_t, a_t, r_t, s_{t+1}]$ in B	
19	Sample a random minibatch of N transitions $[s_i, a_i, r_i, s_{i+1}]$ from B	
20	Set $y_i = r_i + \gamma Q'_{\psi'}(s_{i+1}, F\pi'_{\eta'}(s_{i+1}))$	
21	Update the critic network by minimizing the loss:	
22	$L = \frac{1}{N} \sum_i (y_i - Q_\psi(s_i, a_i))^2$	
23	Update the actor network using the sampled policy gradient:	
24	$\nabla_\eta J \approx \frac{1}{N} \sum_i \nabla_a Q_\psi(s, a) _{s=s_i, a=F\pi_\eta(s_i)} \nabla_\eta \pi_\eta(s) _{s=s_i}$	
25	Update the target networks:	
26	$\psi' \leftarrow \tau\psi + (1 - \tau)\psi'$, $\eta' \leftarrow \tau\eta + (1 - \tau)\eta'$	
27	if the target attractor's basin is reached in s_{t+1} : break	
28	end for	
29	end for	

Apart from the “critic”, DDPG also maintains an “actor” function to map states to a specific action, which is essentially our policy function $\pi(s)$. DDPG uses another neural network parameterized by η as a function approximator of the actor $\pi(s)$, and updates this actor using the gradient of the expected return J with respect to the actor parameters η :

$$\nabla_\eta J \approx \mathbb{E}_{s_t} [\nabla_a Q_\psi(s, a)|_{s=s_t, a=\pi(s_t)} \nabla_\eta \pi_\eta(s)|_{s=s_t}]. \quad (8)$$

In order to enhance the stability of learning, DDPG uses a “replay buffer” and separate “target networks” for calculating the estimated Q-value y_t in Eq. (7). The replay buffer stores transitions $[s_t, a_t, r_t, s_{t+1}]$ from experienced trajectories. The actor $\pi(s)$ and critic $Q(s, a)$ are updated by randomly sampling a minibatch from the buffer, allowing the RL algorithm to benefit from stably learning across uncorrelated transitions. The target networks are copies of actor and critic networks, $\pi'_{\eta'}(s)$ and $Q'_{\psi'}(s, a)$ respectively, that are used for calculating the estimated Q-value y_t . The parameters of these target networks are updated by slowly tracking the learned networks $\pi_\eta(s)$ and $Q_\psi(s, a)$:

$$\begin{aligned} \psi' &\leftarrow \tau\psi + (1 - \tau)\psi', \\ \eta' &\leftarrow \tau\eta + (1 - \tau)\eta', \end{aligned} \quad (9)$$

where $0 < \tau \ll 1$. This soft update constrains the estimated Q-value y_t to change slowly, thus greatly enhancing the stability of learning.

In order to implement DDPG to attractor switching for the energy harvester, several terms needs be defined:

Phase 1: the phase where the system is free of control, i.e., $a(s) = 0$. The system is given a random initial condition at the beginning of Phase 1, waits for dissipation of the transient, and settles down on the initial attractor at the end of Phase 1.

Phase 2: the phase following Phase 1; the system is under control. Phase 2 starts with the system running in the initial attractor, and ends with either reaching the target BoA or exceeding the time limit.

Trajectory: the system's time series trajectories for Phase 2: $[s_t, a_t, r_t, s_{t+1}]$. The data of trajectories are stored in a replay buffer.

Replay Buffer: an implementation of experience replay [48], which randomly selects previously experienced samples to update a control policy. Experience replay stabilizes the RL learning process and reduces the amount of experience required to learn [49].

The entire process for learning a control policy can be summarized as iterative episodes. In each episode, the system runs through Phase 1 and Phase 2 in turn and the control policy is improved using the data of trajectories stored in the replay buffer. Deep deterministic policy gradient (DDPG) is a RL algorithm for updating the control policy.

The detailed learning process using DDPG can be found in Tab. 2. In line 11, the time of Phase 1 is perturbed by adding a random value between 0 and $2\pi/\omega$ (a excitation period). This noise provides diversity of the system's states at the beginning of Phase 2, which enhances generality and helps prevent over-fitting of the control policy network $\pi_\eta(s)$. Both the actor network $\pi_\eta(s)$ and the critic network $Q_\psi(s, a)$ have two hidden layers, each of which has 128 units and an activation function of ReLU [50]. For the actor network, the final output layer is a tanh layer to bound the actions. For the critic network, the input layer consists of only the state s , while the action a is included in the 2nd hidden layer. Adam optimizer [51] was used to learn the neural network parameters with a learning rate of $\tau_\eta = 10^{-4}$ and $\tau_\psi = 10^{-3}$ for the actor and critic respectively. For the update of the target network $\pi'_{\eta'}(s)$ and $Q'_{\psi'}(s, a)$ by Polyak Averaging, we used $\tau = 0.1$. For the system's settling down in Phase 1 we used $T_1 = 2$, and for constraining the time length of control we used $T_2 = 4$. The replay buffer had a size of 10^6 . In each episode, the minibatch of transitions sampled from the replay buffer had a size of $N = 64$.

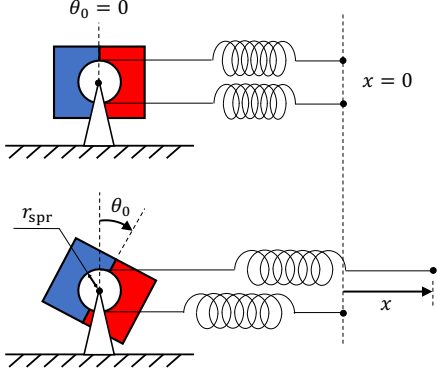


Figure 7: Static equilibria of the offset bias angle θ_0 . It is proportional to the distance between the two springs' ends: $x = 2r_{\text{spr}}\theta_0$.

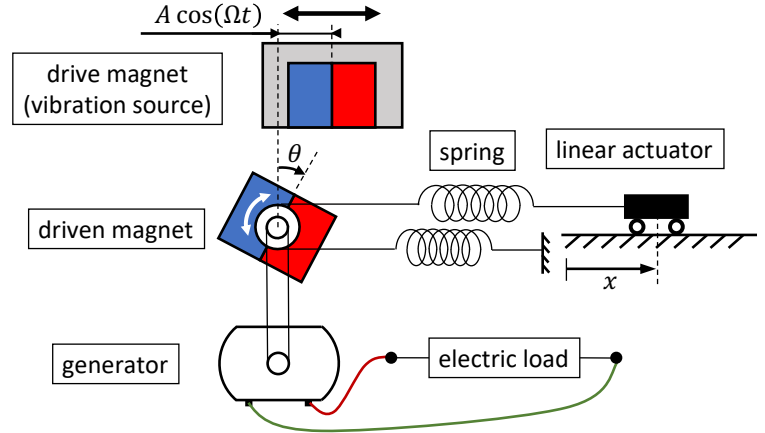


Figure 8: Schematic of control option I, where an external torque on the driven magnet is provided by moving spring position with a linear actuator. The driven magnet transmits rotational motions to a generator to produce electric power. Figure 2 shows the electro-mechanical coupling circuit.

5 Control Option I: Linear Actuator for Moving Spring Position

5.1 Controller Design

Given that the driven magnet exhibits rotational oscillations while the energy harvester is operating, a reasonable control input is an external torque on the driven magnet. Reforming the governing equation Eq. (2) leads to:

$$\begin{aligned} I\ddot{\theta} + c\dot{\theta} + k\theta - \gamma i &= \tau_{\text{mgt}} + \boxed{k\theta_0}, \\ L_g \dot{i} + (R_g + R_{\text{load}}) i + \gamma \dot{\theta} &= 0, \end{aligned} \quad (10)$$

where τ_{mgt} is the magnetic torque, i.e., the right hand side of Eq. (2). The term $k\theta_0$ was intentionally moved to the right hand side, thus can be considered an external torque that is controllable. It forms a torque determined by the offset bias angle θ_0 , which is the static equilibrium angle when the system is only forced by springs. As shown in the Fig. 7, the restoring torque in the system is provided by two linear springs wrapped around the circular plate beneath

the driven magnet. The offset bias angle θ_0 can be altered by moving springs, and has a linear relationship with the distance between ends of springs x :

$$x = 2r_{\text{spr}}\theta_0, \quad (11)$$

where r_{spr} is the radius of the circular plate wrapped by springs. In order to manipulate the distance x , as shown in Fig. 8, the end of one spring is fixed and that of the other spring is attached to a linear actuator. Instead of directly controlling the actuator's position x , its velocity \dot{x} was chosen to be the control input for a more practical control scenario. The system's governing equation becomes:

$$\begin{aligned} I\ddot{\theta} + c\dot{\theta} + k\theta - \gamma i &= \tau_{\text{mgt}} + \frac{k}{2r_{\text{spr}}}x, \\ L_g\dot{i} + (R_g + R_{\text{load}})i + \gamma\dot{\theta} &= 0, \\ \dot{x} &= a, \end{aligned} \quad (12)$$

where a is the control input – the actuator's velocity. Instead of the conventional notation of control input u in control theory, research in reinforcement learning (RL) uses a to represent “actions”. This RL-style notation are used throughout the remainder of this paper.

5.2 Reward Function

As discussed in Sec. 4, the reward function should give 1) a constant positive reward r_{end} when energy harvester reaches a terminal successfully, and 2) penalties r_{cost} (negative rewards) for the cost of actions taken. Evaluating the action cost is tricky, which ideally should be the electric energy consumed on the actuator. However, a realistic actuator is a complex electro-mechanical coupling system, thus making it difficult to calculate its energy consumption especially for a dynamic load. In addition, it's hard to find a generic model to represent all various actuators.

For simplicity, the action cost is defined as the work done by the actuator, that is, “the force pulling on the spring end” times “the distance the spring end moves”. According to Hooke's law, the force can be expressed as $k_{\text{spr}}(x_t - r_{\text{spr}}\theta_t + x_0)$, where k_{spr} is the stiffness of the linear spring, $x_t - r_{\text{spr}}\theta_t$ is the distance stretched, and x_0 is a positive constant to ensure springs are always stretched. The distance the spring end moves can be written as $a_t\Delta t$, where a_t , the control input, is the actuator's instantaneous velocity, and Δt is the time step size for control. The work done by the actuator is therefore $k_{\text{spr}}(x_t - r_{\text{spr}}\theta_t + x_0)a_t\Delta t$. However, this calculation of the action cost is counter-intuitive when $a_t < 0$. The negative a_t leads to a negative work done by the actuator, that is, the actuator is powered by its load. To avoid this paradox, the negative value of a_t was cut off and the reward function becomes:

$$r(s_t, a_t) = -k_{\text{spr}}(x_t - r_{\text{spr}}\theta_t + x_0)\text{Max}(a_t, 0)\Delta t + \begin{cases} r_{\text{end}}, & \text{if } s_{t+1} \text{ reaches target BoA} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

It's worth reiterating that the action cost calculated in this reward function is not the energy consumed on a realistic actuator, but it can still be considered the least energy needed for an action in each time step.

5.3 Results

Figure 9 shows the time series of successful attractor switching using the control policy learned by RL. As observed in Sec. 3, the attractor with a small amplitude of steady-state response (blue lines) is named “LP” (short for low-power) while that with a large amplitude (orange lines) is named “HP” (short for how-power). It's worth noting that each policy only controls a one-way trip of attractor switching. One control policy is needed for transitioning from the LP to HP, and another control policy is needed for the reverse direction. Therefore, two independent sets of parameter η were learned for the actor functions $a(s) = F\pi_{\eta}(s)$. In addition, F is the action bound, which represents the maximum absolute velocity allowed for the actuator. It was set 0.003 m/s for both directions of attractor switching.

Figure 9 includes five phases:

1. controller OFF, operating on the LP attractor (blue lines).
2. controller ON, switching from LP to HP (black lines).
3. controller OFF, waiting for the dissipation of transient process, then operating on the HP attractor (orange lines).
4. controller ON, switching from HP to LP (black lines).
5. controller OFF, waiting for the dissipation of transient process, then operating on the LP attractor (blue lines).

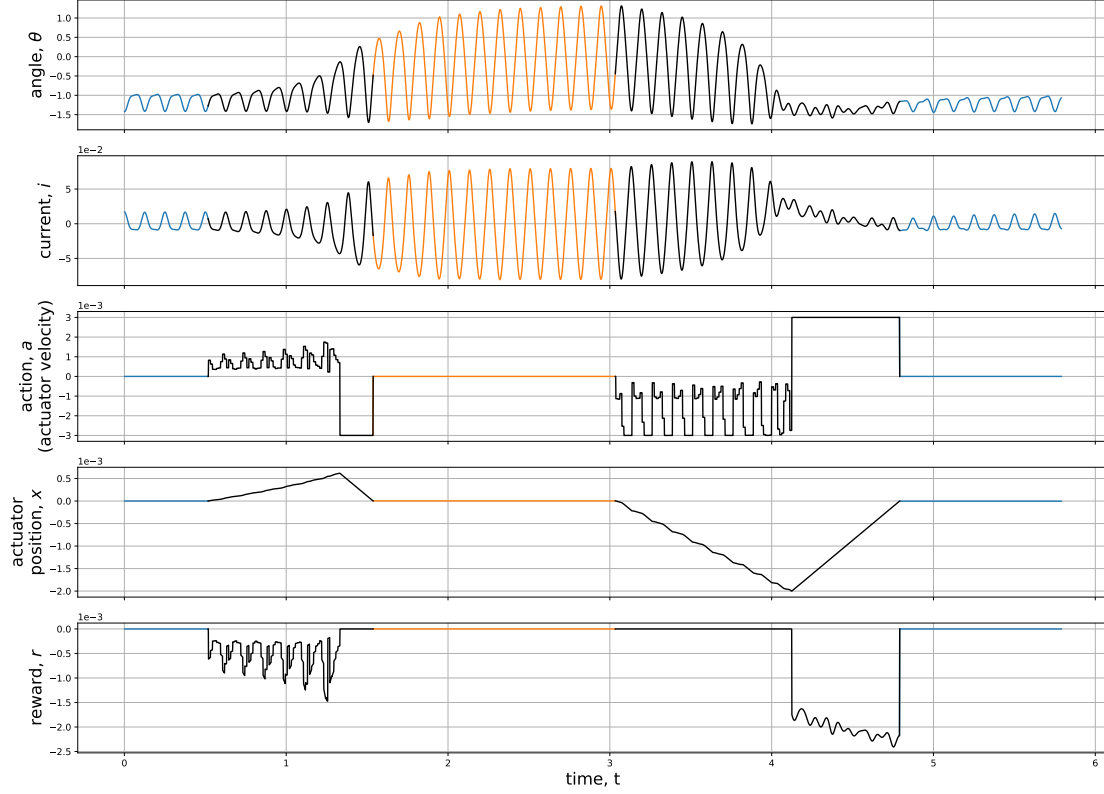


Figure 9: Time series of the attractor switching using the control option I (spring control) + DDPG. The energy harvester governed by Eq. (12) was first switched from LP to HP attractor and then switched back from HP to LP attractor. The blue, orange and black lines represent the system running on the LP attractor, running on the HP attractor, being under control, respectively.

Each time when the controller is ON, one can observe that the action plot consists of two stages: first jagged curve (control policy learned by RL) and then a step function followed (manually-designed control policy). This results from the fact that a successful attractor switching is not only the system’s reaching the target BoA, but also returning the actuator back to its original position, i.e. $x = 0$. If it were otherwise ($x \neq 0$), the attractors themselves will be changed for a non-zero offset bias angle θ_0 (recall that the coexisting attractors observed in Sec. 3 is based on $\theta_0 = 0$ in Tab. 1). Therefore, after the control policy learned by RL drives the energy harvester to the target BoA, additional control policy is needed to reset the actuator. This also explains the trend of actuation position x , where the curve goes away from the origin at first and then returns back.

The reward plot in Fig. 9 demonstrates the work done by the actuator. The area above the curve represents the amount of work, which shows the HP-to-LP switching (1.35×10^{-3} J) needs more energy than the LP-to-HP switching (4.39×10^{-4} J). Furthermore, as defined in Eq. (13), work is done only when the actuator is moving right ($a_t > 0$). This explains that the cost only exists in first half of LP-to-HP switching and second half of HP-to-LP switching.

Although RL proves to realize attractor switching using a linear actuator, the jagged motion of action leads to rapid changes of the actuator’s velocity, which might not work for low-sensitivity actuators. Another control policy with more smooth action was proposed by imitating the trend of the actuator’s position in RL control. For simplicity, the action can only be switched between three states: OFF, ON with maximum positive velocity, and ON with maximum negative velocity. This controller design is similar to a bang-bang controller with binary states (ON & OFF), thus named as “quasi-bang-bang control”. As shown in Fig. 9, the trend of the actuator position is approximately linear. Given that the linear function derivative is a constant, the similar trend of actuator position in RL control can be imitated by implementing constant actuator velocities in quasi-bang-bang control.

Figure 9 shows the time series of successful attractor switching using the quasi-bang-bang control. When the controller is turned on, the actuator starts moving with the maximum velocity. Once the energy harvester reaches the target BoA, the actuator turns to the opposite direction and keeps moving with the maximum velocity until returning to the zero position. LP-to-HP switching requires positive actions first and then negative actions, while HP-to-LP switching has the

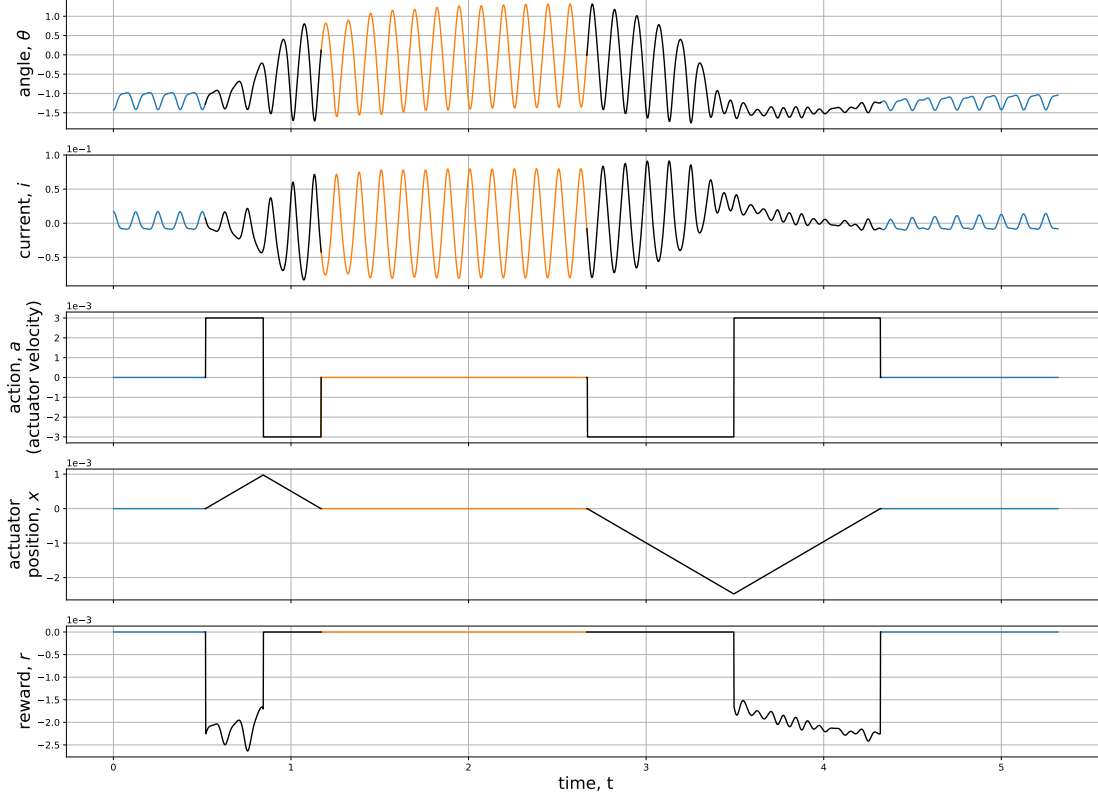


Figure 10: Time series of the attractor switching using the control option I (spring control) + quasi-bang-bang control. The energy harvester governed by Eq. (12) was first switched from LP to HP attractor and then switched back from HP to LP attractor. The blue, orange and black lines represent the system running on the LP attractor, running on the HP attractor, being under control, respectively.

reverse order. This brute-force control method provides a more smooth control than RL, but the lack of optimization over control results in larger energy consumption. By accumulating the rewards in Fig. 9, the HP-to-LP switching needs 1.69×10^{-3} J while the LP-to-HP switching needs 7.62×10^{-4} J.

In summary, this section proposes two control methods based on moving the spring position using a linear actuator. The control policy learned by RL optimizes energy consumption yet shows jagged motions of the actuator, while the quasi-bang-bang control, which was designed by imitating and simplifying the RL control policy, provides a more smooth control input yet lacks in optimization of energy consumption. However, even the RL control gives only an “incomplete” optimization. The requirement of returning the actuator to its original position divides the control process into two stages, and RL only optimizes the energy consumption in the first stage. Therefore, in the next section, another control method is proposed for optimizations over the whole control process.

6 Control Option II: External Voltage on the Motor

6.1 Controller Design

The energy harvester’s attractors are steady-state oscillations determined by three state variables: the driven magnet’s angle, angular velocity and induced current. Apart from exerting an external torque to control the angle and velocity using an actuator in Sec. 5, the induced current could also be controlled by introducing an external power supply in the electro-magnetic circuit. As shown in Fig. 11 and 12, while the energy harvester is operating on an attractor, the generator is driven by the rotating magnet and powering an electric load. When the energy harvester is going to switch to another attractor, the generator is detached from the electric load and connected to a power supply, thus becoming a “motor” to reversely drive the rotating magnet. The circuit will be switched back to connect the electric load once the target attractor is reached.

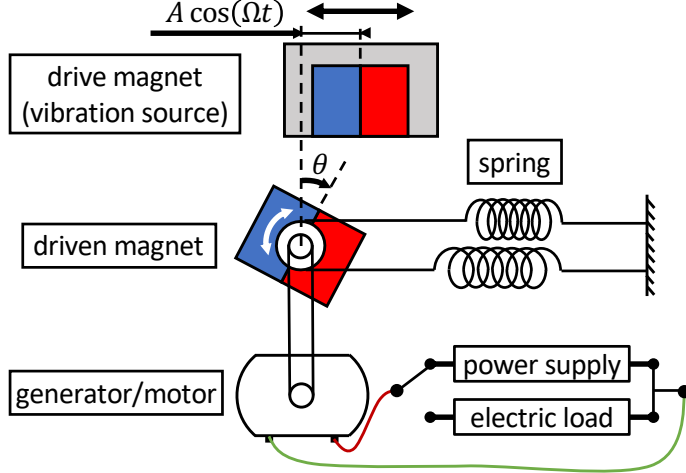


Figure 11: Schematic of control option II (voltage control).

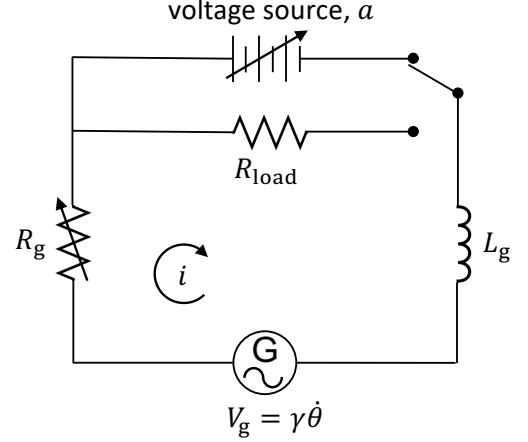


Figure 12: Circuit of control option II (voltage control).

The system's governing equation becomes:

$$\begin{aligned} I\ddot{\theta} + c\dot{\theta} + k\theta - \gamma i &= \tau_{\text{mgt}}, \\ L_g \dot{i} + R_g i + \gamma \dot{\theta} &= \begin{cases} a, & \text{if controller is ON} \\ -R_{\text{load}} i, & \text{if controller is OFF} \end{cases} \end{aligned} \quad (14)$$

where τ_{mgt} is the magnetic torque, i.e., the right hand side of Eq. (2), and a is the control input of voltage. The circuit is switched by using a relay as shown in Fig. 12.

6.2 Reward Function

As discussed in Sec. 4, the reward function should give 1) a constant positive reward r_{end} when energy harvester reaches a terminal successfully, and 2) penalties r_{cost} (negative rewards) for the cost of actions taken. Evaluating the action cost is tricky, which ideally should be the input power of the power supply. However, the complex circuit for a realistic power supply makes it a difficult calculation, especially for a time-varying voltage output. In addition, there is no generic mathematical model to represent various power supplies, which may come from electric power grids, energy storage devices, generators/alternator, etc.

For simplicity, the action cost is defined as the power supply's output power, that is, "the voltage of control input a_t " times "the current flowing through the generator/motor i_t ". The action cost for each time step is therefore $a_t i_t \Delta t$. However, this calculation of the action cost is unrealistic for negative power $a_t i_t < 0$, which means the power supply is being charged by the energy harvester. To avoid this paradox, it's assumed that the power supply has an internal protection mechanism to cut off the negative power. The reward function becomes:

$$r(s_t, a_t) = -\text{Max}(a_t i_t, 0) \Delta t + \begin{cases} r_{\text{end}}, & \text{if } s_{t+1} \text{ reaches target BoA} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

Again, it's worth reiterating that the action cost calculated in this reward function is not the energy consumed on a realistic power supply, but it can still be considered the least energy needed for attractor switching using a voltage control.

6.3 Results

For the attractor switching case that used voltage control, constraints were constructed with different action bounds, i.e., maximum voltage F . Recall that the control term in Eq. (14) can be written as $a(s) = F\pi_\theta(s)$, which is bounded between $-F$ and F . Figure 13 and 14 show the time series of successful attractor switching using the control policy learned by RL, and they have the action bounds of $F = 0.1$ V and $F = 0.2$ V respectively.

Each time-series trajectory has five stages, including LP-to-HP and HP-to-LP switching:

1. controller OFF, operating on the LP attractor (blue lines).

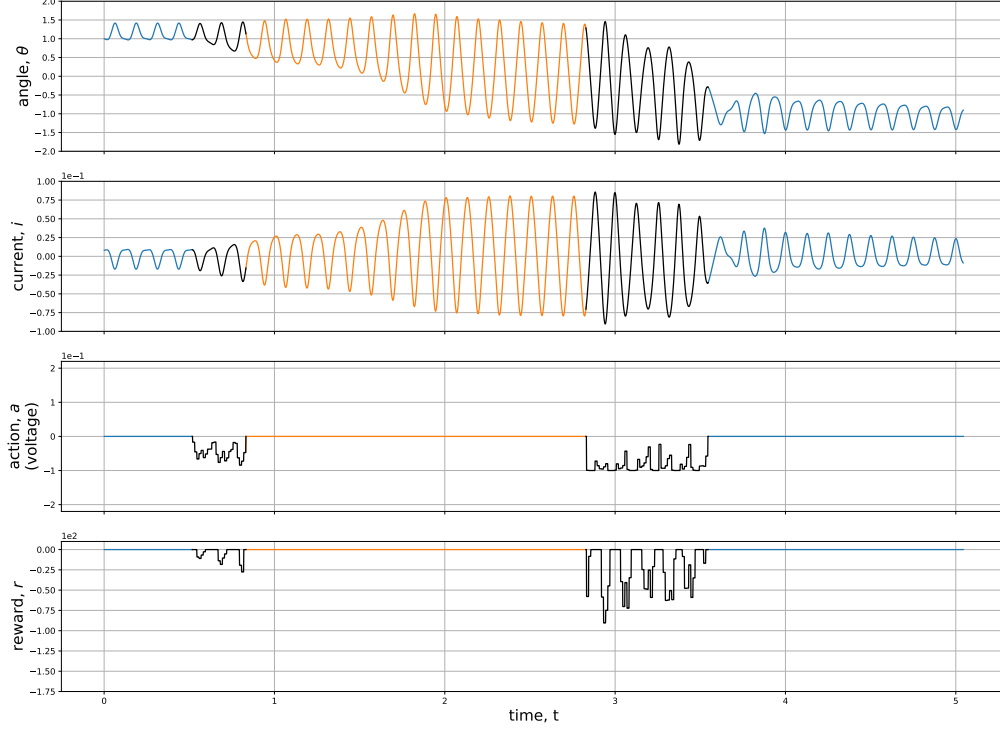


Figure 13: Time series of the attractor switching using the control option II (motor voltage) + voltage bound $F = 0.1$. The energy harvester governed by Eq. (14) was first switched from LP to HP attractor and then switched back from HP to LP attractor. The blue, orange and black lines represent the system running on the LP attractor, running on the HP attractor, being under control, respectively.

2. controller ON, switching from LP to HP (black lines).
3. controller OFF, waiting for the dissipation of transient process, then operating on the HP attractor (orange lines).
4. controller ON, switching from HP to LP (black lines).
5. controller OFF, waiting for the dissipation of transient process, then operating on the LP attractor (blue lines).

Comparing between two action bounds, one can observe that the smaller bound results in a longer time length of control. It can be qualitatively explained by the energy threshold for jumping from one attractor to another. The energy provided by the power supply should be accumulated beyond the energy threshold to push the system away from one attractor. A smaller action bound therefore leads to longer time for the energy accumulation.

Another observation is from the duration of control process. Compared with the short time length for control (region of black lines), the attractor switching spends more time waiting for dissipation of the transient process (especially for the LP-to-HP switching in both Fig. 13 and 14), where the system is automatically approaching the target attractor under no control. This phenomenon shows a smart and efficient strategy the control policy has learned by RL: instead of driving the system precisely to the states of the target attractor, it just drives the system to the attractor's basin, where the system might be far away from the target attractor initially but will reach it without further control effort as time evolves.

6.4 Comparison

The reward plots in Fig. 13 and 14) demonstrate the power supply's output power. The area above the curve represents the total energy consumption for the attractor switching. Although the LP-to-HP switching obviously spends less energy than the HP-to-LP switching, the comparison between two action bounds is obscure. In order to compare the energy consumption quantitatively, their values were calculated by integrating rewards with respective to time. Furthermore, the performance of the energy harvester was evaluated by using the amount of energy harvested in each forcing period, i.e., $E = \int_T i^2 R_{\text{load}}$, where T is the forcing period of vibrational energy source ($T = 2\pi/\Omega$). Given that the response frequencies of both LP and HP attractor is equal to the forcing frequency, this calculation is capable of representing

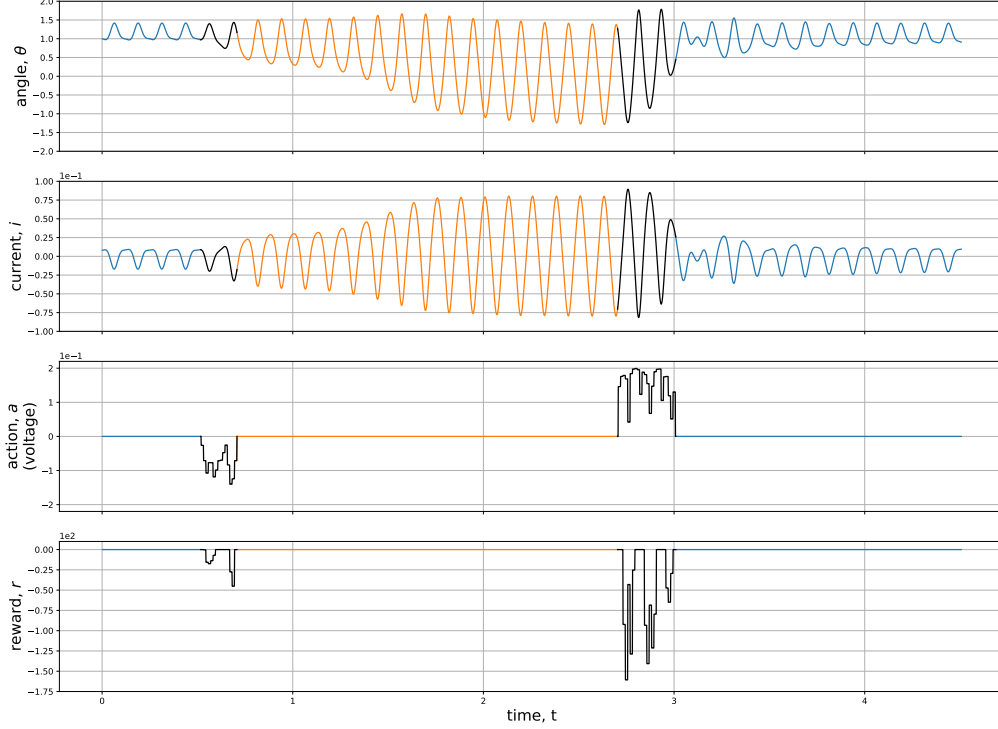


Figure 14: Time series of the attractor switching using the control option II (motor voltage) + voltage bound $F = 0.2$. The energy harvester governed by Eq. (14) was first switched from LP to HP attractor and then switched back from HP to LP attractor. The blue, orange and black lines represent the system running on the LP attractor, running on the HP attractor, being under control, respectively.

the average level of energy harvesting. As a result, in each forcing period, the energy harvested in LP attractor is $E_{LP} = 5.294 \times 10^{-5}$ J while the energy harvested in HP attractor is $E_{HP} = 1.861 \times 10^{-3}$ J.

Figure 15 compares the energy and time consumption for attractor selection under various control scenarios. In Fig. 15(a), the energy consumption is represented by the number of forcing periods to break even, which was calculated by dividing the total energy consumed for attractor switching by the energy harvested in each forcing period (E_{LP} or E_{HP}). There are two numbers above each bar: the top one is the number of forcing periods needed to break even when running on the HP attractor while the bottom one in the parenthesis is for the system running on the LP attractor. In Fig. 15(b), the time consumption is represented by the number of forcing periods to reach target BoA, which was calculated by dividing the time length of control by the forcing period ($2\pi/\Omega$). In order to present a fair comparison, the energy and time consumption for each control scenario is the average of 50 times of attractor switching with random initial conditions.

Several qualitative analysis can be performed from Fig. 15. In general, a large energy consumption corresponds to a longer control duration. For a certain controller, the HP-to-LP switching always consumes more energy than the LP-to-HP switching. The controller II (voltage control) is more energy efficient than the controller I (spring control) for LP-to-HP switching, while it has no obvious advantage in the HP-to-LP switching. For the controller I, the simplified quasi-bang-bang control consumes more energy than the original one learned by reinforcement learning, which agrees with our analysis in Sec. 5. For the controller II with different action bounds (maximum absolute voltage), the smaller bound (i.e. more strict constraint) results in more energy consumption.

7 Conclusion

This paper introduces a nonlinear energy harvester based on a translation-to-rotational magnetic transmission; the harvester's behavior contains coexisting attractors with different levels of electric power output. Two controller designs were investigated to switch the energy harvester's response from one attractor to another. The first controller applies external torques on the driven magnet by altering the position of the spring that provides restoring force. The spring

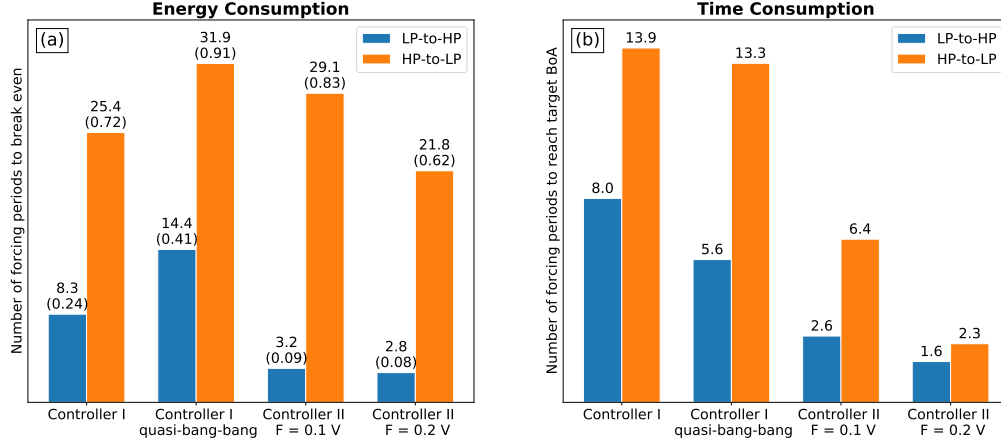


Figure 15: Energy and time consumption for attractor selection in energy harvesting. (a) The energy consumption is represented by the number of forcing periods to break even, which was calculated by dividing the total energy consumed for attractor switching by the energy harvested in each forcing period E_{LP} (or E_{HP}). (b) The time consumption is represented by the number of forcing periods to reach target BoA, which was calculated by dividing the time length of control by the forcing period.

position is controlled by a linear actuator. The second design controls the voltage on the generator in the energy harvester, which temporarily converts the generator into a motor to affect the system’s dynamics.

The control policies based on the two controllers were learned by using deep deterministic policy gradient (DDPG) – a deep reinforcement learning (RL) method. A RL framework for attractor switching was presented, which defines the environment, action, state & observation and reward function. In addition, a neural network classifier was used to rapidly predict the resting attractor based upon the system’s instantaneous state.

Both control methods obtained from RL successfully switch the energy harvester’s response between coexisting attractors. For the control with an actuator, another quasi-bang-bang control was proposed by imitating and simplifying the RL-learned policy. This simplified method is less energy-efficient yet gives more smooth motion of the actuator, which may benefit a realistic actuator. For the voltage control, the RL-learned control policies prove to work under different action bounds (i.e. a power supply’s maximum voltage).

Future work needs to extend our investigations to experimental implementation. Especially for the energy consumption, its measurement on a real linear actuator (controller I) or power supply (controller II) is needed to evaluate the practical performance of an energy harvesting system. Other factors that have not been investigated in this paper (such as observation noise, communication delay, nonlinear damping, etc.) should also be considered in experiments. In order to realize attractor selection experimentally, a feasible solution is to first find a sub-optimal control policy in simulation, as performed in this paper, and then “transfer” the pre-trained policy to the experiments for further optimization. This process is called “transfer learning”, where the heavy-learning workload in experiments is shared with simulations. More studies implementing the attractor selection approach based on real-world experiments and transfer learning are certainly worthy topics for further investigations.

References

- [1] Yi Xi, Hengyu Guo, Yunlong Zi, Xiaogan Li, Jie Wang, Jianan Deng, Shengming Li, Chenguo Hu, Xia Cao, and Zhong Lin Wang. Multifunctional teng for blue energy scavenging and self-powered wind-speed sensor. *Advanced Energy Materials*, 7(12):1602397, 2017.
- [2] Steven P Bastien, Raymond B Sepe, Annette R Grilli, Stephan T Grilli, and Malcolm L Spaulding. Ocean wave energy harvesting buoy for sensors. In *2009 IEEE Energy Conversion Congress and Exposition*, pages 3718–3725. IEEE, 2009.
- [3] J Maxwell Donelan, Qinggua Li, Veronica Naing, JA Hoffer, DJ Weber, and Arthur D Kuo. Biomechanical energy harvesting: generating electricity during walking with minimal user effort. *Science*, 319(5864):807–810, 2008.
- [4] J Paulo and PD Gaspar. Review and future trend of energy harvesting methods for portable medical devices. In *Proceedings of the world congress on engineering*, volume 2, pages 168–196, 2010.

- [5] Gyuhae Park, Tajana Rosing, Michael D Todd, Charles R Farrar, and William Hodgkiss. Energy harvesting for structural health monitoring sensor networks. *Journal of Infrastructure Systems*, 14(1):64–79, 2008.
- [6] TV Galchev, J McCullagh, RL Peterson, and K Najafi. Harvesting traffic-induced vibrations for structural health monitoring of bridges. *Journal of Micromechanics and Microengineering*, 21(10):104005, 2011.
- [7] Lihua Tang, Yaowen Yang, and Chee Kiong Soh. Broadband vibration energy harvesting techniques. In *Advances in energy harvesting methods*, pages 17–61. Springer, 2013.
- [8] Shad Roundy and Yang Zhang. Toward self-tuning adaptive vibration-based microgenerators. In *Smart Structures, Devices, and Systems II*, volume 5649, pages 373–385. International Society for Optics and Photonics, 2005.
- [9] Lei Gu and Carol Livermore. Passive self-tuning energy harvester for extracting energy from rotational motion. *Applied Physics Letters*, 97(8):081904, 2010.
- [10] SE Jo, MS Kim, and YJ Kim. Passive-self-tunable vibrational energy harvester. In *2011 16th International Solid-State Sensors, Actuators and Microsystems Conference*, pages 691–694. IEEE, 2011.
- [11] Timothy Reissman, Eric M Wolff, and Ephraim Garcia. Piezoelectric resonance shifting using tunable nonlinear stiffness. In *Active and Passive Smart Structures and Integrated Systems 2009*, volume 7288, page 72880G. International Society for Optics and Photonics, 2009.
- [12] Dibin Zhu, Stephen Roberts, John Tudor, and Steve Beeby. Closed loop frequency tuning of a vibration-based micro-generator. *University of Exeter*, 2008.
- [13] Christian Peters, Dominic Maurath, Wolfram Schock, Florian Mezger, and Yiannos Manoli. A closed-loop wide-range tunable mechanical resonator for energy harvesting systems. *Journal of Micromechanics and Micro-engineering*, 19(9):094004, 2009.
- [14] Mickaël Lallart, Steven R Anton, and Daniel J Inman. Frequency self-tuning scheme for broadband vibration energy harvesting. *Journal of Intelligent Material Systems and Structures*, 21(9):897–906, 2010.
- [15] Brian P Mann, Samuel C Stanton, and Brian P Bernard. Intentional nonlinearity in energy harvesting systems. In *Proceedings of the 5th International Conference on Applications in Nonlinear Dynamics*, pages 84–95. Springer, 2019.
- [16] BP Mann and ND Sims. Energy harvesting from the nonlinear oscillations of magnetic levitation. *Journal of sound and vibration*, 319(1-2):515–530, 2009.
- [17] BP Mann and BA Owens. Investigations of a nonlinear energy harvester with a bistable potential well. *Journal of Sound and Vibration*, 329(9):1215–1226, 2010.
- [18] Alper Erturk, J Hoffmann, and Daniel J Inman. A piezomagnetoelastic structure for broadband vibration energy harvesting. *Applied Physics Letters*, 94(25):254102, 2009.
- [19] Samuel C Stanton, Clark C McGehee, and Brian P Mann. Reversible hysteresis for broadband magnetopiezoelectric energy harvesting. *Applied Physics Letters*, 95(17):174103, 2009.
- [20] Jamil M Renno, Mohammed F Daqaq, and Daniel J Inman. On the optimal energy harvesting from a vibration source. *Journal of sound and vibration*, 320(1-2):386–405, 2009.
- [21] Angela Triplett and D Dane Quinn. The effect of non-linear piezoelectric coupling on vibration-based energy harvesting. *Journal of Intelligent Material Systems and Structures*, 20(16):1959–1967, 2009.
- [22] Zhen Wu, RL Harne, and KW Wang. Energy harvester synthesis via coupled linear-bistable system with multistable dynamics. *Journal of Applied Mechanics*, 81(6), 2014.
- [23] Samuel C Stanton, Brian P Mann, and Benjamin AM Owens. Melnikov theoretic methods for characterizing the dynamics of the bistable piezoelectric inertial generator in complex spectral environments. *Physica D: Nonlinear Phenomena*, 241(6):711–720, 2012.
- [24] Samuel C Stanton, Alper Erturk, Brian P Mann, and Daniel J Inman. Nonlinear piezoelectricity in electroelastic energy harvesters: modeling and experimental identification. *Journal of Applied Physics*, 108(7):074903, 2010.
- [25] Roszaidi Ramlan, MJ Brennan, BR Mace, and I Kovacic. Potential benefits of a non-linear stiffness in an energy harvesting device. *Nonlinear dynamics*, 59(4):545–558, 2010.
- [26] Benjamin J Bowers and David P Arnold. Spherical, rolling magnet generators for passive energy harvesting from human motion. *Journal of Micromechanics and Microengineering*, 19(9):094008, 2009.
- [27] Xue-She Wang, James D Turner, and Brian P Mann. Constrained attractor selection using deep reinforcement learning. *Journal of Vibration and Control*, page 1077546320930144, 2020.

- [28] Chung Chen and Chi-Cheng Jou. A reinforcement learning control scheme for nonlinear systems with multiple actions. In *Soft Computing in Intelligent Systems and Information Processing. Proceedings of the 1996 Asian Fuzzy Systems Symposium*, pages 43–48, Dec 1996.
- [29] R. S. Sutton, A. G. Barto, and R. J. Williams. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2):19–22, April 1992.
- [30] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.
- [31] Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.
- [32] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.
- [33] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pages 2944–2952, 2015.
- [34] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.
- [35] A Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra. Benchmarking reinforcement learning algorithms on real-world robots. *arXiv preprint arXiv:1809.07731*, 2018.
- [36] Sascha Lange, Martin Riedmiller, and Arne Voigtländer. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2012.
- [37] James D. Turner, Levi H. Manring, and Brian P. Mann. Reinforcement learning for active damping of harmonically excited pendulum with highly nonlinear actuator. In Gaetan Kerschen, M. R. W. Brake, and Ludovic Renson, editors, *Nonlinear Structures and Systems, Volume 1*, pages 119–123, Cham, 2020. Springer International Publishing.
- [38] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [39] Aniruddha Datta, Ashish Choudhary, Michael L Bittner, and Edward R Dougherty. External control in markovian genetic regulatory networks. *Machine learning*, 52(1-2):169–191, 2003.
- [40] Utku Sirin, Faruk Polat, and Reda Alhajj. Employing batch reinforcement learning to control gene regulation without explicitly constructing gene regulatory networks. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [41] Mahdi Imani and Ulisses M Braga-Neto. Control of gene regulatory networks with noisy measurements and uncertain inputs. *IEEE Transactions on Control of Network Systems*, 5(2):760–769, 2017.
- [42] Georgios Papagiannis and Sotiris Moschogiannis. Deep reinforcement learning for control of probabilistic boolean networks. *arXiv preprint arXiv:1909.03331*, 2019.
- [43] Xue-She Wang and Brian P. Mann. Nonlinear dynamics of a non-contact translational-to-rotational magnetic transmission. *Journal of Sound and Vibration*, 459:114861, 2019.
- [44] Xue-She Wang and Brian P Mann. Dynamics of a magnetically excited rotational system. In *Nonlinear Structures and Systems, Volume 1*, pages 99–102. Springer, 2020.
- [45] Brian P Mann, David AW Barton, and Benjamin AM Owens. Uncertainty in performance for linear and nonlinear energy harvesting strategies. *Journal of Intelligent Material Systems and Structures*, 23(13):1451–1460, 2012.
- [46] Xue-She Wang, James D Turner, and Brian P Mann. A model-free sampling method for estimating basins of attraction using hybrid active learning (hal). *arXiv preprint arXiv:2003.10976*, 2020.
- [47] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [48] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.
- [49] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

- [50] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.
- [51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.