Title
The characteristics and application of oscillating robots with spring connection

Author
yong, zhe

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

The characteristics and application of oscillator based robots with spring connection

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Zhe Yong

Committee in charge:

Professor Nicholas G. Gravish, Chair
Professor Nicholas Boechler
Professor Michael T. Tolley

2022

The Thesis of Zhe Yong is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First of all, I would like to sincerely appreciate my advisor Prof. Nicholas Gravish for the opportunity to work in Gravish lab. The research is inspired by his rich experience in dynamic system analysis .Without his knowledge and support, I cannot put all my tentative ideas into practice efficiently and keep refining the core idea.He also inspires me to enrich my research and guide me to overcome difficulties.I learned not only a wealth of knowledge from him, but also how to treat people and do things.

Besides, I want to thank all my lab members, Wei, Rudaong Yang and all the rest members who have been involved in my research project and helped me overcome all kinds of challenges. Wei taught me a lot about the oscillator system and introduced the insight of synchronization to me. His previous work edifies me a lot and his work attitude influenced me to perform better in research. Rundong shared his ideas about the mechanism design with me and gave constructive comments when I met problems. He also assisted me with some experiments, which made the experiments conducted more efficiently.

I also want to give my gratitude to all the friends in the lab. Without their support, I couldn't finish my study so smoothly.

Last but not least, I would also express my thanks to my thesis committee members, Prof. Nicholas Boechle and Prof. Michael Tolley for attending my thesis defense with valuable time and precious advice.

ABSTRACT OF THE THESIS

The characteristics and application of oscillator based robots with spring connection

by

Zhe Yong

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2022

Professor Nicholas G. Gravish, Chair

Many group of organisms that live in proximity are capable of complicated collective movement, which occurs via periodic oscillation of the individuals. One of the fundamental goal for swarm robotics study is to understand how effective and robust the collective behaviors can emerge from simple principles of interaction.In the context, we take advantage of spring connecting to simulate the fluid environment around natural swarm robots, which could give rise to the synchronization.Then we characterize the motion behavior of the spring-oscillator with different control strategy through theoretically and experimentally ways. Critally, The spring could influence the behavior of the oscillator according to certain rules and the coupling robots could achieve desired collective behavior without any network communication.Through

simulation and experiment ,we demonstrate that collective in-phase and antiphase phase behaviors can arise passively through physically adjusting the spring. Finally, we utilize the rules we get to manufacture high stable and adaptable locomotion robots.

Chapter 1

Introduction

The oscillator is a very useful way to generate periodic motion for locomotion robots. Recent studies try to implement the CPG control strategy to control the state [1, 2, 3]. It mainly regulates the phase[4, 1], frequency[5, 6, 7] and amplitude[5] of the oscillators depending on the state of the coupling oscillators.One classical model of CPG control strategy is the Kuramoto model which could adjust the phase of the oscillator by inputting the other oscillators' phase and desired phase difference [8, 9]. Except for getting learning signal from other oscillator, some research shape the limit cycle of the oscillator by using a system that is a universal approximator and approximate the oscillator with this dynamical system [10, 11].And similarly, other research morphs the limit cycle of an existing phase oscillator with phase-based scaling functions to obtain a desired limit cycle behavior [12].There are still some works concentrating on deriving exact and global results on synchronization, antisynchronization, and oscillator death, which help us understand what kind of system could become synchronized.All these methods try to teach the oscillator with the digital signal. Our project proposes a neuromechanical hypothesis for emergent synchronization through soft physical contact so that the system doesn't need to count on receiving other robots' state or learning signal to adjust itself to reach synchronization.

Actually, in the natural world, synchronization often happens in swarm robots [13, 14].Many swimming microorganisms are capable of synchronizing the body or appendage motion by interacting with their neighbors. Recent studies suggest that both long-range hydrodynamic interactions [15, 16] and short-range steric interactions [17, 18] can bring about the stable collective motions.And some researchers investigate the movement of elegants to reveal the intermittent mechanical contact is responsible for synchronization of the undulatory gaits of groups [19]. So for the swarm, instead of communicating with others by digital signal like the Kuramoto model, they prefer to transfer the signal by physical signal such as collision or friction force. Some robots apply this idea to make oscillator-based undulatory robots [20, 21].

In the present work, we hypothesize there are some mechanisms to simulate the fluid environment like the swarm living in. And the mechanisms could be treated as a feedback controller for the oscillator to easily give rise to the synchronization. Critically, this mechanism just involves physical signals caused by the oscillation itself and doesn't require the communication between the systems and thus could greatly simplify coordination of robot groups. In the following we will demonstrate theoretically and experimentally that local oscillators with physical connection will enable in-phase behavior. We further demonstrate that we could control the behavior of the system just by changing the physical characteristics of the part rather than directly altering the controlling parameters inside the controller.Then we investigate how the inner connection between the oscillator and the connecting mechanism and how this connection influences the system.What is more, we apply different types of motor, different oscillator, and different control means to the coupling system and analyze the results derived from different situations. So the conclusion is more general. We lastly apply the results and rules we got to design locomotion robots.

Chapter 2

Position Control

This chapter mainly introduces the previous work including using position control strategy and Hopf-oscillator controller to generate periodic oscillation , applying the phase adaptation function to adjust the phase in order to become synchronized just by receiving the local physical signal such as collision. We also do some improvement from both the controlling function and mechanism. From the controlling function we propose radius adaptation to make the Hopf-oscillator not only adjust the phase but also the radius. From the mechanism we utilize spring instead of collision to deliver the physical signal between robots.After it, verify the improvement by mathematical analysis and relevant simulation. Finally, implement the results we got before to manufacture locomotion robots. The robots will move like a shell with periodic shape change and synchronization behavior.

2.1 Previous Work

The previous work try to use phase adaptation controlling function and servo motor to manufacture the snake like swarm robots. The basic idea of the work is to use a collision-driven adaptive phase Hopf oscillator. It means there is a generic perturbation term p_ϕ influencing the phase of the system:

$$\begin{aligned} \dot{\phi} &= \omega + p_\phi \\ \dot{r} &= (\mu - r^2) r \end{aligned} \tag{2.1}$$

And the perturbation could be expressed as:

$$p_{\phi} = g(\phi_e, \phi) = \gamma \sin(\phi_e - \phi) \tag{2.2}$$

The perturbation term makes sure the robots can receive the physical collision information and then adjust the internal phase to make the whole system become coupling.

We could transform the system from the Phase-Radius coordinate system(PRCS) to Cartesian coordinate system (QCS)

$$\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned} \tag{2.3}$$

x represents the desired position of the rotating joints of the robots and it could be used as the control signal for the motor.

2.2 Improvement

2.2.1 Improvement in control function

However, this controlling function is not perfect. As we can see, the function doesn't make any improvement for the radius part. From Equation 2.1 and Equation 2.3, we know r represents the phase of the oscillator and it could determine the maximum position the robots could reach and μ determines the stable point of the radius. So if robots in the system don't have the same μ. It means the oscillator will not have the same amplitude and then causes the whole system couldn't reach synchronization. Actually, there are some works that try to apply some learning algorithm in the function[7]. The controlling function receives an outside so-called learning signal, and the amplitude r or natural frequency w will gradually converge to the learning signal. But, in our experiment the robots couldn't receive any digital signals except for the physical connection. So inspired by the collision driven adaptation function in the phase evolution part, we apply a similar function to adjust the amplitude of the oscillator according to the amplitude of the adjacent robots:

$$p_r = \gamma_r f(r, r_e) = \gamma_r \sin(r_e - r) \tag{2.4}$$

Likewise , r_e is the estimated radius, and r is the internal radius of the Hopf oscillator.And the γ_r is the gain of the radius adaptation function.

2.2.2 Improvement in Mechanism Design

However, we reconsider the work done before. From previous work[21] , we know in order to make the collisions happen, we firstly need to set a narrow road. This will make collisions happen frequently and then change the phase of each robot in the role. It turns out to work well, but the problem is if robots aren't in a really narrow space, the collisions rarely happen and we couldn't make the whole system become synchronized. It greatly confined the area in which we could apply the strategy. So we want to know if we could directly connect robots together.The connection may have stronger physical signals compared to collisions. The result shows, when they stick together they will immediately become in-phase. But there are still some serious problems. The amplitude of the robots will greatly decrease, the robots couldn't keep constantly oscillating after colliding.

The reason is obvious, just like an inelastic collisionthe sticky surface connects robots rigidly. The connecting force overcomes the torque generated by the motor. However, the sticky surface itself couldn't store energy generated by the motor or drive the robots oscillating on its own. As a result, the robots have a strong tendency to stop.

So in order to solve this problem, we need to find some methods to connect the robots and at the same time the connecting mechanism could store the energy so that the robots could keep oscillating.

One relevant simple way is to use spring. It could undoubtedly store the energy when robots are connected.

What is more, because spring has very good linear property in force generating, and at the same time,it is not hard to get the torque generated by the motor. With this data, we could build the dynamic function of the system.

With the spring connection and servo motor, we could simply model the robots like Fig() shows.

2.3 Dynamic Function, Simply Physical Model and Simulation

2.3.1 Building Dynamic Function

The function could be simply expressed as:

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= T_a + T_s \end{aligned} \tag{2.5}$$

x_1 is the position of the joint $theta$ and x_2 is the velocity of the joint.T_a is the torque generated by the actuator based on the oscillating controller we discussed before, and T_s is the torque caused by the spring.

With this function, we could build the system in the simulation environment to investigate how the improved system perform and do the dynamic analysis in the next step. However, the problem is how we get the torque generated by the motor. The basic idea is even if the servo motor uses position, the motor is actually generating torque to motivate the actuator. For the convenience of building the simulation model, we could suppose the torque generated by the motor is:

$$T_a(\theta, \theta_e) = \lambda(\theta - \theta_e) = \lambda(r\cos\phi - \theta_e) \tag{2.6}$$

θ is the desired position based on the internal phase of the controller and θ_e is the estimated position obtained from the real state of the motor.λ is the positive gain of the actuator.

It is a bit difficult to get the torque generated by the spring,but with the knowledge of forward kinematics of robotics, we could determine the points where the spring is installed in the world frame. For example from Figure 2.1 if we want to install one side of the spring on point A, we could get the position in the world frame based on the homogeneous transformation matrices in 2D space:

$$T_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha & a \\ \sin\alpha & \cos\alpha & b \\ 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & L_1 \\ 0 & 0 & 1 \end{bmatrix}$$
$$A_{body} = \begin{bmatrix} 0 & L_2 & 1 \end{bmatrix}^T \qquad A_{world} = T_1 * T_2 * A_{body} \tag{2.7}$$

And according to the same method, we could get another side of the spring attached to the coupling robot. With the coordinate of two mounting points , we could get the orientation and the length of the spring. In other words, we know the vector of the torque generated by the spring.

This will be the theoretical basis for calculating the spring force later

With all this information, we could finally build the dynamic system in the simulation environment.

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{v_1} \\ \dot{r_1} \\ \dot{\phi_1} \\ \dot{x_2} \\ \dot{v_2} \\ \dot{r_2} \\ \dot{\phi_2} \end{bmatrix} = \begin{bmatrix} v_1 \\ \frac{T_{a1}-T_s}{m} \\ (\mu_1 - r_1^2)r_1 + \gamma_r \sin(r_{e1} - r_1) \\ \omega + \gamma \sin(\arctan\frac{v1}{x1} - \phi_1) \\ v_2 \\ \frac{T_{a2}+T_s}{m} \\ (\mu_2 - r_2^2)r_2 + \gamma_r \sin(r_{e2} - r_2) \\ \omega + \gamma \sin(\arctan\frac{v2}{x2} - \phi_2) \end{bmatrix} \tag{2.8}$$

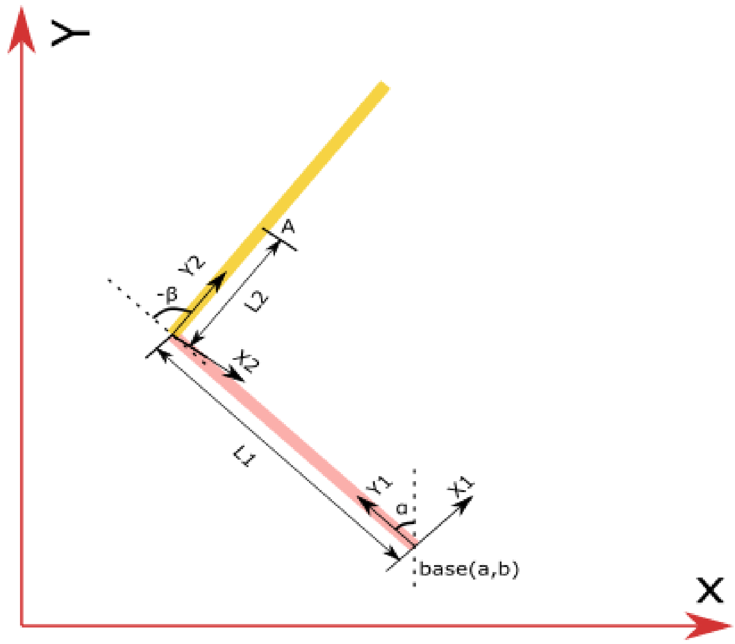

Figure 2.1. The forward kinematic of the one-degree-freedom robots.Based on it, We can know the position of any points in the world coordinate system

2.3.2 Simulation Result

In section 2.31 we know how to express the dynamic function of the system in a mathematical way. With the knowledge of it, we could begin our simulation work. In Matlab, we fix one linkage of the robots to the ground. Just like Figure 2.2shows, a robot has one rotating freedom and one linkage. The linkages are connected together by spring.We treat physical characteristics (e.g. the stiffness and the mounting point) as the variables in the system.

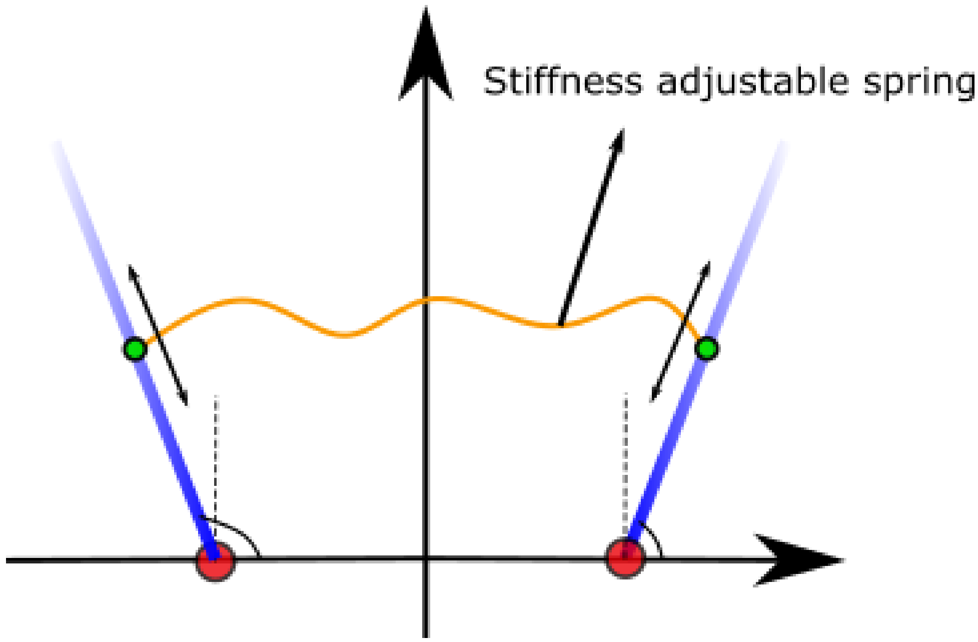

Figure 2.2. The illustration of robots in 2D space. We suppose the spring's stiffness is controllable and there is damping effect between the robots because the configuration of the linkage robots in 2D space. We could also change the mounting point for the spring along the linkage.

The Figure 2.3 shows the simulation results. We are comparing the situation with or without the adaptation function and the spring connection. There are two conclusions:

1. Spring connection could greatly change the state of the controller and force robots become synchronized

2. The Adaptation part of the controlling function could really converge the internal phase and amplitude of the controller to the same value.

3.The radius adaptation function could converge the radius of two oscillating robots so even if the robots have different natural oscillation amplitudes. But there are some problems we couldn't ignore, the radius couldn't stay at a particular value, which means this system will reduce the stability and what is more serious, if the difference of μ_1 and μ_2 is too big, the whole system will become unstable.So the radius adaptation function works on a smaller range comparing to the phase adaptation part.

Then we build a 3D model for the robots and apply the same control strategy in Pybullet, but this time we don't fix one side of the linkage and what is more, we add anisotropic friction to the robots. During the moving process, we try to change the mounting point of the spring. When we change the position where spring is installed, it will directly influence the moving direction. It inspires us if we change the physical characteristics of the system instead of changing the parameters inside the controller, is it possible to change the behavior of the robots. It gives us some new ideas to design locomotion robots.

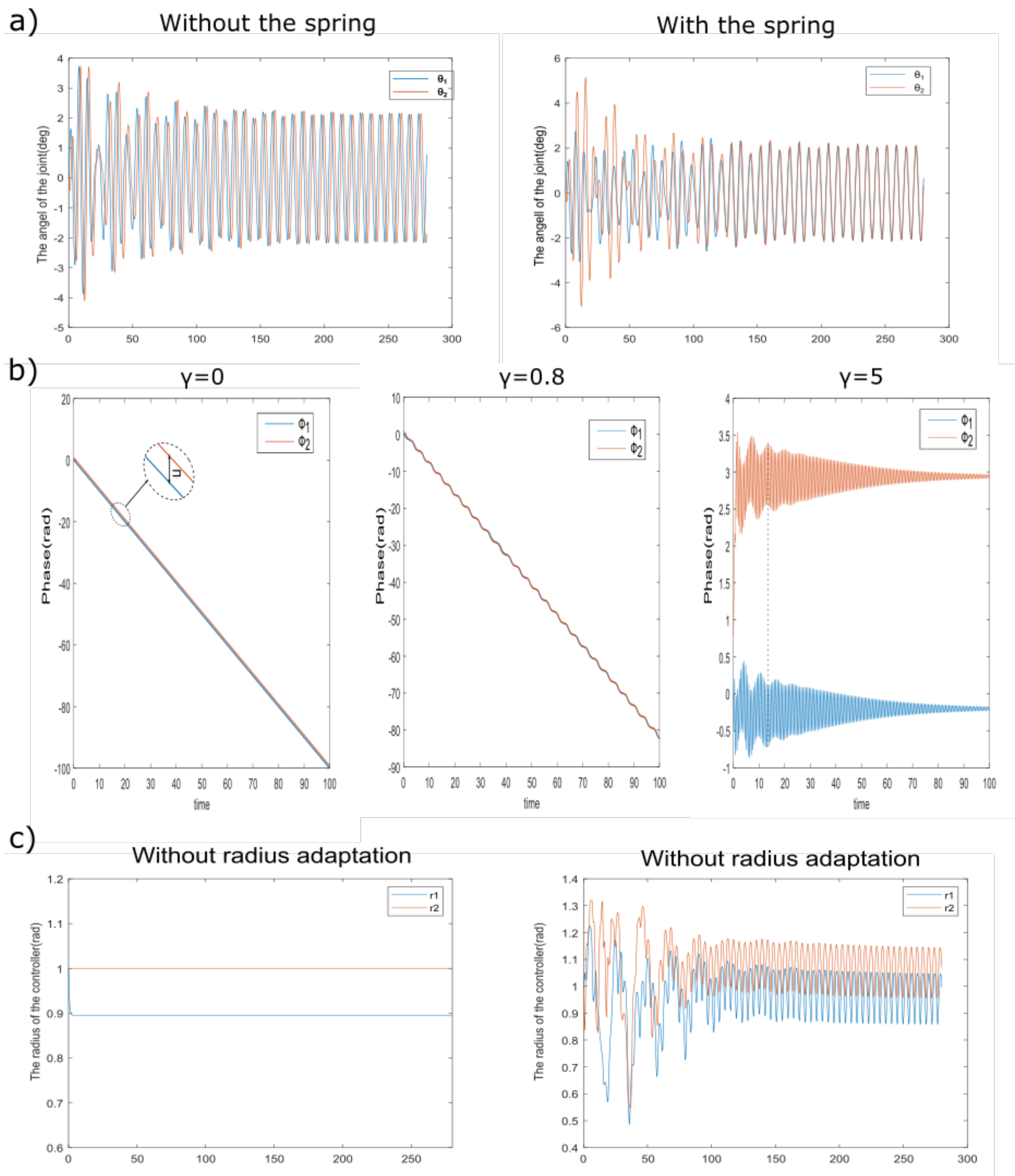

Figure 2.3. The Matlab simulation result of position control. (a) The position evolution of the joint with or without the spring. (b)How the excitatory proprioceptive gain of the phase adaptation part influence

2.4 Analyze the position-control system

From Figure 2.3, we can see sometimes the system will become stable instead of keeping oscillating when we change some control parameters of the system. It is an interesting phenomenon and if we want to know how to utilize the phenomenon we need to find the logic behind it.It means that we need to do the dynamic analysis of the system. First of all, we should realize it is very hard to analyze the system in 2D space,because we need to consider the geometric configuration of the system. It will inevitably involve trigonometric functions which will greatly influence the difficulty of calculation. So we choose to build a similar system in 1D space shown in Figure 2.4 which will greatly reduce the complexity.

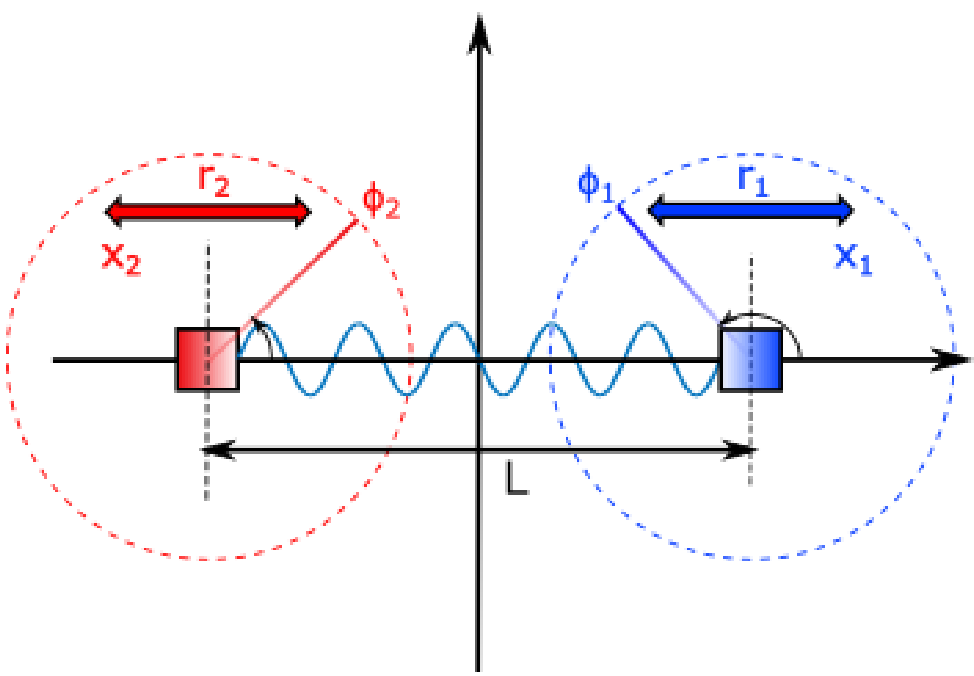

Figure 2.4. The motion of blocks can be equivalently describe by limit cycle diagram,L is the original length of the spring, r_1 and r_2 represent the amplitude of the oscillator at the same time ϕ_1 and ϕ_2 means the phase. Then the x_1 and $x2$ are the corresponding position of the block in Cartesian coordinate system

Before grabbing the exactly expression of the dynammic system, we should make some assumptions and premises first: Even if the center of the limit cycle of the oscillator is not exactly at the origin, but for the convenience of calculation , we still treat the origin as the center. Then because we choose the same μ for two robots, we could ignore the radius part in the calculation.

The dynamic system is just like Equation 2.8.

But now we could get the exact expression for the torque:

$$\begin{aligned} T_{a1} &= \lambda(r_1 \cos\phi_1 - x_1) \\ T_{a2} &= \lambda(r_2 \cos\phi_2 - x_2) \\ T_s &= k(x_1 - x_2 - l) \end{aligned} \tag{2.9}$$

2.4.1 Linearizing the system

Then we know the system will become stable around its equilibrium point which means the system will stop oscillating , so we need to linearize the whole system around the equilibrium. The first step is to get the first-order partial derivatives of all the state variables to form a matrix. This matrix is called Jacobian

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-k-\lambda}{m} & 0 & \lambda\frac{\cos\bar{\phi_1}}{m} & 0 & \frac{k\lambda}{m} & 0 & 0 & 0 \\ 0 & 0 & \mu - 3\bar{r_1}^2 & 0 & 0 & 0 & 0 & 0 \\ -\gamma\frac{\bar{v_1}\cos\alpha_1}{\bar{x_1}^2+\bar{v_1}^2} & \gamma\frac{\bar{x_1}\cos\alpha_1}{\bar{x_1}^2+\bar{v_1}^2\bar{x_1}} & 0 & -\gamma\cos\alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k\lambda}{m} & 0 & 0 & 0 & \frac{-k-\lambda}{m} & \lambda\frac{\cos\bar{\phi_2}}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu - 3\bar{r_2}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma\frac{\bar{v_2}\cos\alpha_2}{\bar{x_2}^2+\bar{v_2}^2} & \gamma\frac{\bar{x_2}\cos\alpha_2}{\bar{x_2}^2+\bar{v_2}^2\bar{x_2}} & 0 & -\gamma\cos\alpha_2 \end{bmatrix} \tag{2.10}$$

The Equation 2.11 represents the equilibrium points,it stands for the value of all the states when the system becomes stable.In other word, it means the value of the all the states to $\dot{x} = 0$:

$$\bar{x} = \begin{bmatrix} \bar{x_1} \\ \bar{v_1} \\ \bar{r_1} \\ \bar{\phi_1} \\ \bar{x_2} \\ \bar{v_2} \\ \bar{r_2} \\ \bar{\phi_2} \end{bmatrix} = \begin{bmatrix} \frac{(k+\lambda)\bar{r_1}\cos\bar{\phi_1}+k\bar{r_2}\cos\phi_2+kl}{2k+\lambda} \\ 0 \\ \mu \\ \arcsin\left(\frac{\omega}{\gamma}\right) \\ \frac{(k+\lambda)\bar{r_2}\cos\bar{\phi_2}+k\bar{r_1}\cos\phi_1-kl}{2k+\lambda} \\ 0 \\ \mu \\ \arcsin\left(\frac{\omega}{\gamma}\right) \end{bmatrix} \tag{2.11}$$

Based on the Jacobian, we could discuss how the parameters influence behavior of the system. For example, because the most important element of the controlling function is the phase adaptation, If we want to know how the gain of the phase adaptation function, we could suppose all the other parameters are the constant. Then we could get the eigenvalue changed by gain.

To simplify the calculation, we choose the value of each controlling variable except for the γ because it scales the adaptation effect of the oscillator's phase,we want to know how it exactly affects the behavior of the system :

$[\omega = 1; \mu = 1; k = 50; l = (2*k+\lambda)/k = 2.2; m = 1; \lambda = 1]$

For this situation, the equilibrium points could be written as:

$$\bar{x} = \begin{bmatrix} \bar{x_1} \\ \bar{v_1} \\ \bar{r_1} \\ \bar{\phi_1} \\ \bar{x_2} \\ \bar{v_2} \\ \bar{r_2} \\ \bar{\phi_2} \end{bmatrix} = \begin{bmatrix} \sqrt{1-\frac{1}{\gamma^2}} - 1 \\ 0 \\ 1 \\ \arcsin\left(-\frac{\omega}{\gamma}\right) \\ \sqrt{1-\frac{1}{\gamma^2}} + 1 \\ 0 \\ \mu \\ \arcsin\left(-\frac{\omega}{\gamma}\right) \end{bmatrix} \tag{2.12}$$

Based on it,we could write the Jacobian matrix.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & \sqrt{1-\frac{1}{\gamma^2}} & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\gamma\sqrt{1-\frac{1}{\gamma^2}}}{\sqrt{1-\frac{1}{\gamma^2}}+1} & 0 & -\gamma\sqrt{1-\frac{1}{\gamma^2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0 & -6 & 0 & \sqrt{1-\frac{1}{\gamma^2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\gamma\sqrt{1-\frac{1}{\gamma^2}}}{\sqrt{1-\frac{1}{\gamma^2}}-1} & 0 & -\gamma\sqrt{1-\frac{1}{\gamma^2}} \end{bmatrix} \tag{2.13}$$

At the end, we could get the eigenvalue of the matrix, which contains eight values basically based on the γ. Six of them are constants or don't have the real part no matter how we change γ. However, there are two eigenvalues that are pretty interesting. The analytical solution for these two eigenvalues is:

$$\lambda_1 = \lambda_2 = -\gamma\sqrt{\frac{\gamma^2 - 1}{\gamma^2}} \tag{2.14}$$

From the expression above, we could get the conclusion that for this system if γ is bigger than one, the value in the square root is more than zero, so we could get the real value. What is more, because γ is always more than zero, the real value, as a result must be less than zero, which means the system has the tendency to converge the equilibrium points we chose before. And, in contrast, if γ is less than one, the value in the square root is less than zero. So there is no negative real part, it will lead the system not to become stable.

Then, we assign the value to the γ. From Figure 2.3 (b), we assign γ to be equal to [0,0.8,5] in order. We could see the evolution of position and phase vary just like we expect from the analytical result. So we could know the gain of the phase adaptation could directly determine if the system became stable or not.

We could treat all the control parameters as the variables in turn,and analyze how these parameters influence the system behavior. It testifies that our system is very flexible. We could change some characteristics of the system to make the system move like we expect.

2.4.2 Experiment based on servo motor

We now focus on the real model experiment based on the position control strategy.

Targeted at the real servo motor based robots, we choose not to fix one side of the robots but make the whole robot float.Naturally, the robots without any constraint could crawl on the ground. It will move like a shell and because all the rigid linkages are connected by the spring, we could treat each robot as a module in the system(Figure 2.5). This could theoretically open up more possibilities for the locomotion system.

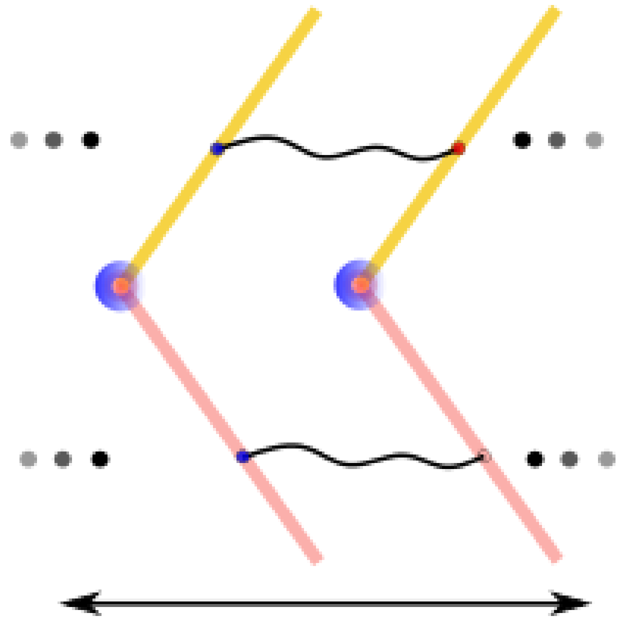

Figure 2.5. The illustration of the locomotion robot.Each robot has two linkages and a one-degree-of-freedom joint motivated by the motor and once the system becomes synchronized and is given anisotropic friction , it will generate the tendency to move in a particular direction. The format of the system could also be modified by attaching or removing the linkage robots.

In the experiment, we set the initial phase difference is π and apply the phase adaptation and radius adaptation to the controlling function. The spring connection could delivery physical information between each robot. The robots are behaving like Figure 2.6 shows:

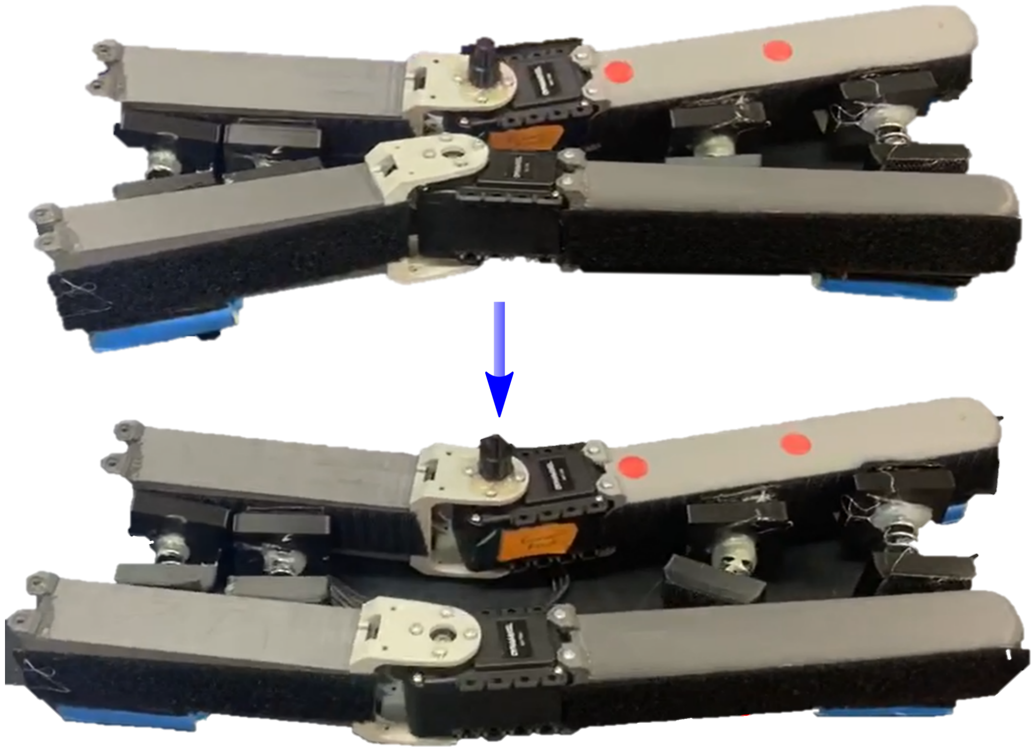

Figure 2.6. The spring forces the system become synchronization, the initial phase difference is random(the upper picture) but eventually the system always becomes synchronized.

The robots firstly try to adjust their phase to become synchronized with the help of the springs and after it become synchronized, the robots will move in one direction very fast due to the anisotropic friction. The moving motion is just like a shell(Figure())

It inspires us if we could make this kind of synchronized oscillation system with anisotropic friction.The robot system will have brilliant moving ability. It could be an important cornerstone for our design of mobile robots.

We also validate how the adaption function help the system reach synchronized by the real model.In the experiment, we connect the robots by spring but control whether the controller has an adaptation equation. We collect some important dates to represent the states of the system during the moving process.It is a comparison with and without adaptation function. We could see the result from Fig(). As Figure 2.7 (a) shows, if the system is without the phase adaptation part, the phase difference will keep oscillating which means the system couldn't become synchronized. But when the system possesses the adaptation part, it can reach synchronization.Figure 2.7 (b) and Figure 2.7 (c) compare the radius evolution in different controlling functions. When the system has the radius adaptation function and even if two robots have different stable values of radius μ, the robots still try to become consistent which means the radius adaptation could force the coupling robots to have the same oscillating amplitude(Figure 2.7 (b)). However, if the controller doesn't have the radius adaptation part, each robot could just reach their stable point based on the value of μ. The two robots, naturallycouldn't have the same amplitude(Figure 2.7 (c)) . The system couldn't become totally synchronized in this situation.

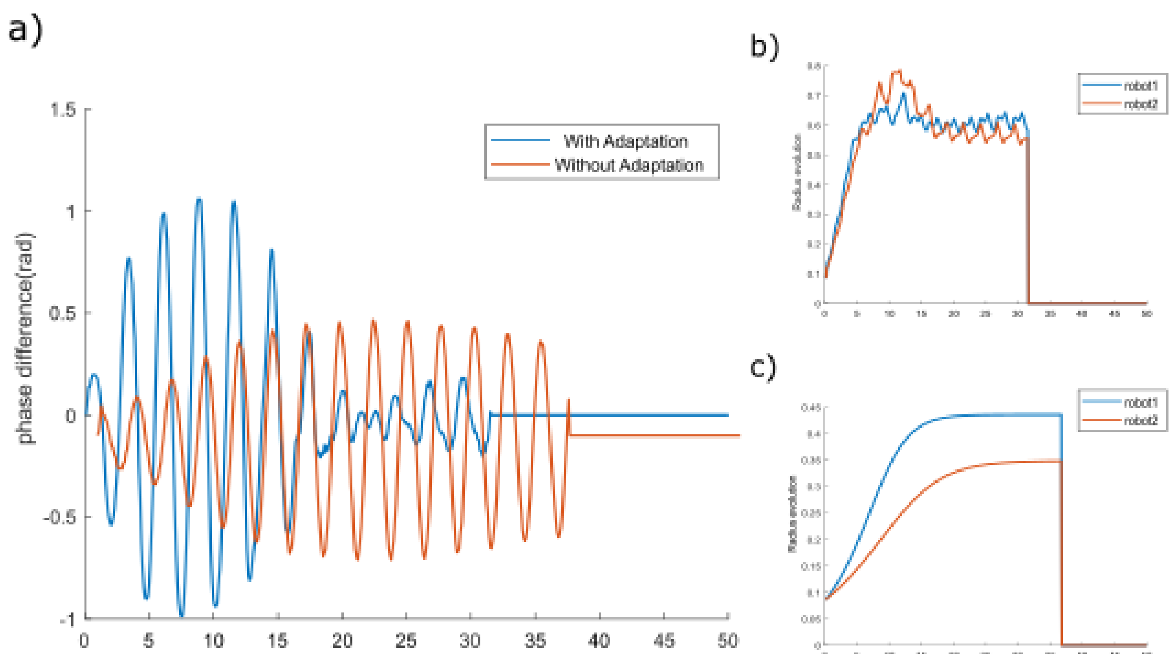

Figure 2.7. The data gotten from the servo motor when the robots are moving. (a) Investigating how the adaptation function influences the phase difference. The robots couldn't become synchronized if we don't apply adaptation functions. (b) The radius evolves when the two robots have different μ but because of the radius adaptation function, the value of the real radius of the oscillator will still converge. (c)Without radius adaptation function, the two robots will converge to their own stable points separately

2.5 Summary

In this Chapter, we use the servo motor and position control strategy to build the robots' system.The controller is based on the Hopf oscillator to generate the periodic motion. The previous work raises the collision-driven model with the phase adaptation function to guide the system becoming synchronized. We propose some new controlling methods to improve the performance of the controlling function. The oscillator now could adjust its amplitude according to the amplitude of the coupling robot. What is more, we think the spring connection could more easily cause the synchronization compared to the collision. Then we get the mathematical expression for the new system and study the system based on the nonlinear system control theory. With the theoretical research, we could comprehend what role does the spring play in the system and quantitatively know the relation between the system behavior and the spring characteristics. In the end, we propose a new shell-like locomotion robot applying the result we got before.If we associate the anisotropic friction to the robots, It could move in one direction after the robots become synchronized.

Chapter 3

Torque Control

In Chapter 2, we use the Hopf oscillator and position control strategy to generate the limit cycle and testify that the spring connection between robots could adjust the phase of the oscillator. Then we discovered some advantages of servo motor and position. In this Chapter, we propose to use the torque motor activated by the DC motor to form the limit cycle and still connect the robots by spring. Build the dynamic function based on the new method and analyze the system from different perspectives and different dimensions to see how the physical characteristics of spring influence the behavior and stability of the system. Besides, we also discussed how to transform the coupling from in-phase to anti-phase (it is basically related to the distance between robots and original length of the spring). A toy model is then utilized to provide physical evidence of this transformation , mathematical validation and dynamical simulation of convergence under the proposed control framework.

3.1 Apply Torque Control

3.1.1 Motor Control

From the previous session, we introduced how to use position control based on Hopf oscillator and spring connection to make the system become synchronized. However, this method still has some serious problems:

1.First, all the dynamic analysis done before is based on the assumption that the torque generated by the servo motor follows the equation: . But in the real physical model, it is not usually like it. The motor always wants to reach its desired input position instantly, so the torque generated by it will not be linear.

2.The servo motor could generate big force to reach the desired position. However, in our project, it is not always a good thing.Because our goal is to control the behavior of the system by changing the physical character besides controlling parameters inside the controller. As a result ,our system, to some extent, should be sensitive and constrained to the elastic spring connection, so it could react to the outside physical signals. So we need to add a phase adaptation part to the controlling function. But sometimes, although we have phase adaptation, it still can't react to the outside physical signal.In this part we set appropriate experiment to make comparison of reaction time to outside collision between DC motor and servo motor. A useful method to test the sensitivity of the system is to see how fast the one degree freedom robot reacts to the collision like Figure 3.1 (a) shows. We mainly observe and record the time interval ΔT. It can be used as a measure of sensitivity.Follwing the idea, we build the platform(Figure 3.1 (b), Figure 3.1 (c)) to compare the time the robots take to bounce back. The Figure 3.2 presents the position evolution when the two kinds of motors are oscillating. They all rebound back because of the fixed wall. But the time interval is totally different. The servo motor will stay at the point of collision for a whilewhile the DC motor barely observable stop time. In order to make the conclusion more general we change the amplitude and frequency of the oscillator. The phenomenon is the same no matter how the controlling parameters is changing the DC motors could be always react to the outside signal faster.

3. Like the second reason shows because we add the phase control part in the position control system , we need to do transformation from PRCS to QCS to get the input controlling signal for the motor and in contrast, we also need to do the transformation from QCS to PRCS to get the feedback signal for our internal controller in each step, which will greatly reduce the accuracy.

4.From Eq, We could see the whole dynamic system for two robots is eighth order, because we need to consider not only the state of the motor but also the phase and radius of the control. It greatly increases the calculating complexity. It is very hard for us to analyze the system in detail.

So according to all these reasons listed above, we need to find another way to design the system

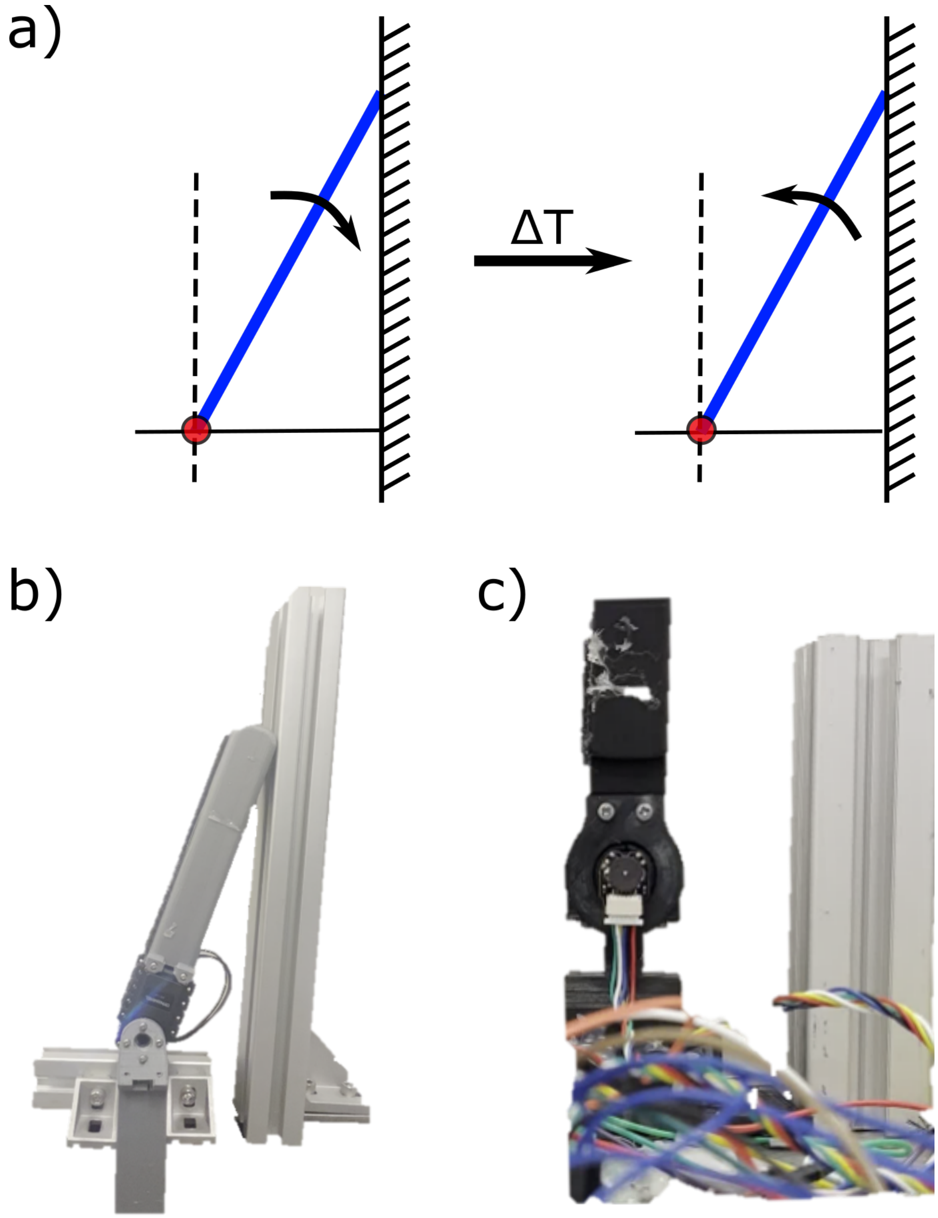

Figure 3.1. The collision test for two type motor. (a)The required time interval for the robots to rebound.(b)Servo motor test.(c)DC motor test.

3.1.2 Oscillator Chosen

One possible way is to use a DC motor to activate the robots. According to others' experiments, we know DC motor could usually be used to generate oscillation motion by directly controlling the torque. What is more, we could easily combine the torque generated by the motor and the force generated by the spring to build the whole dynamic system without making any assumptions. It could effectively increase the accuracy and reliability of the dynamic analysis.

The most important advantage of the DC motor in our project is it is very sensitive to the outside effect, so we don't need to add an extra phase adaptation part in the control part if the outside effect is great enough, it makes us directly control the output torque of the motor, which will greatly reduce the complexity.

Then we use an experiment to test the conclusion we got before. As Figure (b) and Figure (c) show, we build two linkage robots with one joint freedom but are motivated by a DC motor and Servo motor separately . These two robots are all fixed one side to the ground and making them collide with the fixed wall. There are obvious differences between two robots. When a robot is motivated by the Servo motor, it will take some time to rebound back. We could observe there is a manifest interval between the collision and bounding back. However, if we used the DC motor, Interval time dropped significantly.

The position evolution of the oscillation is recorded by Figure 3.2. We could see, if we use DC motor, it will have a very obvious time interval when the robots collide the wall even if we change the amplitude and frequency. The servo always need to take relatively long time to react to the outside physical signal.However, when we turn to the DC motor, the wall could nearly immediately change the moving direction of the robots, but it doesn't have obvious time interval.And we try to reduce the frequency of the system, the reaction time is still be pretty short.

According to the experiment, we could know the DC motor is much more sensitive to the outside physical signal than the Servo motor , so if we want to make the oscillator based robots adjust itself by physical signal, it is better to use the DC motor and torque control.

After deciding the basic method to activate the system, we need to decide which oscillator we should choose to control the motor motor motion.

Finally, we choose the Van der Pol oscillator as the basic part of the system. Van der Pol oscillator evolves in time according to the second order differential equation:

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + k_a x = 0 \tag{3.1}$$

Then for our system we need to get the Two-dimensional form, supposing the transformation $y = \dot{x}$ and making the controller become more controllable[22] ,the oscillator could be written as:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -k_a x + (c - \mu x^2) y \end{aligned} \tag{3.2}$$

In our project x represents the position of the joints ,$y = v$ represent the velocity of the joints.Based on this formation ,we could directly control the torque generated by the DC motor and directly control the phase adaptation. It could skip the transformation from PRCS to QCS which greatly reduces the calculating time and increases the accuracy.

What is more, the Van Der Pol oscillator has a nonlinear damping part and linear spring parts, so it could be directly related to the physical mechanism and at the same time, from our model, robots are also connected by the springs. As a result we could treat our system from a physical perspective like Figure 3.3 shows, it helps us to analyze the system from a more specific way and could understand the system from physical meaning. Based on some material, we know that this kind of oscillator allows for fast phase locking due to their bent isochrones[5].Phase locking means the control system could generate an output signal whose phase is related to the phase of an input signal. And in our project, we need a particular robot in the system that should have some phase relation with the adjacent robots. So the character mentioned above is really helpful in the project.

And for the connection part, we could simply treat it as a spring damping system. We add the damping part in the stimulating environment because due to the geometric configuration in the real world, it will cause dissipation of energy. Adding this part could lead to more realistic simulation results and accurate calculation.

Now, a concise expression of the system can be obtained

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{v_1} \\ \dot{x_2} \\ \dot{v_2} \end{bmatrix} = \begin{bmatrix} v_1 \\ T_{a1} + T_{s1} + T_{d1} \\ v_2 \\ T_{a2} + T_{s2} + T_{d2} \end{bmatrix} \tag{3.3}$$

In the equation, T_a represent the torque generated by the actuator,T_s is the spring force and T_d stand for the damping effect caused by the spring connection.

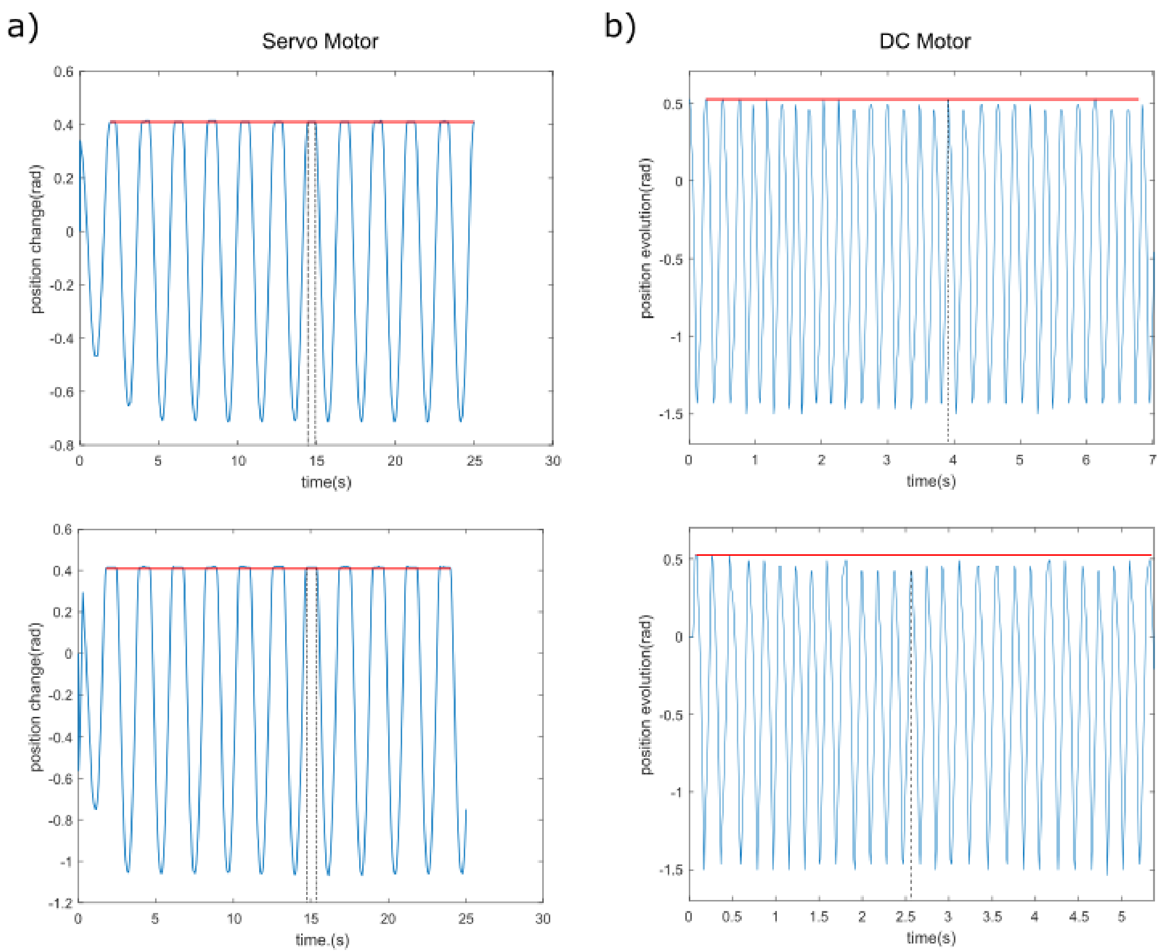

Figure 3.2. Comparing the sensitivity of servos and DC motors to external impacts. (a)The servo motor. (b)The DC motor.

3.2 Possible factors causing robots stop oscillating or keep oscillating in-phase

3.2.1 Build the dynamic system in 1D space

Like we have done before, even if we reduce the order of the system by changing the controlling method, it is still very hard to do mathematical analysis because of the complicated configuration of the robots in 2D space.One way is to simplify the expression of torque caused by spring. We intentionally ignore the complicated configuration of the system in 2D space. Then we build the system

The other way is to try to transform the system into 1D space. For this situation, we could get the exact and concise expression of the dynamic system:

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{v_1} \\ \dot{x_2} \\ \dot{v_2} \end{bmatrix} = \begin{bmatrix} v_1 \\ T_{a1} + T_{s1} + T_{d1} \\ v_2 \\ T_{a2} + T_{s2} + T_{d2} \end{bmatrix} = \begin{bmatrix} v_1 \\ \frac{-k_a x_1 + (c - \mu x_1^2) v_1 - k_s (x_1 - x_2 - l) - k_v (v_1 - v2)}{m} \\ v_2 \\ \frac{-k_a x_2 + (c - \mu x_2^2) v_2 + k_s (x_1 - x_2 - l) - k_v (v_2 - v_1)}{m} \end{bmatrix} \tag{3.4}$$

In this function, k_s represents the stiffness of the spring and k_v represent the strength of damping effect.

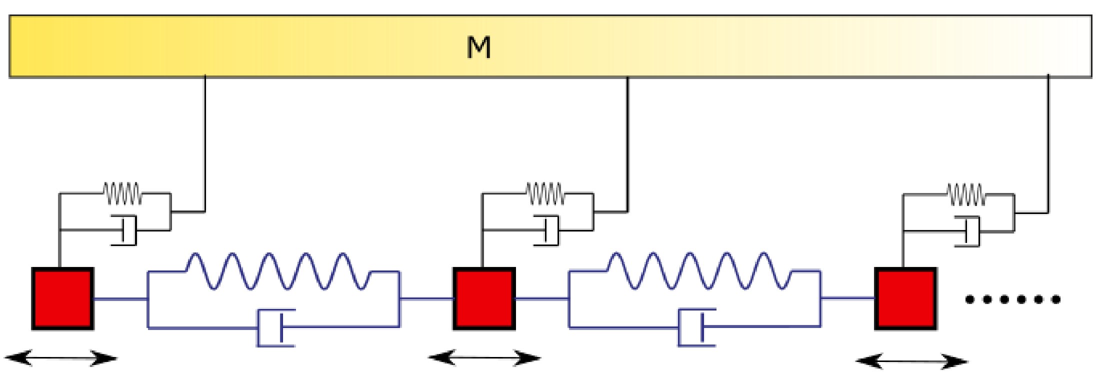

Figure 3.3. The physical meaning of the dynamic system in 1D space, robots is connected by spring-damping system and motivated by spring and nonlinear damping oscillator

3.2.2 Data analysis based on simulation and calculation

We use the similar method to analyze the dynamic characteristics of the system. If we want to linearize the function, we still need to get the Jacobian matrix.

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{v_1} \\ \dot{x_2} \\ \dot{v_2} \end{bmatrix} = \begin{bmatrix} v_1 \\ \frac{-k_a x_1 + (c - \mu x_1^2) v_1 - k_s(x_1 - x_2 - l) - k_v(v_1 - v2)}{m} \\ v_2 \\ \frac{-k_a x_2 + (c - \mu x_2^2) v_2 + k_s(x_1 - x_2 - l) - k_v(v_2 - v_1)}{m} \end{bmatrix} \tag{3.5}$$

And the equilibrium pints are also easy to find.

$$\bar{x} = \begin{bmatrix} \bar{x_1} \\ \bar{v_1} \\ \bar{x_2} \\ \bar{v_2} \end{bmatrix} = \begin{bmatrix} \frac{k_s l}{2k_s + k_a} \\ 0 \\ \frac{-k_s l}{2k_s + k_a} \\ 0 \end{bmatrix} \tag{3.6}$$

In this section, we want to know how the spring connection influences the system behavior. So we choose k_s as the variable and all the other parameters as constants. $k_v = 1; c = 0.5; \mu = 3; k_a = 1; l = 1; m = 1$

Based on all these parameters, the Jacobian matrix could be expressed as :

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_s - 1 & \frac{-(3k_s^2)}{(2k_s+1)^2} - \frac{1}{2} & k_s & 1 \\ 0 & 0 & 0 & 1 \\ k_s & 1 & -k_s - 1 & \frac{-(3k_s^2)}{(2k_s+1)^2} - \frac{1}{2} \end{bmatrix} \tag{3.7}$$

Then the eigenvalue of the matrix is

$$\begin{aligned}
\lambda_1 &= \frac{(4k_s-\sqrt{3})(\sqrt{-(6k_s^2+4k_s+1)(14k_s^2+20k_s+5)}-2k_s^2+1)}{4(2k_s+1)^2}\\
\lambda_2 &= \frac{(4k_s+\sqrt{3})(\sqrt{-(6k_s^2+4k_s+1)(14k_s^2+20k_s+5)}-2k_s^2+1)}{4(2k_s+1)^2}\\
\lambda_3 &= \frac{-(12k_s-\sqrt{-512k_s^5-956k_s^4-848k_s^3-388k_s^2-88_{ks}-7})}{4(2k_s+1)^2}\\
\lambda_4 &= \frac{-(12k_s+\sqrt{-512k_s^5-956k_s^4-848k_s^3-388k_s^2-88_{ks}-7})}{4(2k_s+1)^2}
\end{aligned} \tag{3.8}$$

From the expression, we could see the spring stiffness k_s has great impact on the value of the eigenvalues.Because the spring stiffness $k_s > 0$, the real part of λ_3 and λ_4 is always less than zero. However,for λ_1 and λ_2,because $k_s > 0$, if the k_s is small enough, the real part of these two numbers will be more than zero, and on the contrary, if k_s is big enough , the two values will be less than zero,so all the eigenvalues of the Jacobian matrix are less than zero, which means the system will become stable around the equilibrium point.

We substitute exactly k_s value into the Equation 3.8.From Figure 3.4,we could see some typical evolution of the states variable based on difference k_s value. When $k_s = 0$, the eigenvalue is $[0.25+0.97i 0.25-0.97i -0.75+0.66i -0.75-0.66i]$ for this situation, not all real part of eigenvalues are less than zero, so it will keep oscillating. From physical meaning, without spring, the system will keep oscillating and the robots in the system will not influence each other. when $k_s = 2$, the eigenvalue is $[-0.99+2i -0.99-2i 0.01+1i 0.01-1i]$. For this situation, the positive real part of the eigenvalues is very close to zero. However, the system will not become stable but comparing to $k_s = 0$, the oscillation amplitude decreases, so it means, if we add the spring, it will significantly change the oscillating behaviour. and the stiffness of the spring could directly determine the system become stable or not when $k_s = 3$, the eigenvalue now comes to $[-1.03+2.4i -1.03-2.4i -003+1i -0.03-1i]$, all the real part decreases below zero. The system now really becomes stable, it will stop oscillating after some times. So it means if the spring stiffness is big enough, it will cause the system becoming stable.

From the calculation above, we could say the physical spring connection could influence the oscillator's behavior inside the controller.

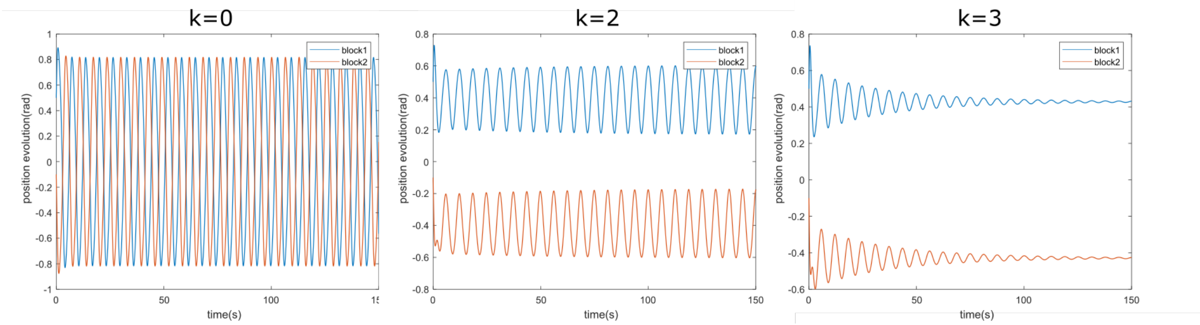

Figure 3.4. The simulation result of Van Der Pol oscillator in 1D space. Comparing the position evolution of blocks under different spring .

3.2.3 Apply Lyapunov Function

In conclusion, by getting the eigenvalue of the Jacobian matrix of the system, we could see how the spring stiffness influences the system behavior and determine the stability .It testifies that the outside physical could really affect the system behavior and what is more this effect is predictable and follow some rules, we could utilize it to construct the robots system.

But if we linearize the system, it, to some extent, just shows the behavior of the system around the equilibrium points and we only know how the spring-damping character influences the system behavior. We don't actually know how the connection between physical characteristics and the control parameters influence the system because for example if we choose more than two parameters as the variable when we want to get the eigenvalue, the expression will be too complex to analyze, and we couldn't get nearly any useful information from it. We want to enlarge the scape and know the connection between all the parameters in the system, we could use the theorem of Lyapunov function to know the stability of the system.

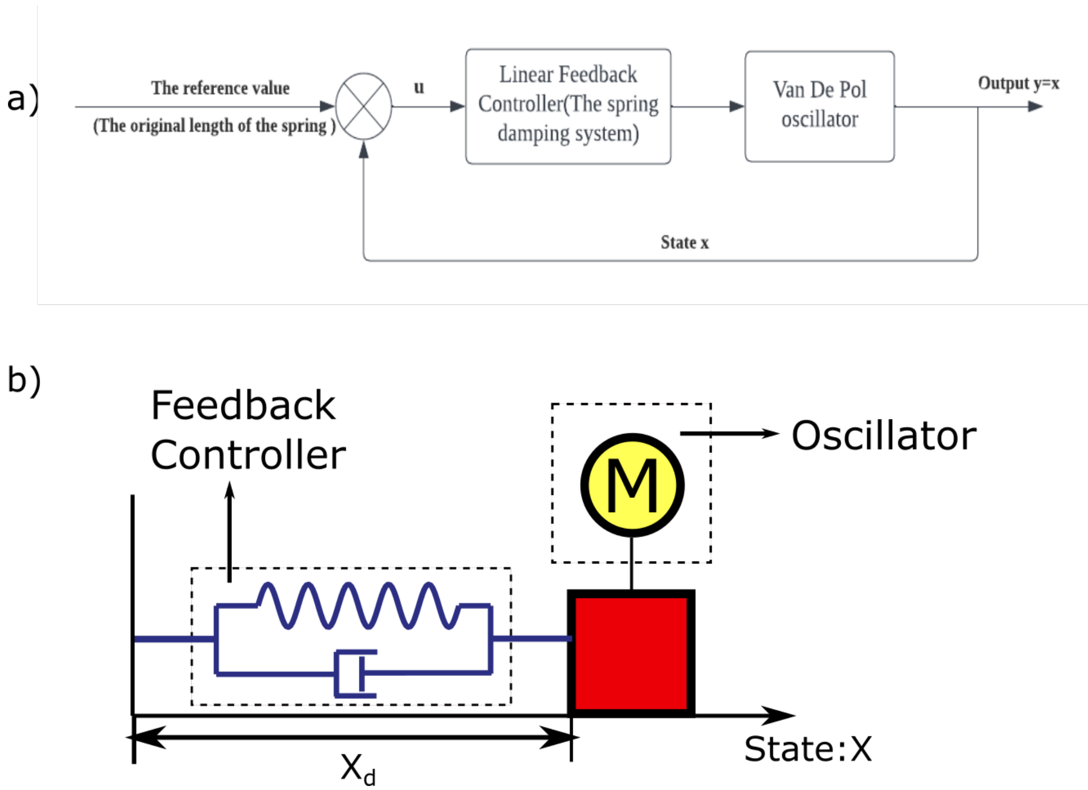

Figure 3.5. The state feedback system based on the model we use. (a)The logic diagram for the state feedback system (b)The connection between our dynamic system and state feedback control system

Before calculating, we know our system could be treated as a linear state feedback control closed-loop system(Figure 3.5 (a)).And like Figure 3.5 (b) shows, the oscillator is the target system and the spring is the linear state feedback controller; the original length of the spring could be treated as the reference meaning the position where the system converges to when it becomes stable . And we also make an important assumption to reduce the complexity of calculation: for this time one side of the spring-damping connecting to the block, but the other side doesn't directly connect the other coupling block but just fix it like Figure 3.5 (b) shows, so one side of the spring will not move, so the state of the system will reduce to $x = [x, \dot{x}]$,which represents the velocity and position of the moving block. The assumption is reasonable, because, in this chapter,we investigate how the all the parameters of the system ,especially the spring damping feedback controller outside the oscillator, influence the stability of the system, we don't really need two coupling robots.

Now we get the Van De pol oscillator with the outside spring-damping control:

$$m\ddot{x}+(\mu x^2-c)v+k_ax=u \tag{3.9}$$

For the convenience of calculation and system design, we rewrite the equation as:

$$\ddot{x}+q_1(\frac{\mu}{c}x^2-1)v+q_2=q_3u \tag{3.10}$$

where, $q_1=\frac{c}{m}$,$q_2=\frac{k_a}{m}$,$q_3=\frac{1}{m}$

The we suppose x_d is the original length of the spring,and $y=x$ is the output, From the system control behave, the x_d represents the setpoint. Then we should to design a linear controller , in other word, choose the suitable physical characteristics of the spring damping system to regulate y to x.Letting $e=y-x$ and we could get the equation $\dot{x}=\dot{e}$ and $\ddot{x}=\ddot{e}$ Then we have the state feedback controller:

$$q_3u=q_2x_d+q_1\frac{\mu}{c}x_d^2\dot{e}-k_pe-k_d\dot{e} \tag{3.11}$$

where $k_p=k\lambda+1-q_2$ and $k_d=k+\lambda+q_1$,and because it is negative feedback control, we need $k>q_1x_d^2$ and $\lambda>0$, k and λ are all control parameters of the feedback controller.

We know for the Van Der Pol oscillator due to the feedback controller law, the close-loop signal $y(t)-x_d$ and $x(\dot{t})$ should converge to zero when t $\rightarrow\infty$.As a result, the equilibrium points is $\bar{e}=[0,0]$. It means with the spring-damping system, the system stop oscillating at the neutral position.

Notice that

$$x^2\dot{x}=(e+x_d)^2\dot{e}=x_d^2\dot{e}+2x_de\dot{e}+e^2\dot{e} \tag{3.12}$$

For the convenience, suppose $\bar{q_1}=\frac{\mu}{c}q_1$ Substituting Equation 3.12 and Equation 3.11 into

Equation 3.10, we obtain:

$$\ddot{e} - q_1\dot{e} + q_2 e + 2\bar{q}_1 x_d e\dot{e} + \bar{q}_1 e^2\dot{e} + k_p e + k_d\dot{e} = 0 \tag{3.13}$$

Suppose $e_1 = e$ and $e_2 = \dot{e}$, the Equation 3.13 becomes:

$$\dot{e} = \begin{bmatrix} \dot{e_1} \\ \dot{e_2} \end{bmatrix} = \begin{bmatrix} e_2 \\ -ae_1 - be_2 - 2\bar{q}_1 x_d e_1 e_2 - \bar{q}_1 e_1^2 e_2 \end{bmatrix} \tag{3.14}$$

where $a = q_2 + k_p$ and $b = k_d - q_1$

Consider the Lyapunov function candidate

$$V = \frac{1}{2}e_1^2 + \frac{1}{2}(e_2 + \lambda e_1)^2 + \frac{\lambda\bar{q}_1}{4}e_1^4 \tag{3.15}$$

Like we discussed before, $e = [0,0]$ is the equilibrium point. The domain area $D \subset R^2$

$$\begin{aligned} V(0) &= 0, \\ V(x) &> 0 \qquad in\ D - \{0\} \end{aligned} \tag{3.16}$$

Then the derivative of along function() is:

$$
\begin{aligned}
\dot{V} =& e_1\dot{e_1} + (e_2+\lambda e_1)(\dot{e_2}+\lambda\dot{e_1}) + \lambda\bar{q_1}e_1^3 e_2 \\
=& -\lambda e_1^2 + (e_2+\lambda e_1)(-ae_1 - be_2 + e_1 + \lambda e_2 - 2\bar{q_1}x_d e_1 e_2 - \bar{q_1}e_1^2 e_2 \\
& + \lambda e_2) + \lambda\bar{q_1}e1^3 e_2 \\
=& -\lambda e_1^2 + (e_2+\lambda e_1)(-ae_1 - be_2 + e_1 + \lambda e_2) \\
& - 2\bar{q_1}x_d e_1 e_2(e_2+\lambda e_1) - \bar{q_1}e_1^2 e_2^2 \\
=& -\lambda e_1^2 - k(e_2+\lambda e_1)^2 - 2\bar{q_1}x_d e_1 e_2(e_2+\lambda e_1) \\
& - \bar{q_1}e_1^2 e_2^2 \\
=& -\lambda e_1^2 - k[(e_2+\lambda e_1) + \frac{\bar{q_1}}{k}e_1 e_2 x_d]^2 - (\bar{q_1} - \frac{\bar{q_1}^2 x_d^2}{k})e_1^2 e_2^2
\end{aligned}
\tag{3.17}
$$

We know when we select

$$
\begin{aligned}
\bar{q_1} &> \frac{\bar{q_1}^2 x_d^2}{k} \\
\Rightarrow k &> \frac{\mu}{c} * \frac{c}{m} x_d^2 = \frac{\mu}{m} x_d^2
\end{aligned}
$$

The $\dot{V}$ is negative in $D - \{0\}$

What is more, we use the similar method, but this time, the Van Der Pol based feedback control system is written as follows:

$$\ddot{x} + q_1^\star(x^2 - \frac{c}{\mu})v + q_2^\star = q_3^\star u \tag{3.18}$$

where $q_2^\star = q_2$ and $q_3^\star = q_3$ are the same, but $q_1^\star = \frac{\mu}{m}$

and this time the linear feedback controller is:

$$q_3^\star u = q_2^\star x_d + q_1^\star x_d^2 \dot{e} - \bar{k_p} e - \bar{k_d} \dot{e} \tag{3.19}$$

where $\bar{k_p} = k_p$ but $\bar{k_d} = k + \lambda + \bar{q_1}\frac{c}{\mu}$

Then we use the similar method and the same Lyapunov function,we get the same result:

$$q_1^\star > \frac{q_1^{\star 2} x_d^2}{k}$$
$$\Rightarrow k > q^\star x_d^2 = \frac{\mu}{m} x_d^2$$

So the extract number doesn't influence the result

Then suppose the gain of the feedback controller:

$$\begin{aligned} k_1 &= k_p = k\lambda + 1 - \frac{k_a}{m} \\ k_2 &= k_d - q_1 \frac{\mu}{c} x_d^2 = k + \lambda + \frac{c}{m} - \frac{c}{m}\frac{\mu}{c} x_d^2 \end{aligned} \tag{3.20}$$

It shows if we want to keep the system stop oscillating,we should control the spring-damping to satisfy the inequality

$$\begin{aligned} k_1 &> \frac{\mu}{m} x_d^2 + 1 - \frac{k_a}{m} \\ k_2 &> \lambda + \frac{c}{m} \end{aligned} \tag{3.21}$$

Now,we know there is a direct connection between the nonlinear damping part the original length of the spring and the spring stiffness.

It means when the gain of the controller is big enough, it will make the system become stable,and in the inequality, and if the original length of the spring is long, if we want to the system become stable, we also need to increase the spring stiffness.

We have solved the most complicated part of the problem.We know the inner connection between the nonlinear damping and spring stiffness. But, we still want to know how the linear damping part influences the stability of the system.This time, we ignore the nonlinear damping gain. Now the system has come to a typical linear system, it is much easier to get the result.

The system be written like it:

$$m\ddot{x} - cv + k_a x = u \tag{3.22}$$

We still need to simplify the equation

$$\ddot{x} - p_1 v + p_2 x = p_3 u \tag{3.23}$$

where $p_1 = \frac{c}{m}$, $p_2 = \frac{k_a}{m}$, $p_3 = \frac{1}{m}$

Correspondingly, the state feedback controller could be considered as:

$$p_3 u = p_2 x_d - k_1 e - k_2 \dot{e} \tag{3.24}$$

k_1 and k_2 are the design parameters

We could get the close loop system based on the close-loop system:

$$\begin{aligned} \dot{e} = \begin{bmatrix} \dot{e_1} \\ \dot{e_2} \end{bmatrix} &= \begin{bmatrix} e_2 \\ p_1 e_2 - p_2 e_1 - k_1 e_1 - k_2 e_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -p_2 - k_1 & p_1 - k_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \end{aligned} \tag{3.25}$$

e_1 and e_2 are the same previous definition.

Suppose:

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ -p_2 - k_1 & p_1 - k_2 \end{bmatrix} \tag{3.26}$$

It is a typical linear system, and if $\bar{A}$ is Hurwitz , the system will become stable.It means each eigenvalue of the matrix $\bar{A}$ should has strictly negative real part.The expression of the eigenvalue is:

$$\lambda_1 = \frac{p_1}{2} - \frac{k_2}{2} - \frac{\sqrt{(k_2-p_1)^2-4k_1-4p_2}}{2}$$
$$\lambda_2 = \frac{p_1}{2} - \frac{k_2}{2} + \frac{\sqrt{(k_2-p_1)^2-4k_1-4p_2}}{2} \tag{3.27}$$

We come to analyze the equation. Both eigenvalues contains the part $-\frac{k_2}{2}$, which means the feedback control parameter k_2 could directly determine the real part of the eigenvalue. And with relative big k_2, the value will more easily have negative real part. And then consider the value in the square root, we find it contains the component $(k_2 - p_1)^2$ and we also know $k_1 > 0$ and $p_2 > 0$ and if $-\frac{k_2}{2} - \frac{p_1}{2} > 0$, we could make sure the real part of the two eigenvalues must be less than zero.

So based on it, if the system want to be stable, it must satisfy the inequality:

$$\frac{p_1}{2} - \frac{k_2}{2} < 0$$
$$\Rightarrow k_2 > \frac{c}{m} \tag{3.28}$$

Now we know the feedback control parameters is also has directly connection with the linear damping part of the controller c.

What is more, if k_1 is big enough, the value in the square root will become negative, which means the maximum real value of two eigenvalues will decrease. This will cause the system more readily become stable.

In this chapter, we simplify the coupling system.It reduces the orders of the state so that we could more easily do the calculation. It gives a robust proof that if we want the system to remain stable, the conditions that should be satisfied between the various parameters of the system.

3.2.4 Further the System in 2D space

Now, we discussed how the spring characteristics and control parameters of the oscillator influence the oscillating system in 1D space. The information obtained from the simulation and calculations shows it is possible to change the behavior of the oscillating system regularly and oscillating robots could adjust its state by physical information from adjacent robots, which will make the connecting robots system have controllable collective behavior. However, if we want to apply this strategy to the rotating joint inside a robot, we should at least testify the property of the system in 2D space(Figure 2.2).

So like we discussed in chapter two, we apply the forward kinematic to get the point where we install the spring in the world frame. And then we know the magnitude and direction of the force. The torque from the motor is also easy to know. In the end, we know all the torque applied to the joint. Based on it, we could do dynamic analysis and simulation to investigate the system.

In general, the system could be written as:

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{v_1} \\ \dot{x_2} \\ \dot{v_2} \end{bmatrix} = \begin{bmatrix} v_1 \\ \frac{k_1(-k_a x_1 + (c-\mu x_1^2)v_1) - k_2 * T_1 + k_3 * T_{damping1}}{j} \\ v_2 \\ \frac{k_2(-k_a x_2 + (c-\mu x_2^2))v_2 + k_2 * T_1 + k_3 * T_{damping2}}{j} \end{bmatrix} \tag{3.29}$$

The expression of spring torque is really complicated, because it involves the topology of the linkage configuration.We need to use the theory of forward kinematic to get the present length of the spring and further get the torque generated by the spring.

$$\begin{aligned} T_1 &= \overrightarrow{l_1} \times \overrightarrow{(-F_s)} \\ &= (l_{fix} * \cos x_1) * (-F_s) * \cos(\arctan \frac{(l_{fix} * \sin x_1 + oy_1) - (l_{fix} * \sin x_2 + oy_2)}{(l_{fix} * \cos x_1 + ox_1) - (l_{fix} * \cos x_2 + ox_2)}) \\ &- (l_{fix} * \sin x_1) * (-F_s) * \sin(\arctan \frac{(l_{fix} * \sin x_1 + oy_1) - (l_{fix} * \sin x_2 + oy_2)}{(l_{fix} * \cos x_1 + ox_1) - (l_{fix} * \cos x_2 + ox_2)}) \end{aligned} \tag{3.30}$$

$$
\begin{aligned}
T_2 &= \overrightarrow{l_2} \times (\overrightarrow{F_s}) \\
&= (l_{fix} * \cos x_2) * (F_s) * \cos(\arctan \frac{(l_{fix} * \sin x_2 + oy_2) - (l_{fix} * \sin x_2 + oy_2)}{(l_{fix} * \cos x_1 + ox_1) - (l_{fix} * \cos x_2 + ox_2)}) \\
&- (l_{fix} * \sin x_2) * (F_s) * \sin(\arctan \frac{(l_{fix} * \sin x_1 + oy_1) - (l_{fix} * \sin x_2 + oy_2)}{(l_{fix} * \cos x_1 + ox_1) - (l_{fix} * \cos x_2 + ox_2)})
\end{aligned} \tag{3.31}
$$

And we suppose there is a damping effect between the robots because the spring connection could't store all the energy. Because unlike compressed and stretched in 1D space, the spring is twisted and it will cause energy waste.

$$
\begin{aligned}
T_{damping1} &= \overrightarrow{l_1} \times [(-k_v) * \overrightarrow{v_1} \times \overrightarrow{l_1}] \\
&= (-k_v) * [(l_{fix} \cos x_1)(v_1 * l_{fix} * \cos x_1) - (l_{fix} * \sin x_1)(v_1 * l_{fix} * \sin x_1)]
\end{aligned} \tag{3.32}
$$

$$
\begin{aligned}
T_{damping2} &= \overrightarrow{l_2} \times [(-k_v) * \overrightarrow{v_2} \times \overrightarrow{l_2}] \\
&= (-k_v) * [(l_{fix} \cos x_2)(v_2 * l_{fix} * \cos x_2) - (l_{fix} * \sin x_2)(v_2 * l_{fix} * \sin x_2)]
\end{aligned} \tag{3.33}
$$

In the equation, T_1 and T_2 represent the torque caused by the spring and $T_{damping1}$ and $T_{damping2}$ are the possible resistant torque caused by damping effect.

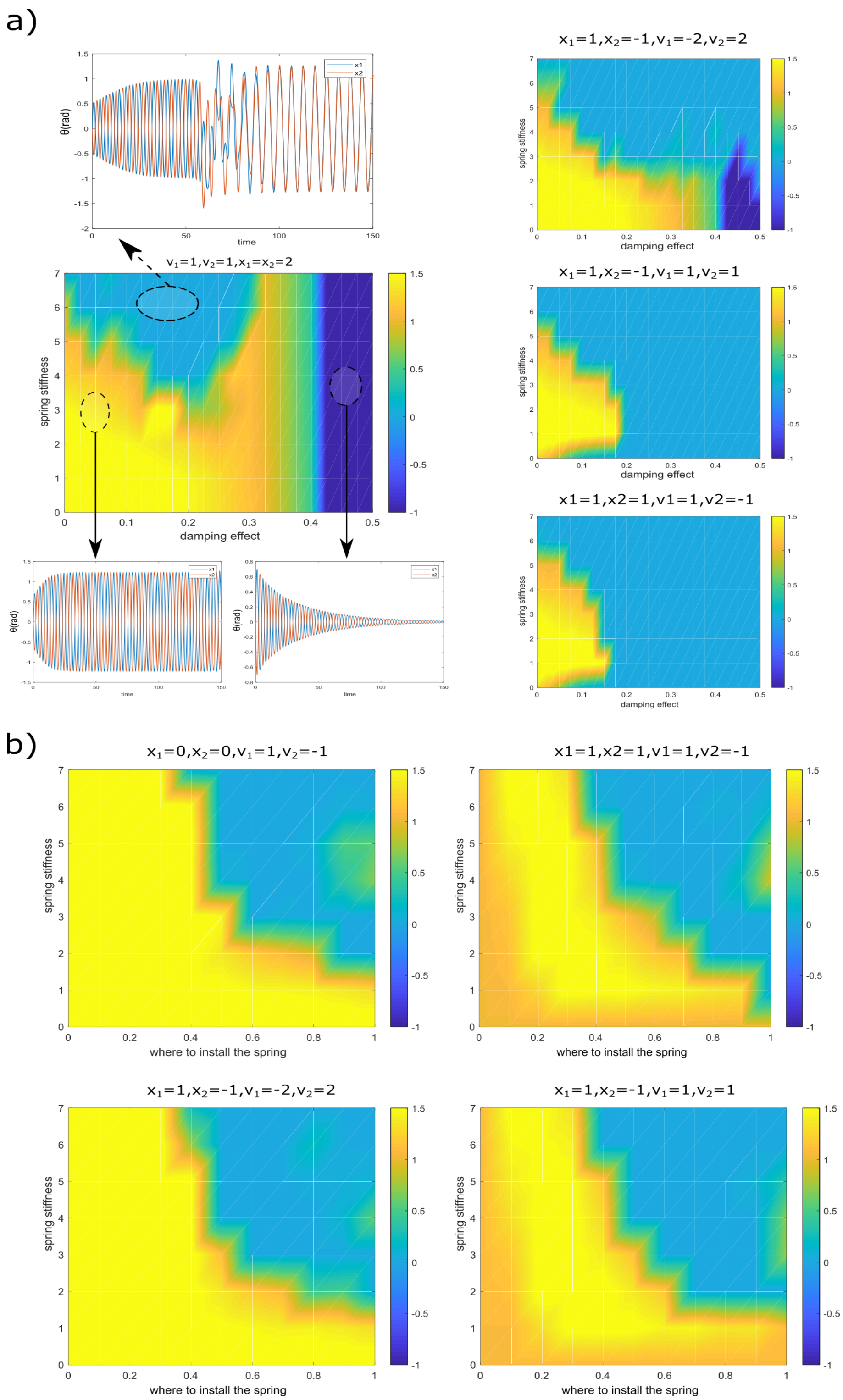

Figure 3.6. The heat map of the phase difference of the coupling system. (a) The X label is damping effect between the robots,Y label is spring stiffness (b) X label is the position where the spring installs, Y label is spring stiffness.

After we get the mathematical expression for the dynamic system in 2D space. The expression of the function is too complicated so it is very hard to grab an analytical solution to see how the spring characteristics influence the whole oscillator-spring system. Taking into account this factor, we try to use heatmap to see the effect of the connection part from a perspective perspective. The Figure 3.6 shows how spring stiffness, damping effect between robots and the mounting point for the spring affect the phase difference of the system.In general, there are three situations. The yellow region means there are quite large phase differences during the oscillating process. Then the light blue area means the system totally becomes synchronized. Finally the dark blue region represents the system stop oscillating.

We could summarize the rules from the heatmap. Firstly, the damping effect greatly determines whether the system will stop oscillating. Then the mounting point and spring stiffness will determine whether the system will be synchronized. With longer distance between mounting points and higher stiffness, the system will more easily reach synchronization. These two rules are very reasonable. The damping effect will lead to energy loss so if the damping effect is strong, the system couldn't keep oscillating. And no matter the higher stiffness nor the longer distance, it will cause larger torque generated by the spring, And the connecting spring is the most important part, forcing the system to become synchronized.

3.3 Transformation from In-phase to anti-phase

The mathematical analysis we discussed before mainly focused on how to influence whether to keep oscillating or not. The spring stiffness and damping effect could vastly effect it.

However, from the perspective of making locomotion robots, we think it is more valuable to discuss how the phase difference of coupling evolves when we change some system characteristics. It will make the locomotion robots' system have some particular movement and then generate some particular gait.

Based on the heatmap we got before, it is undoubtedly that the phase difference of a robot is very sensitive to the changing of physical connection. And in the simulation environment we find if we change the distance between robots on the rail, the robots will transform between in-phase and anti-phase.It is a pretty interesting phenomenon,because just by changing the relevant distance between robots we could make the wave from the robots' system have totally different state like Fig() shows.

We further our simulation in this direction.When the robots' distance is equal to the original length of the spring, we could treat the robots at the equilibrium point. We could see that when the system is at its neutral position, the robots have a strong tendency to become synchronized even if they have different initial values.

3.3.1 Mathematical analysis

Now, we want to know how the phenomenon happens, we need to know the logic behind it.

It is also difficult to analyze the system in the 2D system so we still need to transform it as the spring-block in 1D space(Figure 3.7.But it is different from the calculation we did before for the Van Der Pol controller. Because now we consider the phase difference, the state of the system will not be the position and velocity of the motor.We need to transform the states from PRCS to QCS.

And we want to know how the distance between the robots influence the system , so the oscillating center of each robot couldn't be treated as the origin any more. It should be based on the initial position of the blocks. Then we come to a very important assumption. Because the distance change of the robots is very relevant to the torque generated by the robots. This is the most important part in the calculating process. And the damping part of the oscillator will not be very strong. We could set the scaler c and μ to be small enough,which means. And the nonlinear strength is relatively small,as a result we suppose the radius be a constant R. It will greatly reduce the complexity of calculation.

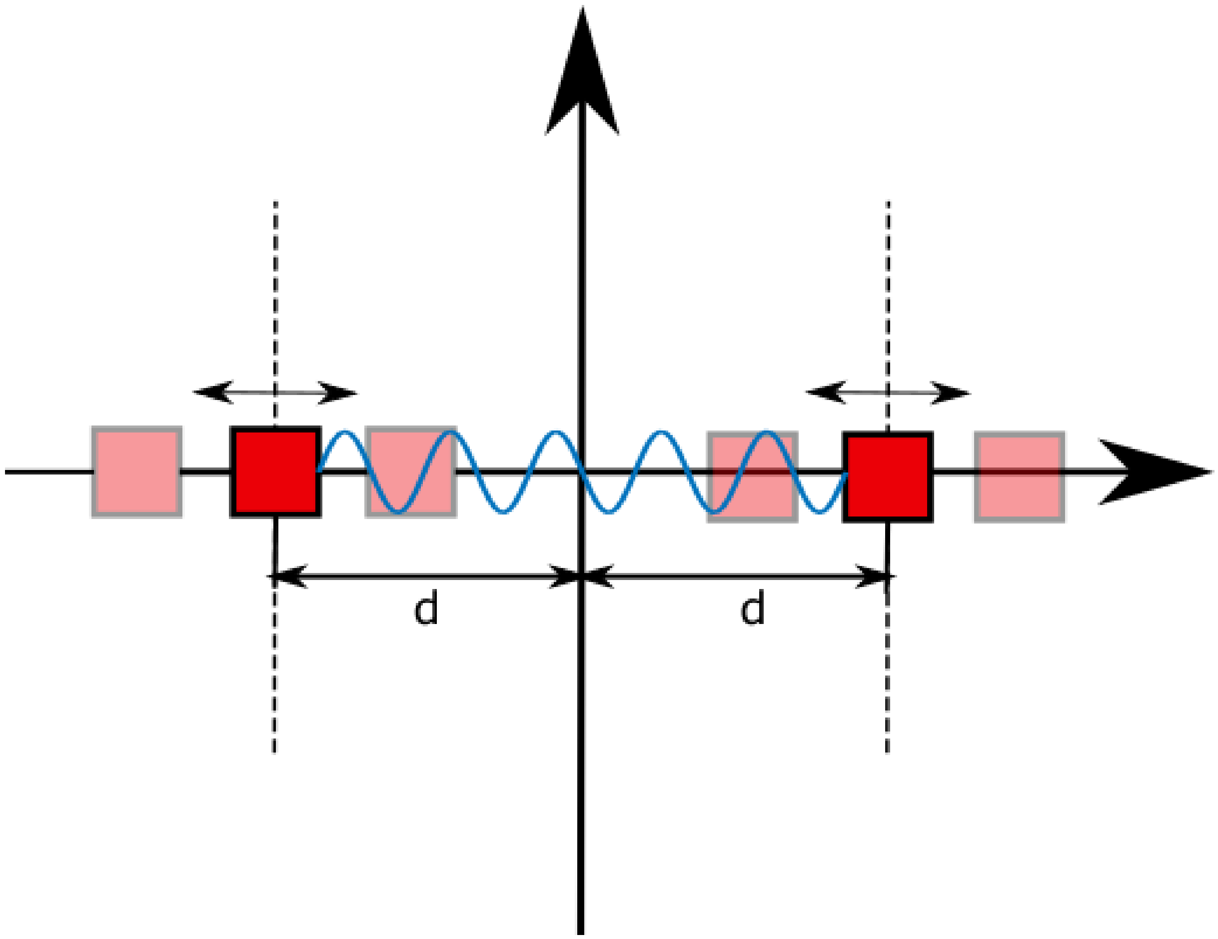

Figure 3.7. The illustration of the block-spring system, *d* represents the distance between the oscillating center of the block and the origin.

$$
\begin{aligned}
x_1 &= R * \cos\psi_1 + d \\
v_1 &= R * \sin\psi_1 \\
x_2 &= R * \cos\psi_2 - d \\
v_2 &= R * \sin\psi_2
\end{aligned}
\tag{3.34}
$$

In the Equation 3.34, *d* represents the the distance between the origin and the oscillating center of the block(Figure 3.7) And based on it, the phase of the system is:

$$\begin{aligned}\psi_1 &= \arctan\frac{v_1}{x_1-d}\\ \psi_2 &= \arctan\frac{v_2}{x_2+d}\end{aligned} \tag{3.35}$$

Then the derivative of the phase is

$$\begin{aligned}\dot{\psi_1} &= \arctan\frac{v_1}{x_1-d} = \frac{\dot{v_1}(x_1-d)-\dot{x_1}v_1}{x_1^2+v_1^2}\\ &= \frac{-k*(R*\cos\psi_1)^2+[c-\mu(R\cos\psi_1)^2]R^2*\cos\psi_1\sin\psi_1}{R^2}\\ &-\sin\psi_1+\frac{d\dot{v_1}-k_s*R\cos\psi_1(R\cos\psi_1-R\cos\psi_2+2d-l)}{R^2}\\ &= -k*\cos^2\psi_1-\sin^2\psi_1-k_s(\cos^2\psi_1-\cos\psi_2\cos\psi_1)-\frac{2d-l}{R}k_s\cos\psi_1+\frac{d*k\cos\psi_1}{R}\\ &+\frac{d*k_s(R\cos\psi_1-R\cos\psi_2+2d-l)}{R^2}\end{aligned} \tag{3.36}$$

And $\dot{\psi_2}$ is very similar with $\dot{\psi_1}$

$$\begin{aligned}\dot{\psi_2} &= -k*\cos^2\psi_2-\sin^2\psi_2+k_s(\cos\psi_1\cos\psi_2-\cos^2\psi_2)+\frac{2d-l}{R}k_s\cos\psi_2-\frac{d*k\cos\psi_2}{R}\\ &+\frac{d*k_s(R\cos\psi_1-R\cos\psi_2+2d-l)}{R^2}\end{aligned} \tag{3.37}$$

And if we don't ignore the nonlinear damping part the equation could be written like it :

$$\dot{\psi_1} = -k * \cos^2\psi_1 - \sin^2\psi_1 - k_s(\cos^2\psi_1 - \cos\psi_2\cos\psi_1) - \frac{2d-l}{R}k_s\cos\psi_1 + \frac{d*k\cos\psi_1}{R}$$
$$+\frac{d*k_s(R\cos\psi_1 - R\cos\psi_2 + 2d - l)}{R^2} + (\frac{c}{2} - \frac{\mu R^4}{4})(\sin 2\psi_1 - \sin 2\psi_2)$$
$$-\frac{\mu R^4}{8}(\sin 4\psi_1 - \sin 4\psi_2)$$
$$\dot{\psi_2} = -k * \cos^2\psi_2 - \sin^2\psi_2 + k_s(\cos\psi_1\cos\psi_2 - \cos^2\psi_2) + \frac{2d-l}{R}k_s\cos\psi_2 - \frac{d*k\cos\psi_2}{R}$$
$$+\frac{d*k_s(R\cos\psi_1 - R\cos\psi_2 + 2d - l)}{R^2} + (\frac{c}{2} - \frac{\mu R^4}{4})(\sin 2\psi_1 + \sin 2\psi_2)$$
$$-\frac{\mu R^4}{8}(\sin 4\psi_1 + \sin 4\psi_2) \tag{3.38}$$

Just like we mentioned before, the damping effect of the oscillator is relatively small, so as a result, for simplifying the calculation, we directly ignore the nonlinear damping part.And in this part we focus on the phase difference evolution. So we should treat the phase difference as the new state variable. It means we need to rewrite the new equation based on the new variable:

$$\begin{aligned} z_1 &= \psi_1 - \psi_2 \\ z_2 &= \psi_1 + \psi_2 \end{aligned} \tag{3.39}$$

z_1 and z_2 are the new variable.

As a result, we could get the derivative of phase difference,which stands for the dynamic system. For the new equation, we just pay attention to the spring force part related to the distance between the robots

$$\begin{aligned} \dot{z_1} &= (k - 1 + 2 * k_s) * \sin z_1 * \sin z_2 - \frac{2k_s(2d-l) * \cos\frac{z_1}{2}\cos\frac{z_2}{2}}{R} \\ \dot{z_2} &= 2 * k_s * \cos\frac{(z_1+z_2)}{2} * \cos\frac{(z_2-z_1)}{2} - k_s(1 + \cos z_1 \cos z_2) - \cos z_1 * \cos z_2 + 1 \\ &+ \frac{2k_s(2d-1)\sin\frac{z_1}{2}\sin\frac{z_2}{2}}{R} \end{aligned} \tag{3.40}$$

The system with the nonlinear damping part, the equation could be written as:

$$
\begin{aligned}
&\dot{z_1} = (k-1+2*k_s)*\sin z_1 * \sin z_2 - \frac{2k_s(2d-l)*\cos\frac{z_1}{2}\cos\frac{z_2}{2}}{R} + \\
&(c-\frac{\mu R^4}{2})\cos z_2 \sin z_1 - \frac{\mu R^4}{4}\cos 2z_2 \sin 2z_1 \\
&\dot{z_2} = 2*k_s*\cos\frac{(z_1+z_2)}{2}*\cos\frac{(z_2-z_1)}{2} - k_s(1+\cos z_1 \cos z_2) - \cos z_1 * \cos z_2 + 1 \\
&+\frac{2k_s(2d-1)\sin\frac{z_1}{2}\sin\frac{z_2}{2}}{R} + (c-\frac{\mu R^4}{2})\sin z_2 \cos z_1 - \frac{\mu R^4}{4}\sin 2z_2 \cos 2z_1
\end{aligned}
\tag{3.41}
$$

We should know difference z_1 will converge to π and as the same principle, if the system is in-phase, z_1 have the tendency to converge to zero.

So for different situations, the system has different equilibrium points.

The next step is to linearize the system around the equilibrium points separately. The Jacobian matrix is :

$$
\begin{bmatrix}
(k-1+2k_s)*\cos z_1 \sin z_2 + \frac{k_s(2d-l)*\sin\frac{z_1}{2}*\cos\frac{z_2}{2}}{R} & (k-1+2k_s)*\sin z_1 \cos z_2 + \frac{k_s(2d-l)*\cos\frac{z_1}{2}*\sin\frac{z_2}{2}}{R} \\
A_1 & A_2
\end{bmatrix}
$$

$$
\begin{aligned}
A_1 = &-k_s \sin\frac{z_1+z_2}{2}\cos\frac{z_2-z_1}{2} + k_s\cos\frac{z_1+z_2}{2}\sin\frac{z_2-z_1}{2} + (k_s+1)\sin z_1 \cos z_2 + \\
&\frac{k_s(2d-l)*\cos\frac{z_1}{2}*\sin\frac{z_2}{2}}{R} \\
A_2 = &-k_s \sin\frac{z_1+z_2}{2}\cos\frac{z_2-z_1}{2} - k_s\cos\frac{z_1+z_2}{2}\sin\frac{z_2-z_1}{2} + (k_s+1)\cos z_1 \sin z_2 + \\
&\frac{k_s(2d-l)*\sin\frac{z_1}{2}*\cos\frac{z_2}{2}}{R}
\end{aligned}
$$

And when the system is in-phase, the matrix could be expressed as:

$$
\begin{bmatrix}
(k-1+2k_s)\sin z_2 & \frac{k_s(2d-l)\sin\frac{z_2}{2}}{R} \\
\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R} & \sin z_2
\end{bmatrix}
\tag{3.42}
$$

And when it is anti-phase:

$$\begin{bmatrix} -(k-1+2k_s)\sin z_2 + \frac{k_s(2d-l)\cos\frac{z_2}{2}}{R} & 0 \\ -2k_s\cos\frac{Z_2}{2}\sin\frac{z_2}{2} & -(k_s+1)\sin z_2 + \frac{k_s(2d-l)\cos\frac{z_2}{2}}{R} \end{bmatrix} \tag{3.43}$$

When we get the eigenvalue of two matrix, we could find there is a obvious difference. for the matrix 3.42 above

$$[\lambda_1 - (k-1+2k_s)\sin z_2][\lambda_2 - \sin z_2] = (\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R})^2 \tag{3.44}$$

for the matrix 3.43 below

$$\begin{aligned} [\lambda_1 + (k-1+2k_s)\sin z_2 - (\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R}][\lambda_2 + (k_s+1)\sin z_2 - (\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R}] = 0 \\ [(\lambda_1 + k - 1 + 2k_s)\sin z_2 * [\lambda_2 + (k_s+1)\sin z_2] - [\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R}][\lambda_1 + \lambda_2 + (k+3k_s)\sin z_2] = \\ -(\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R})^2 \end{aligned} \tag{3.45}$$

we could see for the in-phase and anti-phase situations, the the most obvious difference is the sign of right hand(the constant part) of the equation, and this change greatly influences the eigenvalue of the matrix.For example, when the system stays in its in-phase equilibrium, it means that the eigenvalue of Eq() should be less than zero. And we notice the right hand side of the part $(\frac{k_s(2d-l)\sin\frac{z_2}{2}}{R})^2$, If (2d-l) is not zero, this part must be more than zero. And because it is more than zero, it will cause the real part of the matrix's eigenvalues to be more than zero, which leads to the instability. According to this analysis, we know if we want the system to become in-phase, we should let the d value be equal to $\frac{l}{2}$, which means we need to control the distance between two robots should be equal to the original length of the spring.But for the equation below, because the sign of the right hand is negativethe result obtained by opening the root sign is an imaginary numberwhich will not influence the real part of the eigenvalues. So we could see why the distance between robots this will greatly influence the system behaviour

And if we don't ignore the nonlinear damping part of the oscillator.The equation will be much complicated . So we directly analyze the equilibrium points of the system instead of considering the Jacobian matrix.

We still pay attention to the phase difference z_1 and treat z_2 as a constant. No matter how big μ and c are, from the expression of $\dot{z_1}$, we could see, when $2d - l \neq 0$, if we want the $\dot{z_1} = 0$ to find the equilibrium points, it must include $z_1 = \pi$. But when the robots are at its neutral position, in other word, $2d - l = 0$. The term included $\cos \frac{z1}{2}$ will be removed. As a result, except for $z_1 = \pi$, $z_1 = 0$ could also be the equilibrium points. In other word, when the robots is in its neutral position, the robots could become in-phase.

In short, it could we expressed like it:

$$\begin{cases} \bar{z_1} = 0, & \text{when} \quad 2d - l = 0 \\ \bar{z_1} = \pi & \text{at any situations} \end{cases}$$

But in the real model which we will discuss in the next session , we could see the system always become in-phase when the robots are at the neutral position. It nearly couldn't have any other stable points. It is strongly in conflict with the inference above.

One reasonable assumption is in reality the torque generated by the motor always has limitations. The limitation is based on the type of the motor we choose. And the force from the spring is usually very big. As a result, the motor sometimes couldn't overcome the spring force to maintain anti-phase. Because as we know if we want the system to become anti-phase around the neutral position we need to compress and scratch the spring every oscillating cycle. The motor couldn't reach the goal so the system couldn't stay in anti-phase and of course finally came to in-phase.

So in order to test the assumption, we do the simulation in Matlab. Like Figure 3.8 shows,we choose suitable initial states and control parameters. If we don't add torque limitation for the Van Der Pol actuator, the phase difference of two coupling robots will converge to π, but if we apply the torque limitation, even if all the other stuff are same, the phase difference still comes to zero. The system is in-phase now.

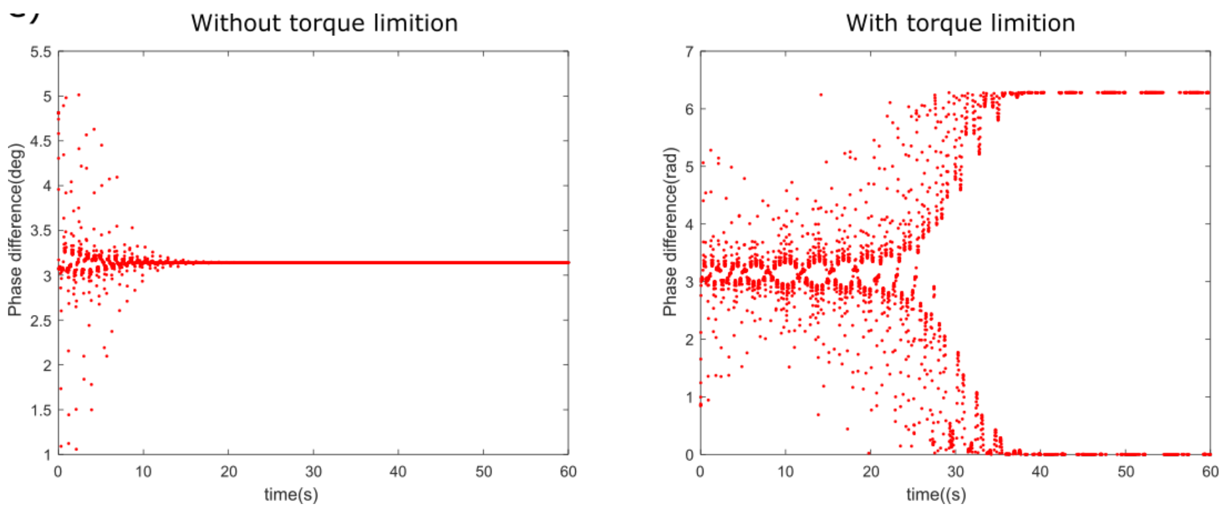

Figure 3.8. The effect of torque limitation of the actuator. When $2d-l$ is equal to zero,there are two equilibrium points of phase difference in theory. But if we set the torque output limit of the motor, the phase difference will always converge to zero, which is consistent with the real experiment data.

Then comes to the obvious difference between the oscillator with the nonlinear damping and without the nonlinear damping. From Figure 3.9 we could see, when we consider the nonlinear damping part, if $2d-l$ is not very big, it will take pretty long time to make the system become anti-phase or just couldn't reach anti-phase.In other word, the larger the value of $2d-l$, the faster for the system to be anti-phase stable. But if we don't consider the damping part, even if $2d-l$ is small but not equal to zero, it still could converge to zero very fast. AND what is more, if the nonlinear effect is big enough it could influence the stable phase difference of the system when the value of $2d-l$ is not equal to zero but small enough (Figure 3.9). And when the value gradually becomes bigger, the nonlinear damping part couldn't change the equilibrium point.

The nonlinear part could also directly affect how fast the system becomes stable when $2d-l \neq 0$. It will take longer for the system to become stable if the system has a relatively stronger nonlinear damping part inside the controller function.

It could provide instruction for the real model experiment.

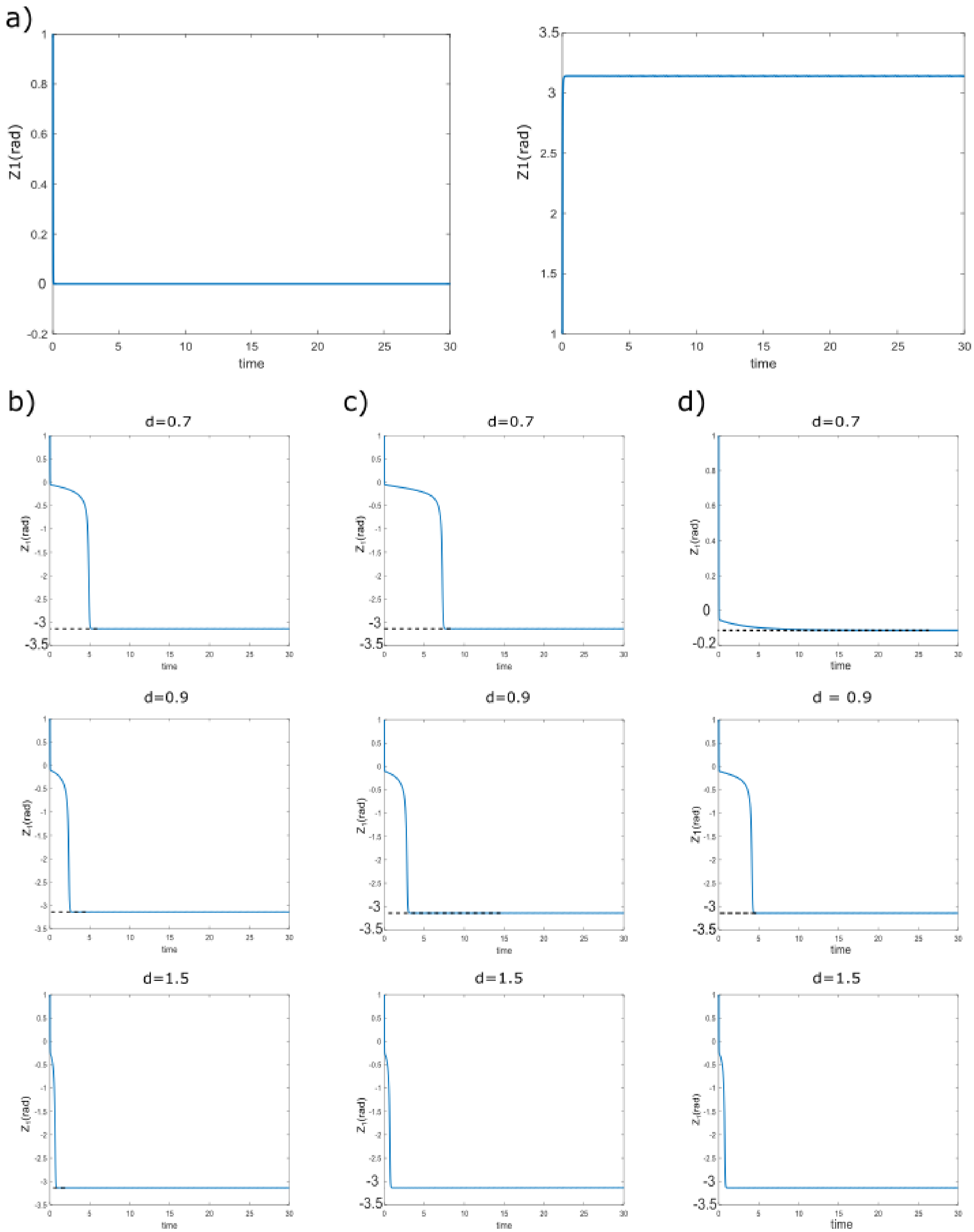

Figure 3.9. The simulation result of the new analysing equation (a) Two equilibrium points when $2d - l = 0$ (b)There is no nonlinear damping effect of the oscillator, the z_1's evolution when we change d. (c)There is weak nonlinear damping effect of the oscillator, the z_1's evolution.(d) There is strong damping effect of the oscillator, the z_1's evolution.

In collusion, we could say, the nonlinear part of the oscillator basically influences the time to stable and the value of the equilibrium points when the robots aren't at their neutral position. And the torque limitation generated by the motor will vastly affect the equilibrium points of the system when robots are at the neutral position.

3.3.2 Simulation proof with randomly initial state value

In this section, we want to know how different initial values influence the behavior of the system. According to the section before, we know if we control the limitation of the torque generated and choose suitable spring stiffnessthe robots will always stay in-phase. Based on it, we randomly choose different initial values for the system to see how it affects the phase difference change. The Figure 3.10 shows the simulation results.The original length of the spring is still equal to 1. When the relative distance of two robots is $d = 2.6$, the robots aren't at their neutral position. And even if the system starts oscillating with different initial states, the phase difference always converges to π. It means the system becomes anti-phase. However, when the relative distance comes to $d = 1$, the robots are now just at their neutral position. We could see the phase difference now has the tendency to come to 0, which means the system becomes in-phase.

However the initial difference still has impact on the system,It will greatly affect the time it takes for the system to reach in-phase. We could see that when the initial phase difference is very close to π when $d = 1$, the system will take a pretty long time to become synchronized.

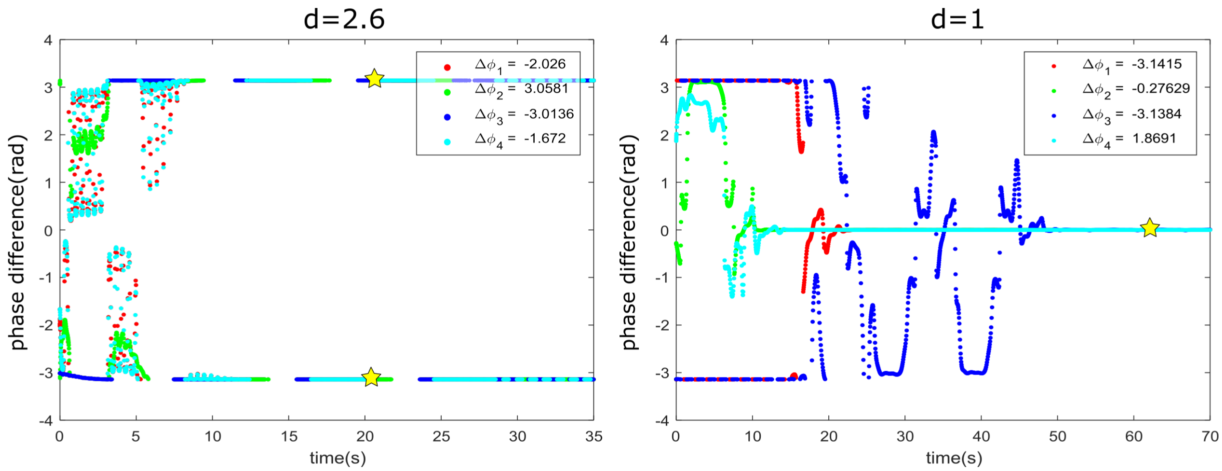

Figure 3.10. The phase difference evolution according to different initial state.

3.3.3 Set physical experimental platform

We have fully discussed the transformation from in-phase to anti-phase in 1D space, but it is very hard to testify it based on the motor we use.What is more, if we just focus on 1D space, it is very hard to apply it to the locomotion robots. So after all, we still need to discuss this phenomenon in 2D space. As the Figure shows, the one-freedom linkage robots are connected by a spring whose original length is l. .The joint of the robots has a particular distance of $2d$. We suppose the l will not change but d will change in the experiment. There are three typical situations corresponding to the analyzing result we got before. From Figure 3.11, we could know the oscillation center will changed depending on the value of d.

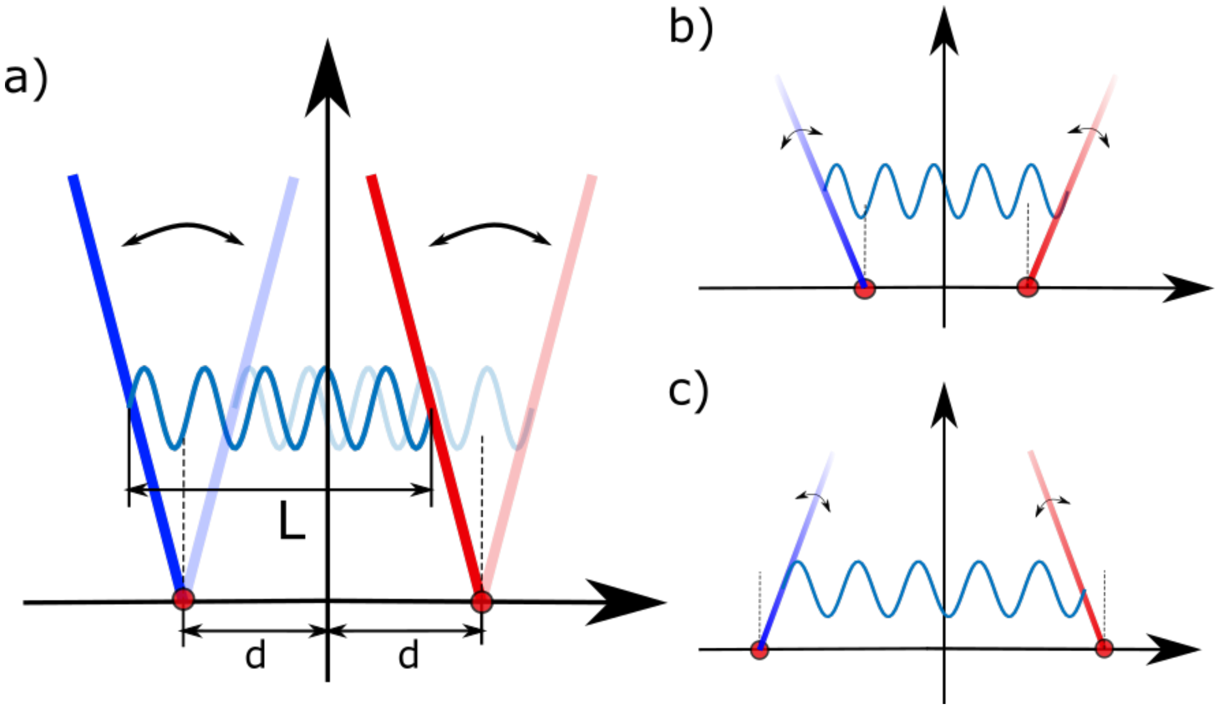

Figure 3.11. Three typical oscillation situations when changing the relative robots' distance in 2D space. (a)The robots are at their neutral position (b)The relative distance between robots is short. (c)The relative between robots is long

In session, we could know theoretically when $2d-l$ is not equal to zero, the coupling system will have the tendency to converge to *pi*, in other word, the two robots will become anti-phase(Figure 3.11 (b),(c)) and if $2d-l$ is equal to zero, the phase difference of the system will converge to zero, The two robots' in this situation will become in-phase(Figure 3.11 (a)).The theory is demonstrated by the calculation and simulation but still need to be further proved by the experiment.

So we need to set up a physical experiment environment that is similar to the illustration to test the result above. We firstly install the motor to the linkage and then fix one side of the robots on the ground.The robots just like Figure 3.12 (a) shows. And at the same time, we want the robots to change their distance between each other. So we also need to design the rail to let it happen. The robots could horizontal move on the rail.

We also need to design the connection part.The spring we bought could be basically divided into two parts the extension spring and the compression spring. And the spring will have different stiffness when it is extended or compressed. So if we want the spring connection don't change the stiffness during the experiment process,We need to connect the compression spring and extension in series or use rubber bands to make a two-sides spring.

Then we could observe how the system behaves when the distance is changing. In the experiment, we fixed one robot and moved the other robots from the neutral position to a relatively far distance. There are basically three situations as the robot is moving on the rail(Figure 3.12)

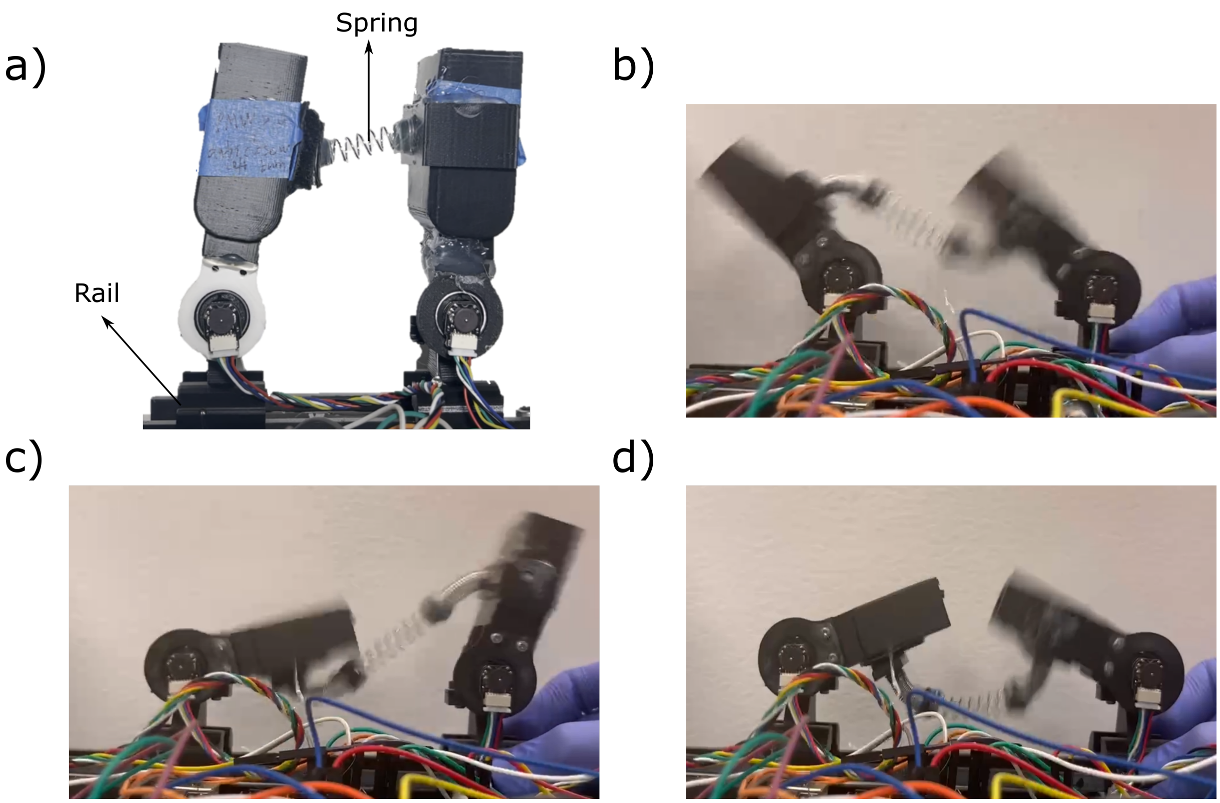

Figure 3.12. Physical experiment on the rail, moving the robots from neutral position to relative far distance. (a) The platform for the experiment. (b) The robots are in their neutral position, the system (c) The robots is in the state between the in-phase and anti-phase. (d) The robots are far away from each other, the system become anti-phase.

The average phase difference of each experiment is shown in Figure 3.13 =.The experiment result corresponds to the calculating analysis we got before. When $2d - l$ transforms from zero to nonzero value, the phase difference of the robots will rapidly increase from zero to π

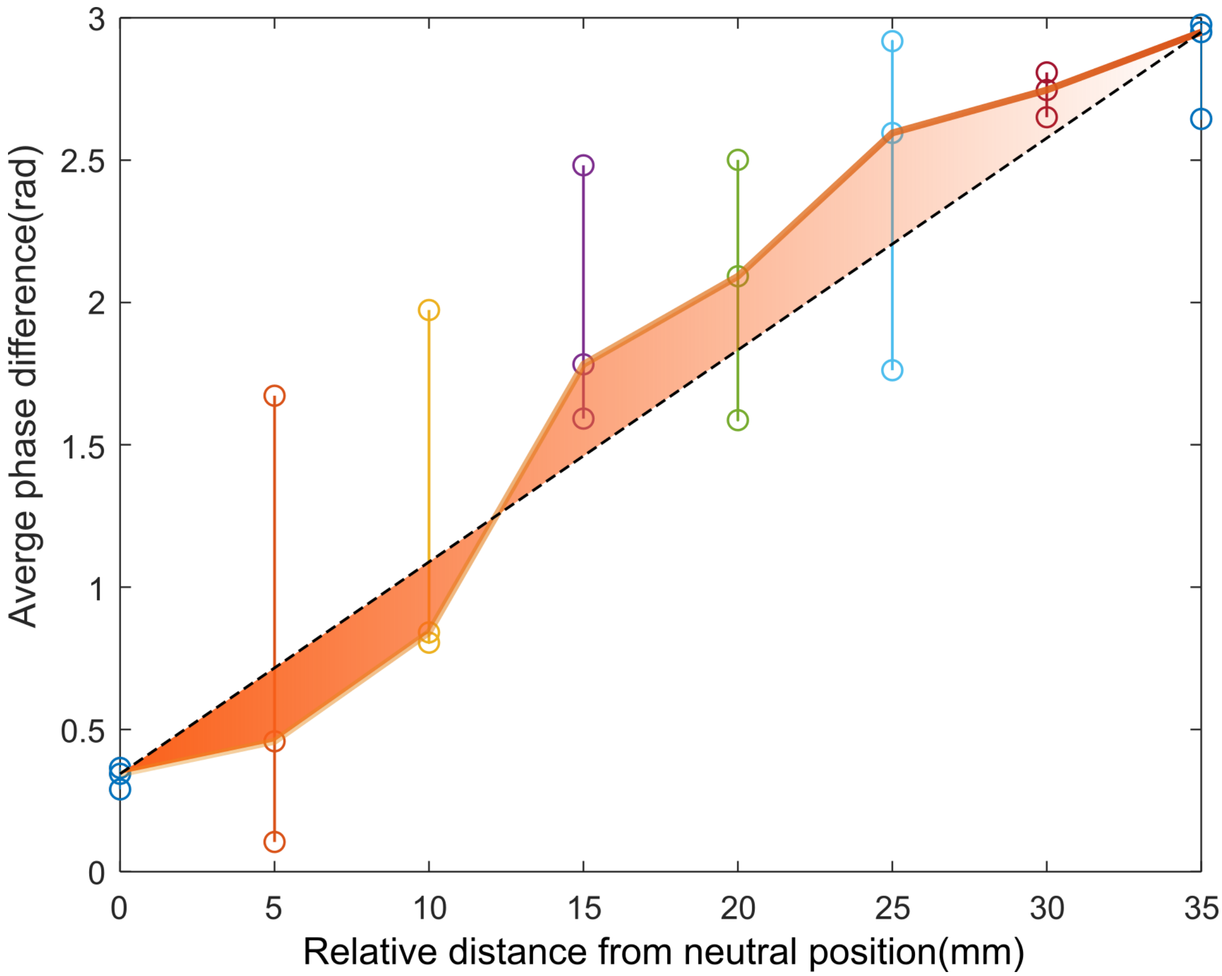

Figure 3.13. The conclusion of average phase difference when separate the robots from the neutral position. It explicitly explains how the relation of the robots' distance and spring's length influences the phase difference between robots.

The Figure 3.14 shows exactly how the joint angel would evolve during each experiment. We add distractions for each experiment manually. The plots exhibit the disturbance will not change the oscillating characteristics (e.g. frequency, amplitude) and the phase difference , which means the oscillator-spring system has strong capability to keep stable when encountering outside perturbation.

But when we gradually separate the two robots, the amplitude ,frequency and the oscillating center will apparently change. For example, when the robots are at the neutral position, the system will have the maximum amplitude and when the two robots go away, due to the constraint of the spring force, the robots will be oscillating at a relatively small range. It means that the relation of the spring length and robots' distance will vastly change the properties. We may use these characteristics to generate asymmetrical force and then design locomotion robots.

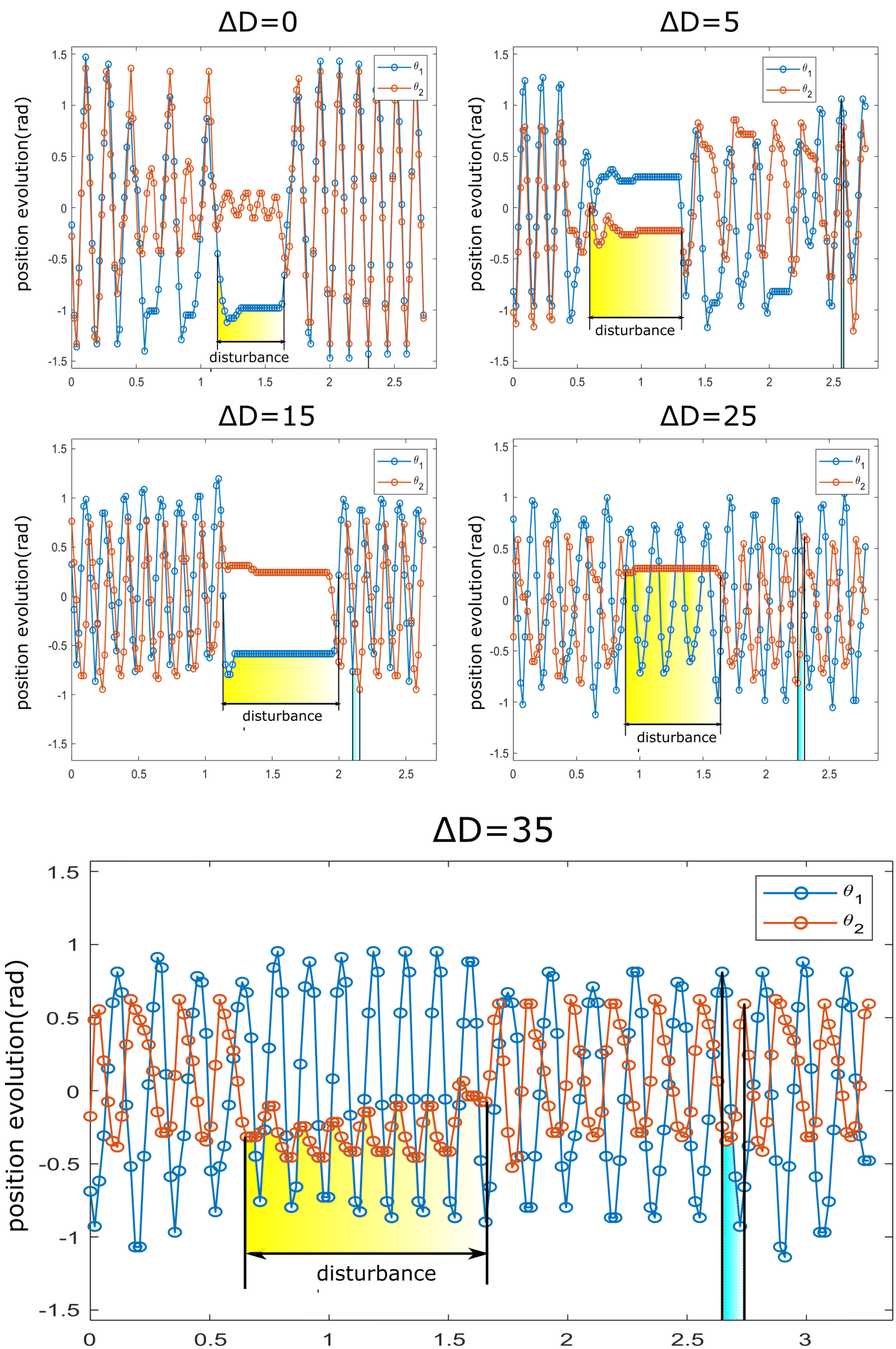

Figure 3.14. The position evolution for different robots' distance.The robots will reach synchronized firstly and gradually become anti-phase.We impose the disturbance(the yellow region) to test the stability of the oscillator-spring system.

3.3.4 Multi-robots Simulation

Then, we also want to know how the system behaves if we add more oscillating robots in the simulation environment.

First, we need to choose the suitable simulation environment. If we apply a mathematical method in Matlab to do the simulation, it will make the dynamic system become very high order, because each robot possesses two state variables. So we decided to do the simulation in the Pybullet, which could provide a relevant simple dynamic simulation environment. We could directly apply torque to the joint of each robot separately. Moreover,in the Pybullet, we could consider the effect of the friction generated by the joints, and the real shape of the linkage robots. It will make the simulation more realistic.

We use the same model we implemented in the physical experiment. And try to add more robots to the environment. The simulation environment is shown in Figure. The simulation contains five robots, and Figure 3.15 (a) stands for the final state when all the robots are in their neutral position. We could know that all the robots basically become in-phase. And Figure 3.15 (b) represents that the distance of all the adjacent robots is greater than the original length of the spring,which means all the robots aren't staying at their neutral position. Now, the system will not stay in-phase but it is also not total anti-phase like the result where there are just two robot coupling.

a)

b)

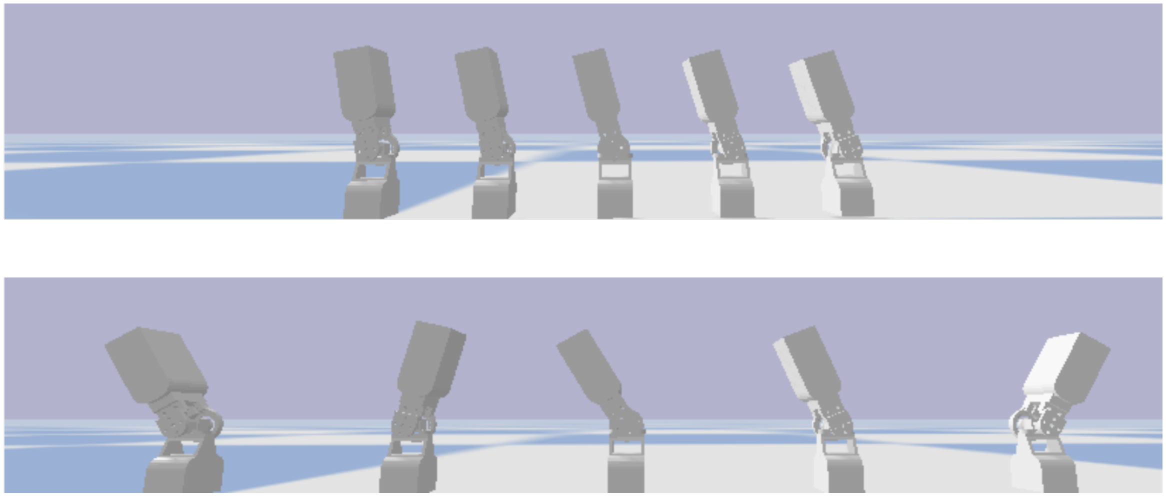

Figure 3.15. The multi-robots in python simulation environment. (a)The final state of system when the robots are in neutral position. (b)The final state of system when the robots aren't in neutral position.The phase difference between adjacent robots is randomly distributed.

Fig() shows the possible simulation result. The robots numbers goes from $N = 2$ to $N = 5$.

The Figure 3.16 (a) shows the situation when the distance between robots is longer than the original length of the spring. If there are just two robots in the system, the simulation corresponds to the result we got before. But when we tried to add more robots to the system, the adjacent robots couldn't keep anti-phase. The possible reasons is that unlike all the robots in phase, the spring is just at its original length, the spring in this situation will always be compressed and stretched. It means the spring system will not stay in its stable state. When there are many robots all staying in this state, the whole system will be very easily influenced by the outside disturbance like friction of the joints. So as the plot shows, when the number of robots in the system is more than two, the phase difference will keep oscillating meaning the system is not stable.

The Figure 3.16 (b) is all the robots at their neutral position, the situation will be much clearer. All the springs have a strong tendency to come to the original length and all the oscillators inside the robots will become in-phase under the action of the springs. The number of the robots could influence the time for robots to reach synchronization, but it is very hard to influence the stability of the system around the equilibrium.

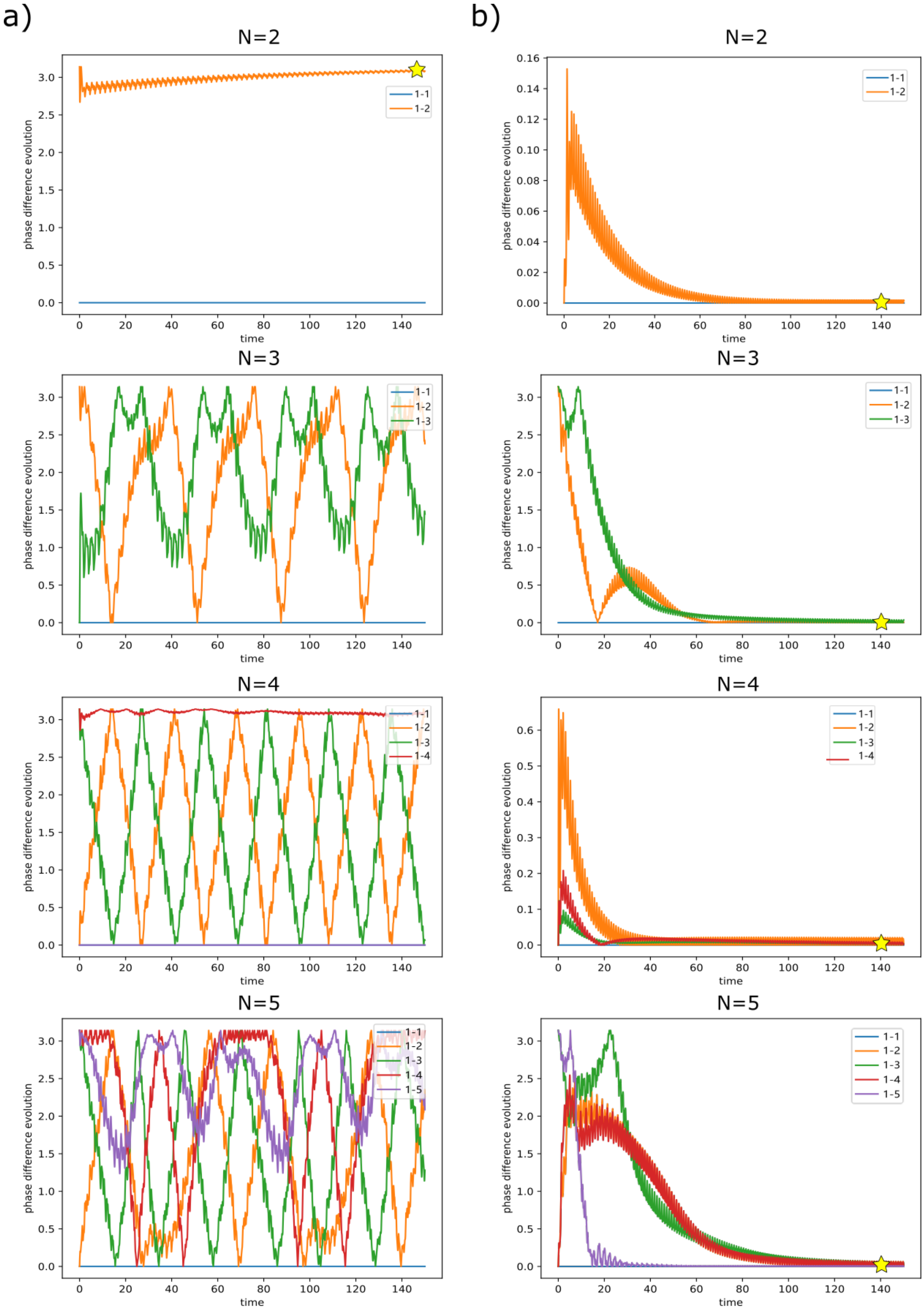

Figure 3.16. Multiple robots oscillating in the system. It shows the the simulation result with robots number $N = [2,3,4,5]$. (a) When robots aren't in its neutral position,the phase difference evolves. (b)When robots are in their neutral position, the phase difference evolve.

3.4 Float robots in simulation environment

Until now, we have been investigating robots in 1D and 2D space. And for 2D space, we all fix one side of the robots to the ground which will simplify the model. However, we still want to know if we can make all the robots float, what will happen? Could spring still help robots become synchronized? It is an important part because if we want to further the theory we got to the locomotion robots, we, at least, make the whole robots' body float.

So this time, we use a new model in the Pybullet. There are two springs connected to the two linkages(Figure 2.5). Based on the theory of forward kinematics (Figure 2.1), We could know the points where the springs are installed in the world frame and then know the present length of the spring to gain the force generated by it. And because the robots are floating, we add friction to the robots to observe the motion of the robots when we change the stiffness of the springs.

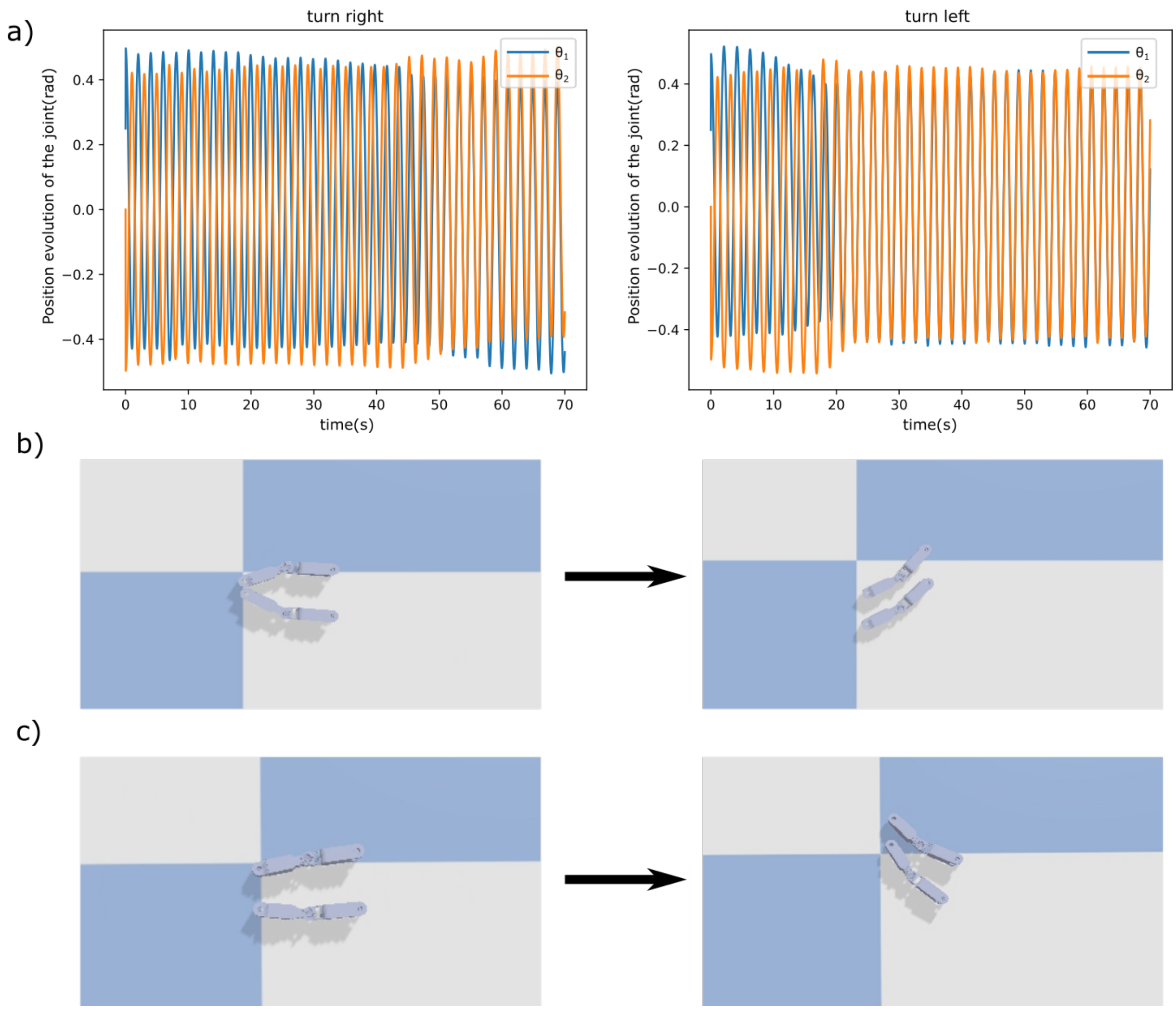

Figure 3.17. The simulation of locomotion robots driven by DC motor. (a) The evolution of joint's angel. (b) Robots turn left. (c) Robots turn right.

From Figure 3.17 (a), we can see that no matter which side of the spring is more stiff, the robots always go becoming synchronized, the simulation tells us even if we don't fix one side of robots to the ground and add one more spring to the other side, the DC motor still could synchronize the system. It is consistent with the locomotion activated by the Servo motor, which means we could also implement this method to fabricate locomotion robots.

What is more, there are two springs attached to the robots, and we set different stiffness for different springs. The robots are going to show the sign of turning(Figure 3.17 (b) and Figure 3.17 (c)). It edifies us that we could make direction controllable robots by altering the stiffness. The combination of this idea and the robots we have made in Chapter 2 will become the foundation of manufacturing locomotion robots in the next chapter.

3.5 Summary

We should make a conclusion of this chapter. In this chapter, we mainly investigate the torque control oscillation robots connected by spring. We build the dynamic function of the system and then analyze the function to know what influences the robot's system to keep oscillating. What is more, we further our study in researching what leads coupling robots not only to keep oscillating but also has a controllable phase difference,transforming the robots from in-phase to anti-phase,etc.

The most important conclusion is that the spring characteristics and the oscillator parameters could directly change the system behavior. And this change usually follows some rules which we could get from the dynamic function.

Then, we still want to make the comparison between torque control and position control. It could reveal some basic logic for the oscillator-based robots.

For position control, we use the Hopf oscillator and adaptation function to control the robots. The adaptation function includes the phase adaptation part and radius adaptation part.However, based on the experiment and simulation, the radius adaptation could just adjust the radius in a relatively small range, if it exceeds its range, the robots are going to stop oscillating or lose stability. As a result, we could say the robots could only efficiently adjust its phase. The shape of the limit cycle will not easily change if we don't change the control parameters of the oscillator. But, when we use DC motors and torque, the situation will be totally different, because the oscillator controller inner the motor generates the torque and the spring part also generates the torque. So the outside physical signal could directly affect the controller. The spring connection is like a negative feedback controller that influences the output of the oscillator. So it is very easy to understand the limit cycle of the Van Der Pol oscillator will be affected by the torque generated by the spring readily.The assumption is fully confirmed by the collision test for the two motors. More specificallythe oscillation center, amplitude, frequency all are influenced by the outside signal such as distance between robots, original length of the spring and stiffness of the spring. And all these changes are regular, we could get the rules by analyzing the whole dynamic system.

These differences make the torque control method more easily react to the outside signal so it is more easily to have collective behavior due to the spring connection. What is more, the system could evenly become anti-phase in some situations and changes its amplitude and frequency adapt to the outside environment.

In the following, we pay much attention to the transformation from in-phase to anti-phase. More specifically we investigate how the relation of spring's original length and relevant distance influences the phase difference of the coupling robots.

The result is,in general, when robots are at their neutral position the robots will go in-phase and when it goes away from the position, it will gradually become anti-phase .

Chapter 4

Locomotion robots

In the project, the generation of periodic shape change in shell-like robots comes from the harmonic actuation source.Chapter 3 shows that the common method to generate autonomous oscillations is to actuate as limit cycle system by DC motor and how the physical characteristics of spring connection influence the synchronization of the system. In this chapter,we will combine it with the locomotion robots we made in Chapter 2 to design a locomotion robot motivated by DC motors. The silicon is utilized to connect the robots instead of the spring and scales are appended to the silicon to generate anisotropic friction.We use filling to change the stiffness of the silicon and based on it to do the motion planning for the robot. The robots could adjust its phase and frequency just by the validation of the silicon without sensor feedback and change the crawling pattern according to the existence of the filling part.Lastly, we control the locomotion robots to move and draw the trajectories to analyze.

4.1 Basic Logic

All the above experiments and analysis give us ideas about how the physical characteristics and controlling parameters influence the coupling robots based on the oscillator controller. We could see all these behaviors are controllable and predictable. The typical locomotion robots are very complicated in control method. Even using the CPG control strategy, a particular motor still needs to know the other motors' state to adjust its phase. It greatly increases the complexity in calculation when robots are working, which may cause instability. But now, we know for some models we could use physical connection instead of signal exchange. What is more, the influence of the physical is not totally uncontrollable. We found that the robots follow some basic rules. It inspires us to utilize this strategy to manufacture locomotion. Like we discussed before, using spring connecting linkage robots, make them become synchronized, and by changing the stiffness of the spring,we could change the collective behavior of the robot's system. So the robot could change its moving direction. The movement doesn't involve any changes of the controller so the controller doesn't need any complicated calculations and just focuses on its own state, which greatly improves the stability and adaptability of the system. So could we design a locomotion robots that adjust the moving behavior following the outside physical change become the most important goal we want to achieve.

4.2 Simulation and Structure Design

4.2.1 Drive Method

We have already made attempts to make locomotion robots in the position control part. It is motivated by the servo motor and connected by the spring.With the anisotropic friction,the robots could become synchronized and move in one direction. But just like we discussed before, we know there are some inevitable problems that can't be solved by this method. What is more, the volume of the servo is too big , and because of the big volume, some good mechanism could be applied to this motor. So we finally decided to use the DC motor to motivate the robots.

4.2.2 Simulation

Like the Figure 4.3 shows, we want to build this kind of robots, but this time we apply Dc motor and we use Van Der Pol oscillator inner the controller. Before

4.2.3 Mechanism Design

We now know we should use DC motors to motivate the system. However, there are still some problems we need to solve. One serious problem is that if we want the system to keep oscillating, we need to set a relatively high frequency and amplitude. However, if we still rigidly connect the robots by spring, the robots will move vigorously. We couldn't manufacture stable and reliable locomotion robots based on this movement. So that we should come up with some ways to make the oscillating become smooth. One way is to use silicone to connect the robots with silicon. Because silicon is very easy to adjust the shape so that the rigid linkage could be installed and also easy to be manufactured.

What is more important, silicon also possesses the similar physical characteristics with spring. It is elastic enough to store the energy and at the same time prevent the robot from vibrating very violently. The oscillating robots connected by the robots is a efficient way to smooth the oscillation and avoid wasting too much energy.

Then we could manufacture the silicon based on the size of the rigid robots. As Figure 4.1 shows, the rigid robots will be installed in the silicon. And the silicon could significantly constrain the amplitude of the oscillation, which makes the whole moving process become much more smooth. Then we add some empty space in the silicon. We could increase the stiffness of the silicon by installing the filling with suitable size .Based on the conclusion we got before, if we change the physical characteristics of the connection part, it will eventually influence the behavior of the robots and then influence the movement of the whole locomotion robots.

Figure 4.3 shows the position evolution of each robot with or without the filling. We could see that when the silicon has the filling, the two robots will more easily become synchronized and in the contrary, if there is no filling, the phase difference of two robots will become pretty big during the whole oscillating process.

Then just like the simulation and experiment we did before, if we want the in-phase oscillation robots to move in one direction, we need to make the whole robot system possess anistropic friction. We could use plastic scales(Figure 4.2 (b)) which could be treated as the leg part of the robots to generate the friction. Now another advantage of using silicon is revealed. Because we could install the leg part when the silicon is forming. So the leg is very closely connected to the silicon. What is more important,because silicon makes the two robots as a whole body. So we could evenly add the anistropic friction, which will make the direction movement become more stable.

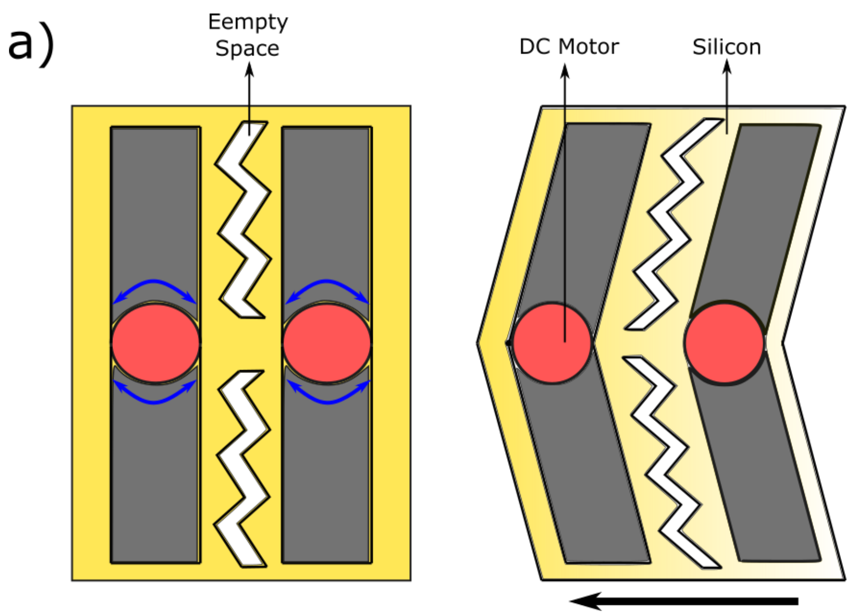

Figure 4.1. The illustration of the locomotion robots. The yellow region stands for the silicon. The white area is the empty space, we could change the stiffness of the silicon between the two rigid linkage by adding or removing the filling.Then because of the elastic silicon, the robots will become synchronized by receiving the local oscillating signal from the other robot.

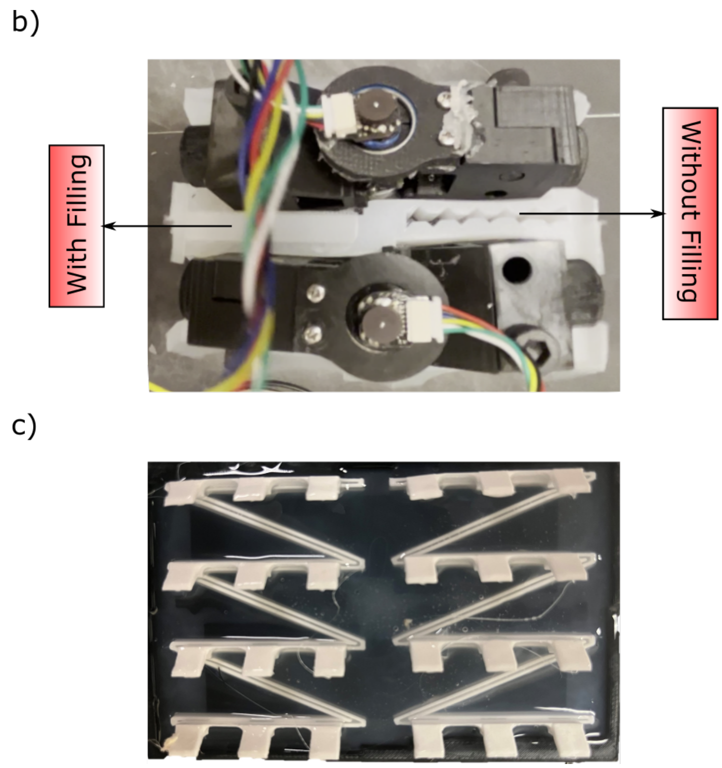

Figure 4.2. The structure of the locomotion robots. (b) The rigid linkage robots are installed in the silicon, there are empty spaces in the middle part of the spring so that we could install or remove the filling. (c) Install the scales when the silicon is forming to compose the leg part of the robot.It could generate anisotropic friction

4.3 Oscillation test Based on Variation of Silicone Stiffness

4.3.1 Setting Experiment

We determine the structure of the locomotion robots in the section before. We apply the result that the stiffness of the spring connection will cause the system to become synchronized,which we got by simulation and calculation. However, because now we use silicon instead of spring to connect the robots, we need to check the result by the real robots model. So, we did a controlled trial. We removed or added the silicon and used the same controlling function to vibrate the robots. Then compare the data of the angle of the joint collected from the encoder.

4.3.2 Date Collection and Analysis

The result is shown in Figure 4.3. We could see the difference is very obvious. When the silicon is filled, the whole system basically keeps synchronized but, in contrast, if we remove the filling, the robots couldn't stay in-phase anymore. What is more, the existence of the filling also influences the frequency. Just like the simulation result shows, with the relatively high stiffness, the system has higher frequency. The experimental data ,to some extent, shows the same result. Quantitatively speaking from the data plot, we know $T_1 = \frac{8}{7}T_2$.

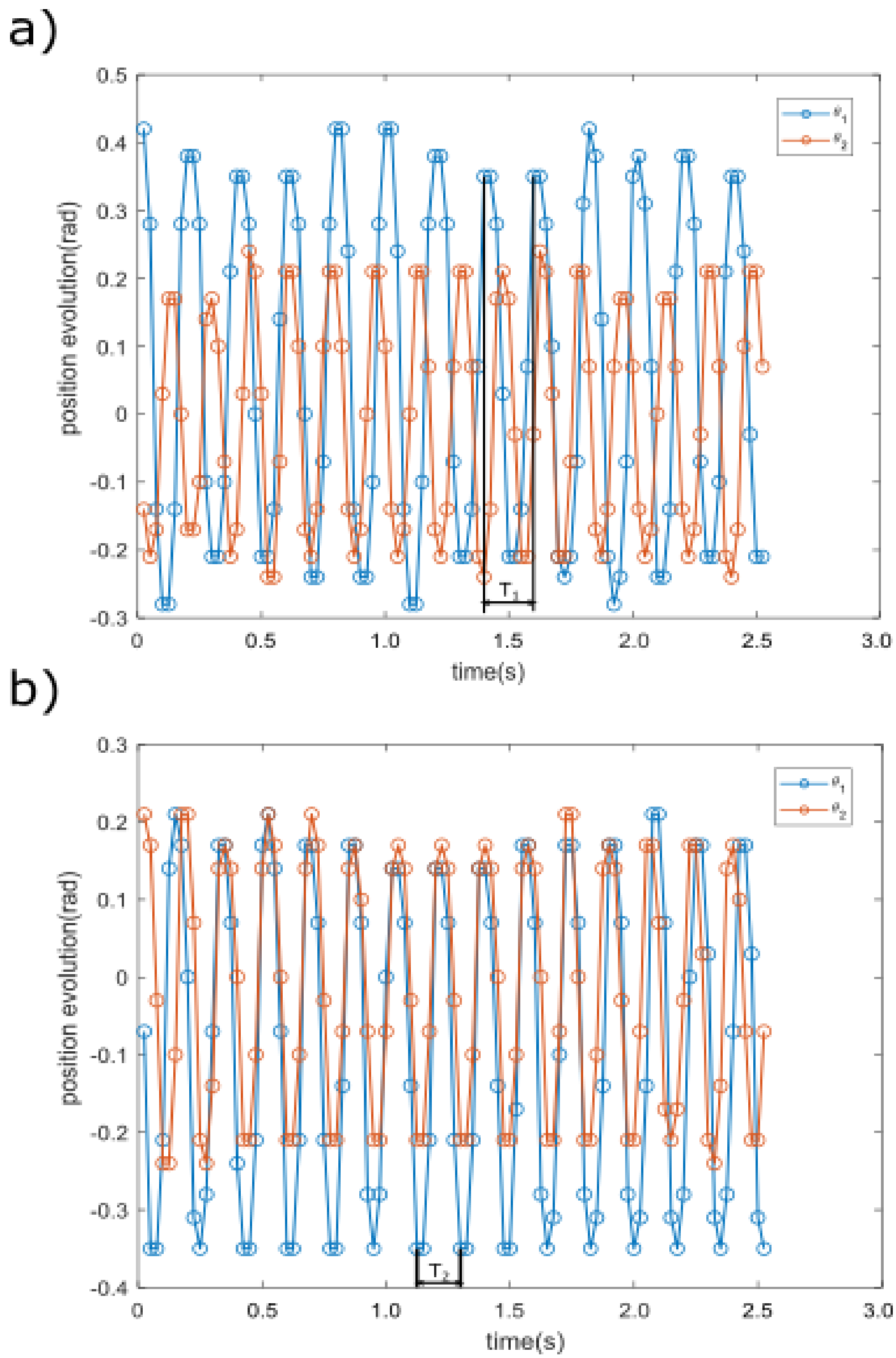

Figure 4.3. The comparison in oscillation based on different silicon stiffness. (a) The silicon is without the filling, the position evolution of two joints. (b) The silicon is with the filling, the position evolution of two joints.

With higher frequency and much easier to keep synchronized, the robots, in theory, have better mobility

4.4 Experiment On The Locomotion Robot

4.4.1 Motion Planning

We now know the filling could change the stiffness of the whole silicon structure, and this change accordingly changes the phase difference of the system. Based on the observation and the conclusion we got before, we could change the moving direction of the robots by changing the stiffness of the silicon instead of changing the controlling parameters, which is novel compared to the general locomotion robots.

So as Figure 4.4 shows, we are changing the robots' moving direction and speed by adding or removing the filling.

When the lift side of the silicon is filled with filling(Figure 4.4 (a) upper row), and right side maintains empty, the robot will turn right, and if we remove the left side filling and add the filling to the right side(Figure 4.4 (a) lower row), in contrast, is going left.

The possible reason behind it is that according to the Figure 4.4 (b) shows, when one side of the silicon is with filling, this side will have higher frequency than the other side and is also much easier to become synchronized. As a result,the side with filling could generate stronger anisotropic friction and eventually lead to turning

And when we add all the two filling to the robots to test the ability to move forward. It is high to correspond with our expectation, when the robot's both sides installed by the filling. It has a higher speed to move forward because of the higher frequency and the existence of synchronization.

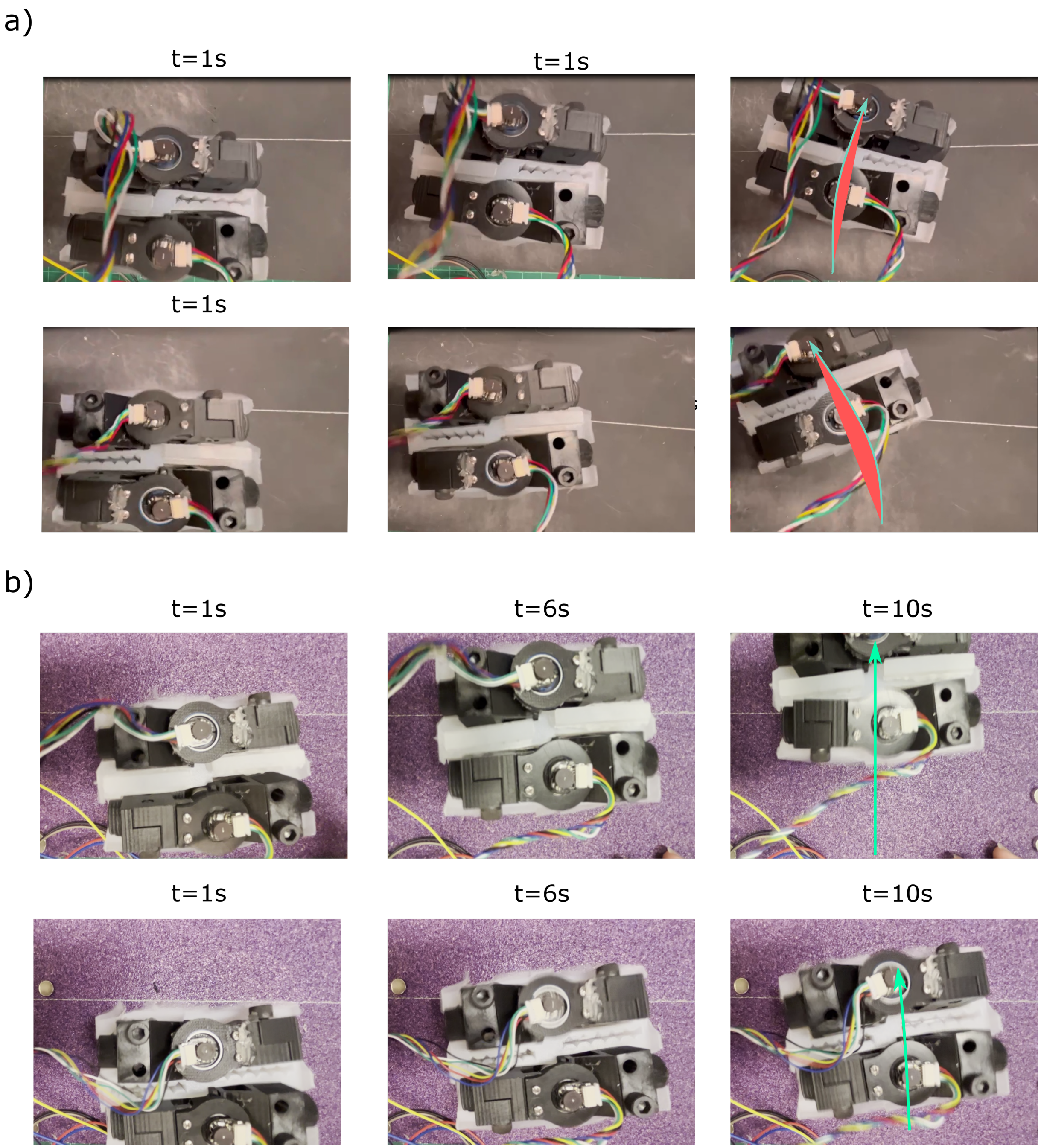

Figure 4.4. The motion planning for locomotion robots, changing the moving mode by adding or removing the filling. (a)The upper row, when the lift side of the silicon has the filling , the robots turn right. The lower row, the robots will turn left with right filling. (b) How the filling influences the speed of the robots. It shows how far the robots move in 10 seconds. The robots with two fillings (upper row) will move faster than the robots without filling (lower row).

In the end, we draw the conclusion that just like the simulation we did before, the stiffness of the connection part could directly influence the coupling robots to become synchronized and the robots' frequency. The locomotion robot could utilize these characteristics to control movement without changing any controlling parameter inside the robot. It greatly reduces the complexity of the algorithm with a relatively simple controlling method. The robots ,in theory ,could be adapted to rough environments more easily and be more difficult to be disturbed by external signals

4.4.2 Trajectory Generation

We repeated the experiment for many times. And drew the trajectory for each experiment. The experiment result could be divided into three situations like Figure 4.5 shows.

Nearly all the trajectories follow the basic rules from session 4.42. However, when the left side of the silicon is filled, not all the trajectories are all the way to the right. And when the robot has two fling, it will lean to the left when moving forward. So compare these two exceptions, we could know the leg part of the robot couldn't produce totally symmetric friction for the whole body, which may be caused by the manufacturing and installation errors. So that the robot has the tendency to tilt to one side. So some of the trajectories will violate the conclusion we get before.

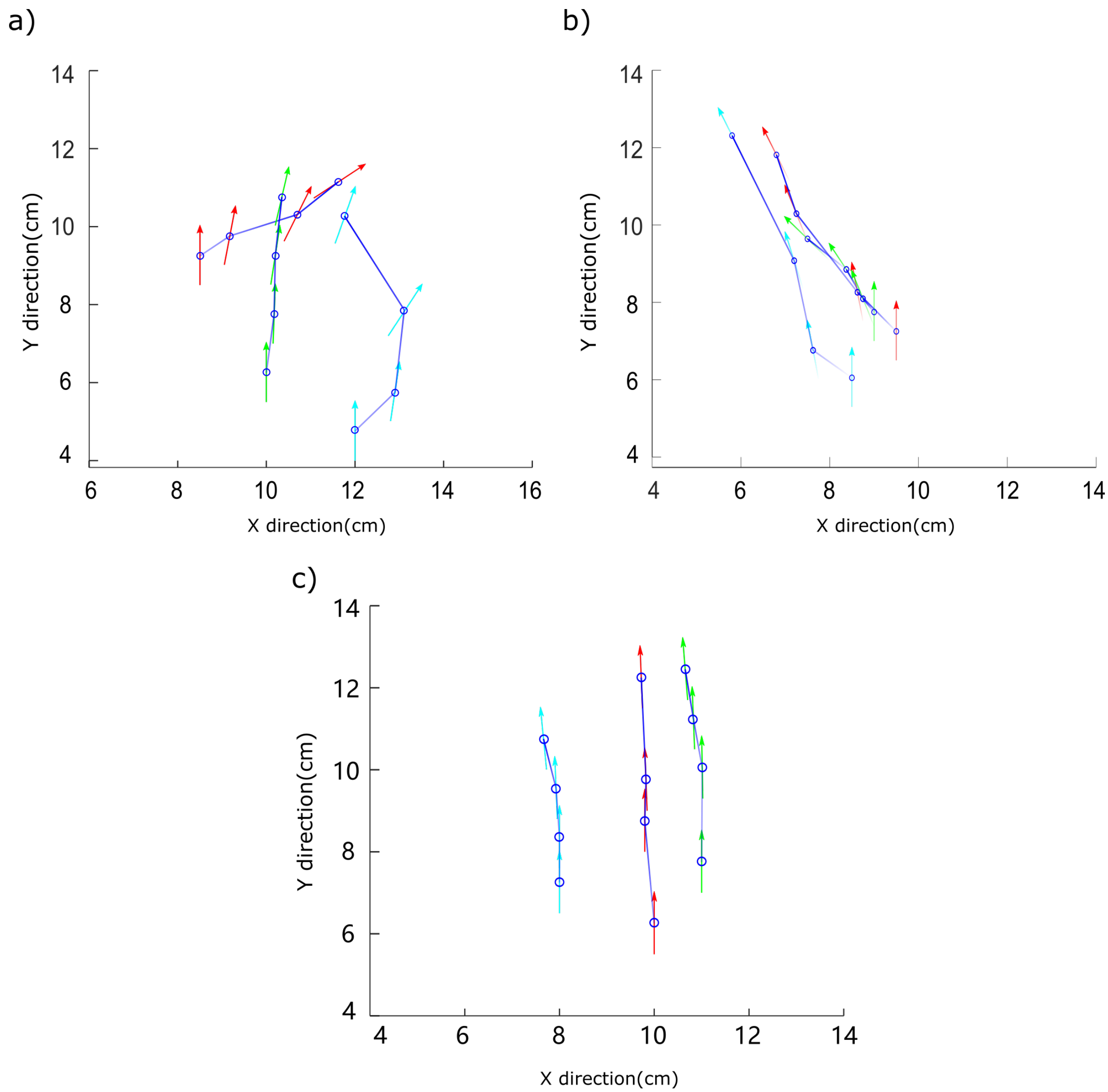

Figure 4.5. The trajectory of the locomotion robots. (a) The left side of the silicon is with the filling, (b) The right side of the silicon is with the filling. (c) The both sides of the silicon all have filling

4.5 Summary

Based on the theory we got before, the chapter develops a way to apply the theory to the locomotion robot. We replace the spring with silicon to form the synchronization gait and at the same time utilize scales to produce anisotropic friction. The gait and friction helps robots move like a shell. We also propose a way to change the stiffness of silicon and due to it, control the motion of the robot. Furthermore, we analyze the trajectory of each experiment to explains why there are exceptions that conflict with the theory.

Chapter 5

Conclusion

In this study, we demonstrate that oscillation motion in biologically inspired robots can be synchronized through contact interactions without the need for robot-robot communication. One of the efficient ways to deliver the contact information is to use spring. Then we use the spring connection to reach synchronization.We use Servo motors and DC motors with different kinds of oscillators to analyze how the physical characteristics of spring influences the behavior of the system and compare the difference with these two methods in detail. We also try to know what factors could directly determine the pattern of collective behaviors (the in-phase pattern and anti-phase pattern). Finally, we try to utilize the law we got from analyzing to design and manufacture locomotion robots. Overall, these results suggest new methods for designing feedback control of emergent synchronization in robot swarms and inspire us to combine oscillators (controlling strategy) with spring (mechanism) to create mobile robots with simple algorithms and strong adaptability.

And what is more intriguing is because the robots are linked by the silicon, it is very easy to be attached and separated. And in session 3.3.4, we realize that even for multi-robots connected together, all the robots in the system could become synchronized if we use suitable controlling parameters.As a result, we could treat each linkage as a unit and make modular robots composed by these units.

Actually, there are some works dealing with crawling robots based on the topology of modulate robots [23, 24]. It could be our future work.

Bibliography

[1] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, 2003.

[2] A. Sproewitz, R. Moeckel, J. Maye, and A. J. Ijspeert, "Learning to move in modular robots using central pattern generators and online optimization," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 423–443, 2008.

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

[4] L. Righetti and A. J. Ijspeert, "Design methodologies for central pattern generators: an application to crawling humanoids," in *Proceedings of robotics: Science and systems*, no. CONF, 2006, pp. 191–198.

[5] J. Buchli, L. Righetti, and A. J. Ijspeert, "Engineering entrainment and adaptation in limit cycle systems," *Biological Cybernetics*, vol. 95, no. 6, pp. 645–664, 2006.

[6] L. Righetti and A. J. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1585–1590.

[7] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[8] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, and R. Spigler, "The kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of modern physics*, vol. 77, no. 1, p. 137, 2005.

[9] L. Tsimring, N. Rulkov, M. Larsen, and M. Gabbay, "Repulsive synchronization in an array of phase oscillators," *Physical review letters*, vol. 95, no. 1, p. 014101, 2005.

[10] A. Ruiz, D. H. Owens, and S. Townley, "Existence, learning, and replication of periodic motions in recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 651–661, 1998.

[11] M. Galicki, L. Leistritz, and H. Witte, "Learning continuous trajectories in recurrent neural networks with time-dependent weights," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 741–756, 1999.

[12] M. Ajallooeian, J. van den Kieboom, A. Mukovskiy, M. A. Giese, and A. J. Ijspeert, "A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape," *Physica D: Nonlinear Phenomena*, vol. 263, pp. 41–56, 2013.

[13] Y. Yang, V. Marceau, and G. Gompper, "Swarm behavior of self-propelled rods and swimming flagella," *Physical Review E*, vol. 82, no. 3, p. 031904, 2010.

[14] F. Ginelli, F. Peruani, M. Bär, and H. Chaté, "Large-scale collective properties of self-propelled rods," *Physical review letters*, vol. 104, no. 18, p. 184502, 2010.

[15] S. Gueron, K. Levit-Gurevich, N. Liron, and J. J. Blum, "Cilia internal mechanism and metachronal coordination as the result of hydrodynamical coupling," *Proceedings of the National Academy of Sciences*, vol. 94, no. 12, pp. 6001–6006, 1997.

[16] C. Wollin and H. Stark, "Metachronal waves in a chain of rowers with hydrodynamic interactions," *The European Physical Journal E*, vol. 34, no. 4, pp. 1–10, 2011.

[17] B. Button, L.-H. Cai, C. Ehre, M. Kesimer, D. B. Hill, J. K. Sheehan, R. C. Boucher, and M. Rubinstein, "A periciliary brush promotes the lung health by separating the mucus layer from airway epithelia," *Science*, vol. 337, no. 6097, pp. 937–941, 2012.

[18] A. Gopinath and L. Mahadevan, "Elastohydrodynamics of wet bristles, carpets and brushes," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 467, no. 2130, pp. 1665–1685, 2011.

[19] J. Yuan, D. M. Raizen, and H. H. Bau, "Gait synchronization in caenorhabditis elegans," *Proceedings of the National Academy of Sciences*, vol. 111, no. 19, pp. 6865–6870, 2014.

[20] W. Zhou, Z. Hao, and N. Gravish, "Collective synchronization of undulatory movement through contact," *Physical Review X*, vol. 11, no. 3, p. 031051, 2021.

[21] Z. Hao, W. Zhou, and N. Gravish, "Proprioceptive feedback design for gait synchronization in collective undulatory robots," *Advanced Robotics*, pp. 1–16, 2022.

[22] S. V. Zuev, "Non-limit integration of differential equations. general solution for van der pol equation," *arXiv preprint arXiv:1305.3562*, 2013.

[23] H. Hamann, J. Stradner, T. Schmickl, and K. Crailsheim, "A hormone-based controller for evolutionary multi-modular robotics: From single modules to gait learning," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.

[24] B. Deng, M. Zanaty, A. E. Forte, and K. Bertoldi, "Topological solitons make metamaterials crawl," *Physical Review Applied*, vol. 17, no. 1, p. 014004, 2022.