

ON FRICTIONAL MECHANICAL SYSTEMS AND THEIR COMPUTATIONAL POWER *

JOHN H. REIF[†] AND ZHENG SUN[‡]

Abstract. In this paper we define a class of mechanical systems consisting of rigid objects (defined by linear or quadratic surface patches) connected by frictional contact linkages between surfaces. (This class of mechanisms is similar to the Analytical Engine developed by Babbage in 1800s except that we assume frictional surfaces instead of toothed gears.) We prove that a universal Turing Machine (TM) can be simulated by a (universal) frictional mechanical system in this class consisting of a constant number of parts. Our universal frictional mechanical system has the property that it can reach a distinguished final configuration through a sequence of legal movements if and only if the universal TM accepts the input string encoded by its initial configuration. There are two implications from this result. First, the robotic mover’s problem is undecidable when there are frictional linkages. Second, a mechanical computer can be constructed that has the computational power of any conventional electronic computer and yet has only a constant number of mechanical parts.

Previous constructions for mechanical computing devices (such as Babbage’s Analytical Engine) either provided no general construction for finite state control or the control was provided by electronic devices (as was common in electro-mechanical computers such as Mark I subsequent to Turing’s result). Our result seems to be the first to provide a general proof of the simulation of a universal TM via a purely mechanical mechanism.

In addition, we discuss the universal frictional mechanical system in the context of an error model that allows an error up to ϵ in each mechanical operation. We first show that, for a universal TM M , a frictional mechanical system in this ϵ -error model can be constructed such that, given any space bound S , the system can simulate the computation of M on any input string ω if M decides ω in space bound S , provided that $\epsilon < 2^{-cS}$ for some constant c . We also show that, for any universal TM M and space bound S , there exists a frictional mechanical system in the ϵ -error model with $\epsilon = \Omega(1)$; it has $O(S)$ parts and can simulate M on any input ω that M decides in space bound S .

Key words. robotics, motion planning, complexity, mechanical computing device

AMS subject classifications. 68T40, 93C85, 03D15, 68Q15, 68Q17

1. Introduction.

1.1. Motivation. There are two contexts where our result may be of interest: robotic mover’s problems and mechanical computing machines.

1.1.1. Robotic Mover’s Problems. The objective of *robotic mover’s problems* (or *motion planning problems*) is to plan the motion of a robot between distinguished configurations under specified physical constraints (e.g., avoiding obstacles) and possibly also dynamic constraints. There are two categories of problems studied in this area: decision problems and optimization problems. The goal of a decision problem is to decide, given the physical and dynamic constraints, whether or not there is a sequence of legal movements that will allow the robot to reach the goal configuration; the goal of an optimization problem is to find an optimal path (trajectory) that leads to the goal configuration, according to a predefined cost function on paths (trajectories).

*A preliminary version of this paper appeared in *Proceedings of the 3rd Workshop on Algorithmic Foundations of Robotics* (WAFR98).

[†]Department of Computer Science, Duke University, Box 90129, Durham, NC 27708-0129, USA (reif@cs.duke.edu).

[‡]Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong (sunz@comp.hkbu.edu.hk).

The first hardness result for a robotic mover’s problem was presented by Reif [9]. He showed that the generalized mover’s problem of moving n linked polyhedra through a set of $3\mathcal{D}$ obstacles is PSPACE-hard. His proof used a reduction of the computation of any reversible Turing Machine on an input string to an instance of the mover’s problem. Hopcroft *et al.* [7] improved on this result by proving that the mover’s problem for $2\mathcal{D}$ linkages is PSPACE-hard. Later the generalized mover’s problem was proved to be in PSPACE by Canny [2]. Using a *path coding* technique, Canny and Reif [3] also proved that computing the shortest path for a point robot moving amidst polyhedral obstacles is NP-hard. Asano *et al.* [1] introduced the problem of computing the d_1 -optimal motion for a $2\mathcal{D}$ rod (defined by a directed line segment) amidst polygonal obstacles and showed that this problem is NP-hard.

There are also many hardness results on various extensions of the basic robotic mover’s problem, such as moving obstacles, multi-robot, etc. For example, in [8] Hopcroft *et al.* proved that motion planning for multiple independent rectangular boxes sliding inside a rectangular box is PSPACE-hard. A similar problem of moving multiple discs inside a polygon in a $2\mathcal{D}$ space, however, could only be proved to be strongly NP-hard [14]. Reif and Sharir [10] introduced the $3\mathcal{D}$ mover’s problem in the presence of moving obstacles and showed that this problem is PSPACE-hard even in a case where the object to be moved is a disc with bounded velocity. By extending the path coding technique of [3], Reif and Wang [12] proved that the two dimensional curvature-constrained shortest-path problem is NP-hard. Later, Reif and Sun [11] showed that the time-optimum path planning problem for a point robot in a $3\mathcal{D}$ space composed of polyhedral regions with flows is PSPACE-hard.

Most robotic mover’s problems assume that the obstacles are the only objects in robot’s workspace besides the robot itself. One exception is the *movable object problem*, which is to ask whether a robot can move certain objects amidst obstacles in a space to reach a target configuration. That is, there are two types of objects in the space: obstacles that can not be moved and penetrated (or even touched) by the robot, and objects whose placements can be changed by the robot. The goal of the robot is to either rearrange the movable objects in the space to a desired configuration or reach a target configuration of itself, or both. The first result on the movable object problem was given by Wilfong [15], who studied this problem for the case of a polygonal robot moving in translation amidst polygonal movable objects in a bounded polygonal space. He proved that, if the final configurations of the objects are not specified as part of the goal of the motion planning problem, this problem is NP-hard; otherwise, it is PSPACE-hard. He also gave one algorithm for each of the two cases where only one movable object is present.

In Wilfong’s model, a robot can only grasp an object from a finite number of positions. This problem is considered to be more difficult when this number is infinite. Chen and Hwang [4] gave a heuristic algorithm to solve one model of this problem where the total weight of objects moved by a robot is to be minimized. Dacre-Wright *et al.* [5] extended Wilfong’s work by providing an $O(n^3 \log n)$ algorithm for the infinite grasping position case where the final configurations of the objects are specified as part of the goal.

1.1.2. Mechanical Computing Machines. In 1822, Charles Babbage designed and constructed the Difference Engine. This machine was specially designed for the evaluation of polynomials. Later in 1833, Babbage proposed (but did not fully construct) a new device, Analytical Engine, which was conceived to solve general arithmetic problems. It resembled the modern digital computer in the following

ways:

Input Device Just like any modern computer, the Analytical Engine was designed to have an input mechanism, such as punch cards.

Memory The Analytical Engine was supposed to store the data encoded by mechanical positions, e.g., distinct digits were stored via rotational positions of distinct mechanical dials.

Arithmetic Unit The machine was able to manipulate the data and, in particular, to execute the arithmetic operations. This was done by a part called the “mill” using various gearing mechanisms.

Control Unit The machine was also envisioned by Babbage to have a mechanism that could control the sequence of operations to carry out the computations.

About half a century after Babbage’s death, Dr. Vannevar Bush resumed the work of building a mechanical computing device. In 1925 he, along with some associates, made a mechanical calculator powered by electric motor. His machine was an analog one, in the sense that arithmetic operations were carried out by mechanical means and in terms of physical measurements.

In 1939, Howard Aiken, in collaboration with four I.B.M. engineers, built a general purpose computing machine, the so called Automatic Sequence Controlled Calculator, Mark I. Just like Babbage’s Analytical Engine, Mark I performed computation by manipulating mechanical devices. The key difference between Mark I and the previous mechanical computers such as Analytical Engine and Bush’s machine is that, while the Analytical Engine and Bush’s computer were purely mechanical, the operations of mechanical parts of Mark I were controlled electrically.

As electronic devices were not available in 1800’s, Babbage had to exploit a purely mechanical system to build a computer. Subsequent electro-mechanical computers could exploit electronics for control, and of course so do the modern computers. Today, Babbage’s concept of a purely mechanical computer would at first seem to be out of date, as computers built by much faster electronic technology prevail in every corner of the world. However, the emergence of nanotechnology provides new motivation on studies of mechanical computers.

1.2. Our Results.

1.2.1. Frictional Mechanical System and Frictional Mover’s Problem.

All the robotic mover’s problems mentioned above assume that there is no friction between objects, and most of the models only allow collision-free movements so that different objects cannot even contact with each other. The only work that addressed on motion planning in the presence of friction is by Sellen [13]. He proved that dynamic motion planning problem with forbidden movements (in particular, sliding) is undecidable by showing that the actions of a TM can be realized by logical and arithmetic operations, which can be implemented by mechanical means. However, in his model, the motions of the objects corresponding to the computations of the TM can not be generated deterministically. Therefore, this model can not be used for constructing a mechanical computer.

We define a *frictional mechanical system* to be a collection of rigid objects in $3D$ space whose surfaces are composed of linear or quadratic surface patches specified by rational coefficients. All objects are non-penetrable, i.e., the only allowed intersection is via surface contact. Each surface patch of each object is also specified as either *frictional* or *sliding* (non-frictional). If two objects with frictional surfaces contact

with each other, it is assumed that there is no sliding¹ between them. If at least one of the two contacting surfaces is designated to be a sliding surface, there will be no friction between them so that they can slide freely. Furthermore, some objects in the space are specified to move monotonically. In particular, there may be some discs in the system that can only rotate in one direction (clockwise or counterclockwise). We say that the objects can be moved *legally* if all the above constraints are satisfied.

We define the *resource bound*, denoted by R , to be the number of distinct objects in the frictional mechanical system. As each object can be specified by a constant number of surface patches, each of which can be specified by a constant number of rational coefficients, the total number of binary bits used to encode an object is bounded by a constant. Hence, the total number of bits in the binary representation of the frictional mechanical system is $O(R)$.

The *frictional mover's problem* is to determine whether these objects in the frictional mechanical system can be moved legally from a specified initial configuration to a specified final configuration. This problem can be regarded as a generalization of the movable object problem. Compared to the previous works, our model is in $3D$ space and the surfaces of the objects in the space can be non-linear. Further, in addition to moving an object by grasping or pushing (directly or indirectly), a robot in our model can move objects by using the friction between it and surrounding objects. More specifically, a *power disc* in the frictional mechanical system can be deemed as a rather “dumb” robot; it is restricted to rotate in a specified direction without translation. And the problem is to ask whether this robot can rearrange the objects in the system to a target configuration by its rotation.

We prove that the frictional mover's problem is undecidable by reducing the acceptance problem for Turing Machine² (TM), A_{TM} ³, to the frictional mover's problem. Given a universal TM⁴ M , we construct a frictional mechanical system to simulate this machine. This frictional mechanical system will have the property that the objects in this system can be moved from an initial configuration, which encodes an input string ω of M , to a configuration corresponding to the accepting state of M if and only if M accepts ω . Therefore, as the acceptance problem for Turing Machine is undecidable, so is the frictional mover's problem. This implies that there is no realistic machine that can solve this problem.

An interesting property of this frictional mechanical system is that, if M accepts ω , there will be a unique simple path (i.e., one that does not repeat the same configuration) from the initial configuration to the final configuration.

The proof will actually construct, for any given TM M , a frictional mechanical system that simulates M . Every movable object in the system is engaged or linked directly or indirectly with the power disc so that, when the power disc rotates, it will make those objects move accordingly. For any input string ω of M , this frictional mechanical system can be set to an initial configuration encoding ω so that, after the power disc has rotated a sufficient number of cycles, this system will result in a

¹Sliding is a move in a direction tangent to the surfaces at the contact point.

²A Turing Machine is an abstract machine with a finite state control and a tape that can store an infinite string of symbols. There is a read-write head on the tape that allows the machine to read or write the symbol at the current position of the head. The machine can write to the current position and move the head left or right according to the current state and the current symbol in a specified way.

³The acceptance problem A_{TM} for Turing Machine is to determine, given the description of a TM M and its input ω , whether M accepts ω .

⁴A universal TM M will take the description of any TM M' and any input string ω of M' as an input and simulate the behavior of M' presented with input ω .

configuration encoding the accepting state (rejecting state) of M , if and only if M accepts (rejects) ω .

Even though Babbage claimed that the Analytical Machine could be used to solve any arithmetic problem, it is doubtful whether this machine is as powerful as a universal computing machine. The difficulty is that Babbage did not have a general concept of an abstract computing machine, such as a Turing Machine (TM), which was later introduced by Alan Turing in 1936. To show that a mechanical system has the computational power of an electronic computer, it is sufficient and necessary to show that this mechanical system can simulate a universal TM, as proposed by Turing. Although Babbage designed his Analytical Machine to have a control unit used to guide the arithmetic operations, it was not explicitly shown by Babbage how it could simulate a general finite state control, which is the core of a universal TM. The electro-mechanical computer, Mark I, was designed to be capable of fulfilling any computing tasks and thus should be as powerful as the modern computers. However, it used an electronic device as the central controller. To our knowledge, our mechanical system, as it will be described in later sections, is the first mechanical system that can perform general-purpose computation without using any electronic devices.

Another limitation of Babbage's Analytical Engine is its representation of numbers. The Analytical Engine was a digital computer. Each digit of a number was represented by a mechanical device, such as a dial. The accuracy of Babbage's machine, in terms of number of decimal places, was determined by the number of dials used to represent a number. Therefore, no matter how precisely the machine was built, it would not improve its accuracy. The only way to improve the accuracy of the machine's computation (or increase the number of data bits) is to increase the number of dials used to represent a number. This will change the structure of the Analytical Engine and make it bigger and more complicated. Our frictional mechanical system, however, is an analog computer, in the sense that a number (in fact, the entire data of the computation) is represented by the rotational position of a single disc. The accuracy of our system depends on the accuracy of the measurement and the precision of the mechanical devices. Therefore, it can be arbitrarily accurate as we reduce the error in transitional (or rotational) measurement as well as the error in building mechanical devices.

For our proof of the undecidability of the frictional mover's problem, we adopt a simple deterministic model for frictional contacts between objects. Note that there exists a number of considerably more complex models (for example, see [6]) for frictional contacts between objects where the objects in contact with surfaces may make non-deterministic motions. However, our simple deterministic model for frictional contacts will suffice for us to adequately model frictional contacts in the simple cases we employ in our constructions, and so to prove our undecidability result for movement planning with frictional contacts. Moreover, many of these more complex models for frictional contacts reduce to our simple deterministic model for frictional contacts in the simple cases we employ in our constructions.

1.2.2. Frictional Mechanical System with Error. We prove that a frictional mechanical system can be constructed to simulate a universal TM. Therefore, this system can be used for arbitrary finite computation, just like any conventional computer. However, the underlying assumption is that this frictional mechanical system can be constructed exactly as it is specified. We are also interested in the computational power of such a frictional mechanical system in the case inaccuracy is allowed in the construction of the mechanical devices in the system.

There are many factors that might induce errors in the computations of a frictional mechanical system, including the precision of manufacture of the parts. For example, the circumference of a disc may not exactly be manufactured to be a circle. The radius of a disc may not be manufactured to be exactly as it is specified. When two discs are very close but still not in contact with each other theoretically, they may already have surface contact so that the rotation of one disc will move the other one, even though they are not supposed to do so.

Since there are constant number of mechanical devices in our mechanical system, we can let ϵ to be the upper bound for the errors that occur in a single operation. This is our ϵ -error model. We prove that, given a space bound S , our frictional mechanical system in this ϵ -model can simulate the universal TM M on any input string ω that can be decided by M in space bound S , provided that $\epsilon = O(2^{-cS})$ for some constant c .

We also prove that, given a universal TM M with space bound S , there exists an $\epsilon = \Omega(1)$ such that a frictional mechanical system in ϵ -error model can simulate the computation of M presented with any input ω if M decides ω in space bound S . This result provides decreased required precision of parts at the expense of increased number of parts, which increases with S .

1.3. Notations. In the following sections, the universal TM with end-marks is denoted by quintuple $M = (Q, \Sigma, \delta, q_0, \{q_{\sigma-2}, q_{\sigma-1}\})$ as follows:

- 1 $Q = \{q_0, q_1, \dots, q_{\sigma-1}\}$ is the set of states.
- 2 $\Sigma = \{0, 1, \dots, m-1\}$ is the tape alphabet. Here 0 denotes the *blank* symbol. 1 and 2 are the left and right end-marks respectively.
- 3 $\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R\}$ is the transition function.
- 4 q_0 is the start state.
- 5 $\{q_{\sigma-2}, q_{\sigma-1}\}$ is the set of halting states. In particular, $q_{\sigma-2}$ is the rejecting state and $q_{\sigma-1}$ is the accepting state.

In our following discussion, m and σ are considered as constants.

At anytime during the computation, we use the *current working space* of the TM to denote the portion of the tape that the read-write head has visited so far. The current working space is always bounded by a left end-mark and a right end-mark. We denote the tape status by a string $\omega_1 \omega_2 \dots \omega_{k_1-1} \check{\omega}_{k_1} \omega_{k_1+1} \dots \omega_{k_2}$. Here ω_1 and ω_{k_2} are the left and right end-mark respectively, and $\check{\omega}_{k_1}$ denotes that read-write head is at the k_1 -th cell of the tape. The read-write head will never replace the left end-mark by another symbol nor will it go beyond the left end-mark. Whenever the read-write head replace the right end-mark by another symbol, it will pad a right end-mark to the right of the symbol.

It is easy to see that, for any TM (without end-marks) M' , there is a equivalent TM with end-marks as described above. Therefore, all universal TMs mentioned in the following discussion are assumed to have end-marks.

Since discs, in particular, the rotational positions of discs, play a very important role in our frictional mechanical system, we want to specify several terminologies which we will be using frequently in the following discussion.

Each disc used in our system has a specified orientation called the *initial orientation*. Therefore, the rotational position of a disc D can be specified by the angle the disc has rotated from its initial orientation. For our convenience, we will use "the angle of D " to denote this angle. Also, we will say that the angle recorded (or represented) by the rotational position of disc D is θ . Further, if we say "to increase (decrease) the angle or rotational position of a disc D by an angle of θ ," we mean to

rotate D counterclockwise (clockwise, respectively) by an angle of θ .

A *partial disc* is a portion of a disc bounded by two radii and the remaining portion of the circumference. As we will show later, partial discs are very useful in our system, too.

1.4. Organization of this Paper. Section 2 presents several basic mechanical devices widely used in our frictional mechanical system. After that, we provide the full description of the frictional mechanical system that simulates a universal Turing Machine. In the last section, the model of the frictional mechanical system with errors is discussed.

2. Basic Gadgets. We prove the reduction from A_{TM} to the frictional mover's problem by constructing a frictional mechanical system for a universal TM M . This system is composed of discs, partial discs, cylinders, and other geometric objects. There is a special disc, the power disc, which is specified to only rotate clockwise. The rotation of this disc will force other objects in the system to move, due to frictional linkages between them. This frictional mechanical system can “simulate” M in the sense that, for any input string ω of M , this system can be set to a corresponding initial configuration so that a distinguished final configuration can be reached if and only if M accepts ω . Therefore, if one can decide this instance of frictional mover's problem, he can also decide the corresponding instance of A_{TM} . As it is well known that A_{TM} is undecidable, so must be the frictional mover's problem.

Friction is very important in this system as it only provides method of moving other objects in the system but also preserves the state of the system and thus the state of the Turing Machine the system simulates. It also guarantees the system properly transfers from one state to another.

Before giving the construction of the frictional mechanical system, we first introduce in this section several “gadgets” that perform the basic functionalities and thus are used widely in our system.

There are seven kinds of basic gadgets:

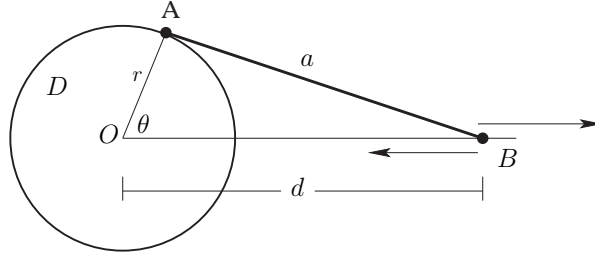
Converting device A converting device is used to convert between rotational displacement and transitional displacement.

Sequencer A sequencer is composed of a group of discs and partial discs that share the same axis and rotate with same angular speed at any time. These discs (and partial discs) are engaged with some surrounding discs in the frictional mechanical system so that when rotating, they will make the surrounding discs rotate with them and in turn move other objects in the system. The sequencer is a key component in the frictional mechanical system as it provides a mechanism for moving all the objects in the system in a specified way.

Transitional Movement Sequence Controller A transitional event sequencer controller allows the sequencer to periodically move an object in the system by a constant distance and then move it back.

Rotational Movement Sequence Controller A rotational event sequencer controller allows the sequencer to periodically rotate a disc by a constant angle.

Resettable Rotational Disc A resettable rotational disc has two states: the free rotation state when it can be rotated by engaging with another disc; and the reset state when it will return to a specified initial orientation no matter how much it is rotated in the free rotation period. Virtually all devices need to use resettable rotational discs.

FIG. 2.1. *Converting Device*

Non-linear Mapping Controller A non-linear mapping device is used to implement a non-linear mapping between rotations. This is a very crucial component in the construction of the simulation of a finite state control.

Selection Controller A selection controller helps the sequencer to initiate one among several event sequences depending on the rotational (or transitional) position of an object in the system. This is yet another key component in the construction of the simulation of a finite state control.

In the following subsections, we will give detailed description for each of the basic gadgets.

2.1. Converting Device. The most basic device is one that can convert a rotational displacement to a transitional displacement, or vice versa. This can be done by the device described in Figure 2.1.

As it can be seen from the figure, one end A of an arm of length a is attached to a fixed position on the rim of the disc D . The other end B of this arm is restricted to move along a line l passing the center of the disc. The rotation of the disc D , θ , and the distance d between the center of the disc and B satisfy $d = r \cos \theta + \sqrt{a^2 - r^2 \sin^2 \theta}$. Here r is the radius of D .

Therefore, if θ varies between $\pi/4$ and $3\pi/4$, d will vary between $\sqrt{a^2 - r^2/2} - r/\sqrt{2}$ and $\sqrt{a^2 - r^2/2} + r/\sqrt{2}$. Further, d is a monotone function of θ . Therefore, the rotational displacement of disc D can be converted to the transitional displacement of B and vice versa.

2.2. Sequencer. All the objects in the system are moved, directly or indirectly, by a mechanism called sequencer, which is composed of a group of discs and partial discs. These discs and partial discs share the same axis and rotate with same the angular speed at anytime. Each disc or partial disc is engaged with one or more surrounding discs. Therefore, when the sequencer is rotating, it will make these surrounding discs rotate. Also, the rotation of these surrounding discs will then move other objects in the system.

In Figure 2.2, there are three discs and two partial discs in the sequencer. Each of the discs of the sequencer will make its surrounding disc rotate with it, possibly at a different angular speed. Each of the partial discs, however, will move the surrounding disc for a certain period, and then lose contact with it.

Each cycle of the sequencer finishes one step of computation of the universal TM it simulates. As the rotation of the sequencer determines the starting and finishing of any movement sequence of any other object linked with it, we may use an interval $[\theta_1, \theta_2]$ to denote the time period during which the sequencer rotates from angle θ_1 to angle θ_2 in each cycle rather than using the actual time.

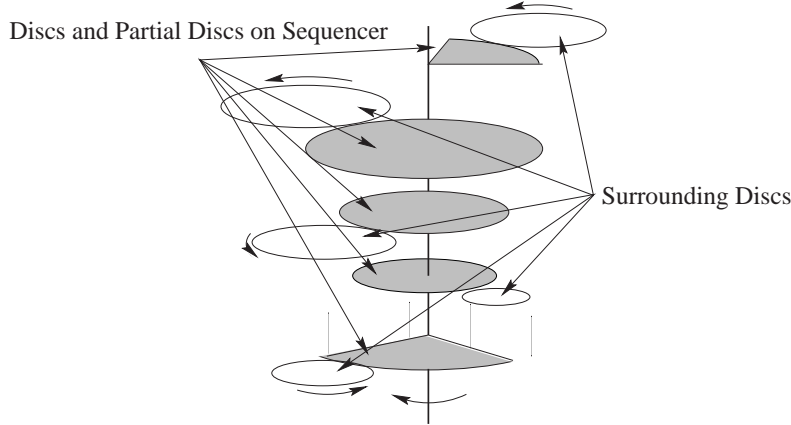


FIG. 2.2. *Sequencer*

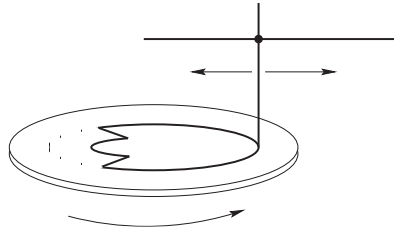


FIG. 2.3. *Transitional Movement Sequence Controller*

2.3. Transitional Movement Sequence Controller. It will be useful if the sequencer can move an object to a specified position during time period $[\theta_1, \theta_2]$ of each rotational cycle, and move it away from that position otherwise. This can be done by the controller shown in Figure 2.3.

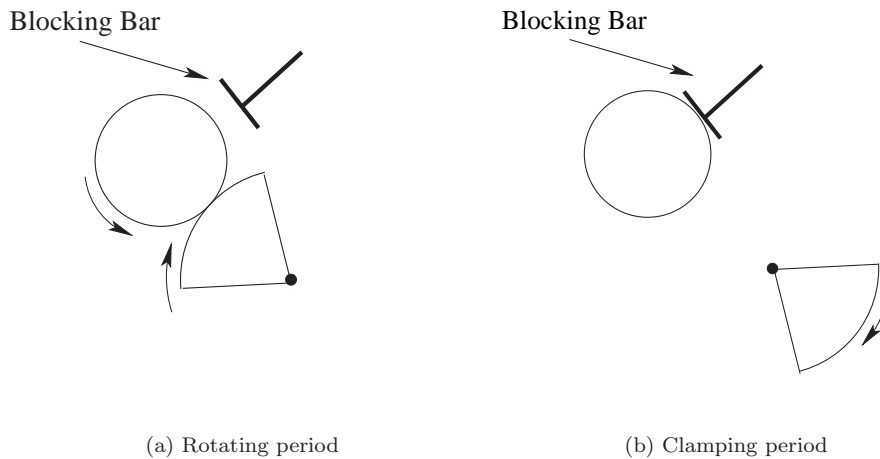
There is a circular gap on the surface of the disc which is on the sequencer. The curve of this gap is defined by $r = r(\theta)$ in a polar coordinate with origin at the center of the disc. A vertical bar is restricted in the circular gap. Also, it is restricted to move on a line passing the center of the disc. $r(\theta)$ has the property that

$$\begin{aligned} r(\theta) &= r_0 && \text{for } \theta \in [\theta_1, \theta_2], \\ r(\theta) &> r_0 && \text{for } \theta \notin [\theta_1, \theta_2]. \end{aligned}$$

Here r_0 is a constant. Therefore, when the controller is rotating in one direction, the bar will move back and forth along the line. More specifically, in time interval $[\theta_1, \theta_2]$, the vertical bar is at the position of $(r_0, 0)$ (in polar coordinate). In other time, it is at some position $(r', 0)$ where $r' > r_0$.

For our convenience, in the following discussion, we may say “the sequencer moves an object to a certain position during a certain time period in each rotational cycle” implicitly assuming that we have constructed a transitional movement sequence controller, as described here, that fulfills this task.

Transitional movement sequence controllers will be used frequently in our frictional system. All the rotational discs, except those on the sequencer, need to be clamped down whenever they are not rotated, directly or indirectly, by the sequencer

FIG. 2.4. *Rotational Movement Sequence Controller*

so that their orientations will be maintained. This can be done by moving a *blocking bar* (a bar with a frictional surface) to a position where it has surface contact with the disc in the clamping period of each cycle.

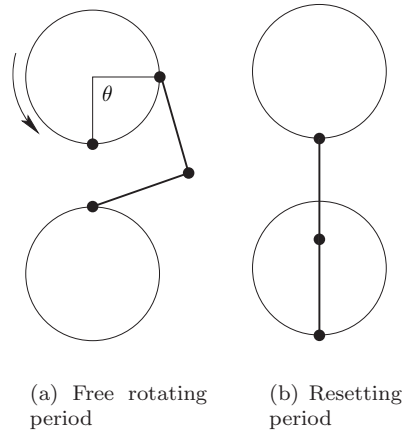
2.4. Rotational Movement Sequence Controller. A rotational movement sequence controller is a partial disc in the sequencer used to rotate another disc, called *target disc*, in a specified time interval during each cycle. With a transitional movement sequence controller, a blocking bar can be moved to clamp the target disc when the partial disc loses contact with the target disc. When the partial disc rotates to such an orientation that its rim contacts the rim of the target disc, this bar is moved away from the target disc so that this disc can be rotated by the partial disc.

Therefore, the disc will stay stationary during the clamping period, and rotate with the partial disc otherwise. The length of the rotating period is determined by the angle of the partial disc. And the total angle the target disc rotates during the rotating period is determined by the length of the arc of the partial disc as well as the radius of the target disc. Therefore, by setting these parameters appropriately, we can let the target disc to rotate a certain angle, say, π in each cycle.

Again, for our convenience, in the following discussion, we will say “the sequencer rotates a disc by a specified angle during a certain time period” implicitly assuming that we have constructed a rotational movement sequence controller, as described in this subsection, to do it.

2.5. Resettable Rotational Disc. With a rotational movement sequence controller, the sequencer can rotate a disc by a constant angle during each cycle. Frequently it is necessary to rotate a target disc by a certain angle θ recorded by a *source disc* and then reset it by rotating it by θ in the reverse direction, no matter what θ is. This kind of disc is called *resettable rotational disc*. As shown in the figure, two discs are linked by a two-segment arm. Each of these two discs has a radius of 1. The distance between the centers of the two discs is d , and the total length of the two-segment arm is d too.

The upper disc is the target disc. The lower disc connects to a rotational movement sequence controller, so that at the time when we want to reset the target disc,

FIG. 2.5. *Resettable Rotational Disc*

the lower disc will rotate by an angle of π . Because the length of the two-segment is exactly the same as the distance between the centers of two discs, the upper disc will be forced to rotate to such an orientation that the point where the arm is attached to is at the lowest position. When it is the time to release the target disc and allow it to be rotated freely by another disc it engages with, the lower disc rotates by another π so that it restores to its initial orientation.

Therefore, each resettable rotational disc can be regarded as a register. It can hold a value that is represented by its current angle θ .

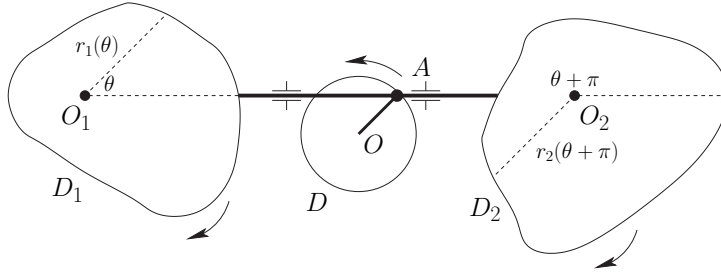
Using resettable rotational discs, we can construct a mechanical device that will increase the angle of a disc by the angle recorded by another disc. Suppose we have two resettable rotational discs. Disc D_1 is set at θ_1 and D_2 is set at θ_2 . We want to increase the angle of D_2 by θ_1 while having the angle of D_1 remain unchanged. A third disc D_3 is needed here and it is set at its initial orientation. The radii of D_1 , D_2 and D_3 are the same.

First both D_2 and D_3 are engaged with D_1 and then D_1 is reset to its initial orientation. D_2 (and D_3) will be rotated counterclockwise by an angle of θ_1 . This will set D_2 to angle $\theta_1 + \theta_2$ and D_3 to angle θ_1 . Then, D_2 is disengaged from D_1 . The next step is to reset D_3 to its initial orientation. As D_3 is still engaged with D_1 , D_1 will be rotated counterclockwise by an angle of θ_1 . Therefore, the orientations of D_1 and D_3 will be set just the same as when they were before, while the angle of D_2 is increased by θ_1 , which is the angle recorded by D_1 .

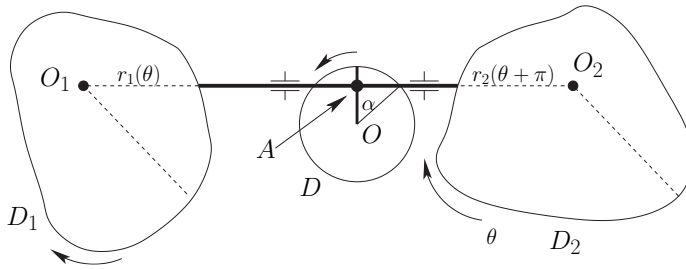
Observe that we can amplify or reduce θ_1 by a constant factor C before adding it to D_2 . This can be done by setting $r_1 = r_3 = Cr_2$, where r_1 , r_2 and r_3 are the radii of the three discs respectively.

Here we will have a digression to address the necessity of using discs with frictional linkages. If gears instead of discs were used, it might be possible that teeth of the gears could not match after they resume contact with each other, since the two gears might rotate by a different angle after they are disengaged.

2.6. A Non-linear Mapping Device. Engaging a source disc and a target disc will allow us to perform linear mapping from the angle of rotation of the source disc to the angle of rotation of the target disc. By “linear mapping,” we mean that if



(a) Initial position

(b) After rotating by a degree of θ FIG. 2.6. *Non-linear Mapping Device*

the source disc rotates by an angle of θ , the target disc will rotate by an angle of $C \cdot \theta$, where C is the radius ratio between the two discs. However, sometime it is necessary to perform non-linear (or even non-monotone) mapping on the rotations of discs. In particular, a device with the following property is needed:

PROPERTY 1. *A non-linear mapping device has two discs, an input disc and an output disc, each of which can rotate by any angle $\theta \in [0, \pi/2]$ from its initial orientation. Given $\theta_1, \theta_2, \dots, \theta_n$, $\alpha_1, \alpha_2, \dots, \alpha_n$, $0 \leq \theta_1 < \theta_2 < \dots < \theta_n \leq \pi/2$, $0 \leq \alpha_i \leq \pi/2$, if the rotation of the input disc is θ_i , the rotation of the output disc should be α_i . Further, if $\alpha_1 < \alpha_2 < \dots < \alpha_n$, the mapping should be monotone. That is, for any θ , $\theta_i < \theta < \theta_{i+1}$, if the rotation of the input disc is θ , the rotation of the output disc, α , should be in the range of (α_i, α_{i+1}) .*

Non-linear mapping can be implemented by the device described in Figure 2.6. It consists of four parts, a central (regular) disc D centered at point O , a horizontal bar, and two irregular discs D_1 and D_2 centered at O_1 and O_2 respectively. In each operation D_1 and D_2 will rotate clockwise by an angle of θ_i for some $i, 1 \leq i \leq n$. The motion of D_1 and D_2 will cause D , through the horizontal bar connecting them, to rotate by an angle of α_i from its initial orientation.

The horizontal bar, which lies on line $\overline{O_1 O_2}$ between D_1 and D_2 , is restricted to move horizontally only. Each end of the horizontal bar touches, but is not attached to, the rim (circumference) of one of the two irregular discs. The bar is connected to D through a joint A between the bar and a spoke (a radiating bar from the center of a disc to its circumference) of D . The joint has the property that it can move along

the spoke but will remain stationary with respect to the bar. Therefore, when D_1 and D_2 rotate, they will in turn push the bar back and forth horizontally, which in turn will move the spoke and thus make D rotate accordingly.

During each operation, D_1 and D_2 need to “complement” each other in the sense that they keep constant contact with the horizontal bar. Nor can the two irregular disc “squeeze” the bar, as we assume that all objects are rigid and cannot be compressed. Therefore, we want to design the shapes of D_1 and D_2 in such a way that, when D_1 and D_2 rotate clockwise with the same angular speed, the distance between D_1 and D_2 on line $\overline{O_1O_2}$ is a constant. This implies that $r_1(\theta) + r_2(\theta + \pi)$ is a constant for any $\theta \in [0, \pi/2]$, where for each $j = 1, 2$ $r_j(\theta)$ is the distance between O_j and the intersection point of the rim of D_j and the ray with angle θ that starts from O_j .

Initially, D is set at the orientation such that the angle between the spoke of D and the x-axis is $\pi/4$, as shown in Figure 2.6.a. We choose the vertical distance between O and the horizontal bar to be $R/\sqrt{2}$, where R is the radius of D . Therefore, the range of the angle between the spoke and the x-axis is $[\pi/4, 3\pi/4]$. Let h_1 be the horizontal distance between O and O_1 , and let d_1 be the length of the part of the horizontal bar between its left end and the joint A with the spoke. To achieve the desired property, it is sufficient that the following equality holds:

$$(2.1) \quad \frac{r_1(\theta_i) + d_1 - h_1}{R/\sqrt{2}} = \tan(\pi/4 - \alpha_i), \text{ for } i = 1, 2, \dots, n$$

This can be done by choosing $r_1(\theta_i)$ to be $h_1 - d_1 + R \tan(\pi/4 - \alpha_i)/\sqrt{2}$, for all $i, 1 \leq i \leq n$. Also, the initial orientation of D implies that $\alpha = 0$ if $\theta = 0$. Therefore, we have $r_1(0) = h_1 - d_1 + R/\sqrt{2}$. For our convenience, we let $\theta_0 = 0$ and $\alpha_0 = 0$.

To ensure that the mapping is monotone in interval $[\theta_i, \theta_{i+1}]$, for any $\theta \in (\theta_i, \theta_{i+1})$, we specify

$$(2.2) \quad r_1(\theta) = \frac{(\theta - \theta_i) \cdot r_1(\theta_{i+1}) + (\theta_{i+1} - \theta) \cdot r_1(\theta_i)}{\theta_{i+1} - \theta_i}.$$

Note that $r_1(\theta)$ is a linear function of θ inside interval (θ_i, θ_{i+1}) . If $\alpha_1 < \alpha_2 < \dots < \alpha_n$, the non-linear mapping will be monotone in interval $[0, \theta_n]$.

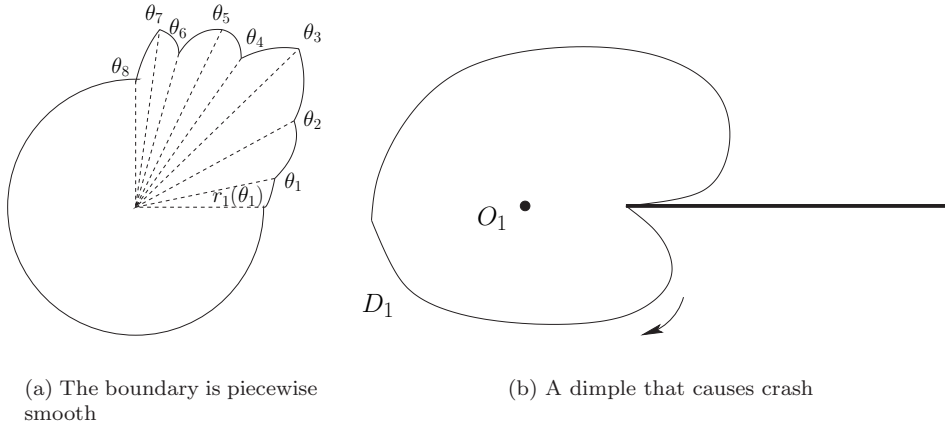
In the above we have shown how to decide $r_1(\theta)$ for any $\theta \in [0, \theta_n]$. For any $\theta \in (\theta_n, 2\pi)$, we let

$$r_1(\theta) = \frac{(\theta - \theta_n) \cdot r_1(0) + (2\pi - \theta) \cdot r_1(\theta_n)}{2\pi - \theta_n}.$$

This completes the specification of D_1 . For D_2 , we only need to let $r_2(\theta) = |\overline{O_1O_2}| - d - r_1(\theta - \pi)$, so that $r_1(\theta) + r_2(\theta + \pi)$ is a constant for any $\theta \in [0, 2\pi)$. Here d is the length of the horizontal bar.

As shown in Figure 2.7.a, the boundary of each irregular disc D_j described above is piecewise smooth, with derivative $r'_j(\theta)$ defined everywhere except for $\theta = \theta_0, \theta_1, \dots, \theta_n$. At these values function $r_j(\theta)$ has a “left derivative” $r'_j(\theta^-)$ and a “right derivative” $r'_j(\theta^+)$, which in most cases are different. The difference between $r'_j(\theta^-)$ and $r'_j(\theta^+)$ would not hinder the proper functioning of the device.

The only concern is that there should not be a “dimple” on the boundary, where $r'_j(\theta^+) = +\infty$ for some $\theta \in [0, 2\pi)$. In this case, the irregular disc would not be able to push the horizontal bar away from it, and, even worse, the bar would actually prohibit the irregular disc from further rotating, as shown in Figure 2.7.b. This would lead to a

FIG. 2.7. *Irregular Discs*

complete stop of the computation of the frictional mechanical system. However, since inside each interval $[\theta_i, \theta_{i+1}]$, the curve $r_j(\theta)$ is a linear function of θ (see Equation 2.2), $r_j'(\theta^+)$ cannot be $+\infty$ as long as each $r_j(\theta_i)$ is chosen to be a finite number. Therefore, we have ruled out this potential difficulty.

Observe that even though the above mapping device can only implement a mapping from $[0, \pi/2]$ to $[0, \pi/2]$, with a linear mapping device that can amplify or reduce angle, for any non-linear mapping, we can construct a device to simulate it.

2.7. Selection Controller. Another useful device is the so-called *selection controller*, which will rotate one of several cylinders depending on the transitional displacement of a bar from its initial position or, equivalently, the rotational displacement of a disc from its initial orientation. Therefore, different position of the bar will incur different movement of the objects in the space.

Suppose there are n cylinders, called *choice cylinders*, from which one is chosen to be rotated depending on the transitional displacement of a sliding bar. Let l denote the transitional displacement of the sliding bar. We assume that $0 < l < 1$. Further, if $2i/(2n-1) < l < (2i+1)/(2n-1)$ for some i , $0 \leq i \leq n-1$, the selection controller will select the i -th choice cylinder to rotate. This can be implemented by the device in the Figure 2.8.

To the left there is a long cylinder S engaged with a rotational movement sequence controller so that S is rotated by an angle of 2π during a certain period of each cycle and remains static in other time. To the right there is a pile of n identical cylinders, D_0, D_1, \dots, D_{n-1} . These cylinders share the same axis but each can rotate independently. The height of each cylinder is $1/(2n-1)$ and the distance between two contiguous cylinders is also $1/(2n-1)$. The radii of D_0, D_1, \dots, D_{n-1} and D are all r . The distance between D 's axis and D_i 's axis is $2r + 2R$.

The sliding bar is attached to what we call a *selective engager* shown in the middle of Figure 2.8. The selective engager consists of a disc D' with radius R and $2n-2$ blocking bars, each of which has length R . $n-1$ of these blocking bars are above D' and the others are below D' . The vertical distance between any two contiguous bars is $2/(2n-1)$, and so are the distance between D' and the bar immediately above D' as well as the distance between D' and the bar immediately below D' .

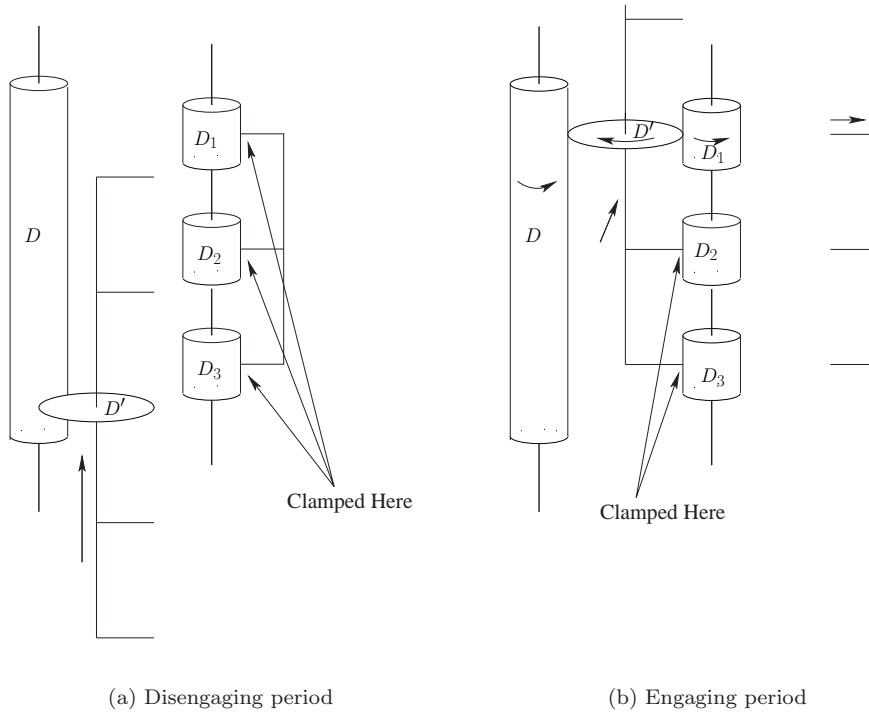


FIG. 2.8. Selection Controller

In each cycle, initially the selective engager is placed in such a position that D' is at the same height as the lower face of D_1 . Also, the selective engager does not contact either D or any of D_i s. After the sliding bar moves vertically by a distance of l , $2i/(2n - 1) < l < (2i + 1)/(2n - 1)$ for some i , the selective engager will move upwards by a distance of l accordingly so that it will be at the same height as D_i .

Then, the selective engager is moved horizontally to the position right between D and D_i s. Since the distance between D 's axis and D_i s' axis is $2r + 2R$, D' will contact both D and D_i . Also, any of D_0, D_1, \dots, D_{n-1} other than D_i will contact one of those blocking bars so that they are forbidden from rotating.

After removing the blocking bars originally clamping the n cylinders, only n_i can rotate freely. Therefore, when D is rotated by an angle of 2π , it will rotate D' and in turn rotates D_i . Since D and D_i have the same radius, D_i will be rotated by exactly 2π .

The functionality of this device is just like the **switch—case** structure in a C program. In our remaining section, we will say “If the transitional (rotational) displacement of a bar (disc, respectively) is in the range of (a, b) , the sequencer will do ...” implicitly assuming that we have constructed a particular selection controller as described here to do this task.

3. Simulation of a Universal Turing Machine. Given the basic gadgets described above, we are ready to construct a frictional mechanical system that simulates a universal TM. Our frictional mechanical system consists of two components, a *control component* simulating the finite state control and a *tape component* simulating

the read-write tape. All the objects in the system are linked or engaged with a sequencer, which controls the behavior of those objects. The sequencer is engaged with the power disc that is constantly rotating clockwise. Therefore, the sequencer will be rotating counterclockwise constantly and it will drive all the objects in the system in a specified way as it rotates.

In the control component, there is a disc D_{state} whose rotational position represents the current state of the universal TM. Similarly, in the tape component, there are three discs $D_{current}$, D_{left} and D_{right} whose rotational positions together represent the status of the read-write tape. (The status of the tape includes the contents of the tape as well as the current position of the read-write head.)

The control component will take the current symbol and the current state of the universal TM, both of which are represented by rotational positions of certain discs, as input and compute the next state, the next move of the head, and the symbol replacing the current symbol, all of which are also represented by rotational positions. The tape component will then modify the rotational positions of $D_{current}$, D_{left} and D_{right} depending on the next move and the replacing symbol provided by the control component.

The first subsection presents the methods used to encode the current state of the universal TM and the current status of the read-write tape by rotational positions. The next two subsections describe the construction of the control component and the tape component respectively. The last subsection shows how these two components work together to simulate a universal TM and subsequently how the two main theorems of this paper are proved.

3.1. Encoding the Configuration of the Universal Turing Machine. The configuration of a TM includes the current state, the current tape contents and the current head location. To be able to simulate a TM, our frictional mechanical system should be able to record the configuration of the TM by rotational positions of discs in the system.

The status of the read-write tape of the universal TM can be encoded by the rotational positions of three discs $D_{current}$, D_{left} and D_{right} . If $\omega_1\omega_2\cdots\omega_{k_1-1}\omega_{k_1}\omega_{k_1+1}\cdots\omega_{k_2}$ is the current tape status, the rotation of $D_{current}$, $\beta_{current}$, the rotation of D_{left} , β_{left} , and the rotation of D_{right} , β_{right} , are set as follows:

$$(3.1) \quad \beta_{current}(\omega_1\omega_2\cdots\omega_{k_1}\cdots\omega_{k_2}) = \frac{2\omega_{k_1}\pi}{2m+1}$$

$$(3.2) \quad \beta_{left}(\omega_1\omega_2\cdots\omega_{k_1}\cdots\omega_{k_2}) = 2\pi \left(\sum_{i=1}^{k_1-1} \frac{\omega_{k_1-i}}{(2m+1)^i} \right)$$

$$(3.3) \quad \beta_{right}(\omega_1\omega_2\cdots\omega_{k_1}\cdots\omega_{k_2}) = 2\pi \left(\sum_{i=1}^{k_2-k_1} \frac{\omega_{i+k_1}}{(2m+1)^i} \right)$$

In particular, $\beta_{left} = 0$ if $k_1 = 1$, and $\beta_{right} = 0$ if $k_1 = k_2$.

Therefore, $\beta_{current}$ encodes the current symbol, and D_{left} (D_{right} , respectively) encodes the sub-string to the left (right, respectively) of the current symbol.

It is easy to see that this encoding function has the following properties: (i) $(2i-1)\pi/(2m+1) < \beta_{current} < (2i+1)\pi/(2m+1)$, if $\omega_{k_1} = i$; (ii) $(2i-1)\pi/(2m+1) < \beta_{left} < (2i+1)\pi/(2m+1)$, if $\omega_{k_1-1} = i$; and (iii) $(2i-1)\pi/(2m+1) < \beta_{right} < (2i+1)\pi/(2m+1)$, if $\omega_{k_1+1} = i$.

Therefore, linking a selection controller with each of $D_{current}$, D_{left} and D_{right} will allow the frictional mechanical system to decide the current symbol of the tape and the two symbols next to it, which will be used by the control component and the tape component.

During the simulation of the universal TM, if the current state of the universal TM is q_i , D_{state} will be set at a rotational position of $\lambda_i = i\pi/(2\sigma)$. Observe that every rotational position of D_{state} that represents a state is between 0 and $\pi/2$. Again, linking a selection controller with D_{state} will allow the frictional mechanical system to decide the current state of the tape.

3.2. Finite State Control Component. We let $\delta_1 : Q \times \Sigma \rightarrow Q$, $\delta_2 : Q \times \Sigma \rightarrow \Sigma$ and $\delta_3 : Q \times \Sigma \rightarrow \{L, R\}$ denote the three components of the transition function δ respectively. More specifically, if the current state is q_j and the current symbol is i , the next state is $\delta_1(q_j, i)$, the symbol to replace the current symbol is $\delta_2(q_j, i)$ and the next move of the head is $\delta_3(q_j, i)$.

The control component consists of $3m$ non-linear mapping devices. The first m of these non-linear mapping devices, $M_{1,0}, M_{1,1}, \dots, M_{1,m-1}$, are used to implement δ_1 , the next state function. For each i , $0 \leq i \leq m-1$, $M_{1,i}$ is designed in such a way that, if the input disc of $M_{1,i}$ is rotated by an angle of $\lambda_j = j\pi/(2\sigma)$ from its initial orientation, the output disc will be rotated by an angle of $\lambda_{j'} = j'\pi/(2\sigma)$. Here $\delta_1(q_j, i) = q_{j'}$. In other words, if the input disc is rotated by an angle corresponding to the current state of the universal TM, the output disc will be rotated by an angle corresponding to the next state of the TM.

Suppose the current state is q_j and the current symbol is i . As the current symbol can be determined by applying a selection controller to $D_{current}$, the sequencer will be able to choose $M_{1,i}$ from $M_{1,0}, M_{1,1}, \dots, M_{1,m-1}$ and rotate the input disc of $M_{1,i}$ by an angle of λ_j , which is recorded by D_{state} . Then, D_{state} will be set at the angle recorded by the output disc of $M_{1,i}$ and the non-linear mapping device $M_{1,i}$ will be reset to its initial position.

In the remaining $2m$ non-linear mapping devices, $M_{2,0}, M_{2,1}, \dots, M_{2,m-1}$ are used to implement δ_2 ; and $M_{3,0}, M_{3,1}, \dots, M_{3,m-1}$ are used to implement δ_3 . For each i , $0 \leq i \leq m-1$, $M_{2,i}$ has the property that, if the input disc of $M_{2,i}$ is rotated by an angle of $\lambda_j = j\pi/(2\sigma)$ from its initial orientation, the output disc will be rotated by an angle of $2\delta_2(q_j, i)\pi/(2m+1)$ from its initial orientation. Similarly, for each i , $0 \leq i \leq m-1$, $M_{3,i}$ has the property that, if the input disc of $M_{3,i}$ is rotated by an angle of λ_j , the output disc will be rotated by an angle of $\pi/4$ if $\delta_3(q_j, i) = L$ or $\pi/2$ if $\delta_3(q_j, i) = R$.

Therefore, the next move and the replacing symbol will be represented by rotational positions of certain discs. These rotational positions will be used by the tape component to update the rotational positions of $D_{current}$, D_{left} and D_{right} according to the transition function δ .

3.3. Read-Write Tape Component. As mentioned above, the control component will generate two rotational positions, one representing the next move and the other representing the replacing symbol. Let $\omega_1\omega_2 \cdots \omega_{k_1} \cdots \omega_{k_2}$ be the current tape status and ω'_{k_1} be the replacing symbol.

For our convenience, we use $\beta_{current,t}$ to denote the angle of $D_{current}$ at the beginning of the t -th rotational cycle (of the sequencer). $\beta_{left,t}$, $\beta_{right,t}$ and $\beta_{state,t}$ are defined likewise. First suppose the next move of the head is to go rightward. There are three cases:

1 $k_1 = k_2$, i.e., the head is at the right end-mark:

The status of the tape will be changed to $\omega_1\omega_2\cdots\omega'_{k_1}\omega_{k_2+1}$. Here $\omega_{k_2+1} = 2$ (indicating the right end-mark) as the head will pad a right end-mark to the right. Therefore, we have the following recursive equations:

$$(3.4) \quad \beta_{current,t+1} = 2\frac{2\pi}{2m+1}$$

$$(3.5) \quad \beta_{right,t+1} = 0$$

$$(3.6) \quad \beta_{left,t+1} = \frac{\beta_{left,t}}{2m+1} + \frac{2\pi\omega'_{k_1}}{2m+1}$$

2 $k_1 = 1$, i.e., the head is at the left end-mark:

As the universal TM will never overwrite the left end-mark, ω_1 will still be 1 (indicating the left end-mark) and the status of the tape will be changed to $\omega_1\check{\omega}_2\omega_3\cdots\omega_t$. Therefore, we have

$$(3.7) \quad \beta_{current,t+1} = \frac{2\pi\omega_2}{2m+1}$$

$$(3.8) \quad \beta_{right,t+1} = \left(\beta_{right,t} - \frac{2\pi\omega_2}{2m+1}\right) \cdot (2m+1)$$

$$(3.9) \quad \beta_{left,t+1} = \frac{2\pi}{2m+1}$$

3 $1 < k_1 < k_2$:

The status of the tape will be changed to $\omega_1\omega_2\cdots\omega'_{k_1}\omega_{k_1+1}\omega_{k_1+2}\cdots\omega_{k_2}$. Thus, correspondingly, the recursive equations are

$$(3.10) \quad \beta_{current,t+1} = \frac{2\pi\omega_{k_1+1}}{2m+1}$$

$$(3.11) \quad \beta_{right,t+1} = \left(\beta_{right,t} - \frac{2\pi\omega_{k_1+1}}{2m+1}\right) \cdot (2m+1)$$

$$(3.12) \quad \beta_{left,t+1} = \frac{\beta_{left,t}}{2m+1} + \frac{2\pi\omega'_{k_1}}{2m+1}$$

Similarly, if the next move is to go leftward, there are two cases (the read-write head can not go leftward if it is at the left end-mark):

1 If $k_1 = k_2$, i.e., the head is at the right end-mark:

$$(3.13) \quad \beta_{current,t+1} = \frac{2\pi\omega_{k_1-1}}{2m+1}$$

$$(3.14) \quad \beta_{right,t+1} = \begin{cases} \frac{2\cdot 2\pi}{2m+1} & \text{if } \omega'_{k_1} = 2 \\ \frac{2\pi\omega'_{k_1}}{2m+1} + \frac{2\cdot 2\pi}{(2m+1)^2} & \text{if } \omega'_{k_1} \neq 2 \end{cases}$$

$$(3.15) \quad \beta_{left,t+1} = \left(\beta_{left,t} - \frac{2\pi\omega_{k_1-1}}{2m+1}\right)(2m+1)$$

2 If $1 < k_1 < k_2$:

$$(3.16) \quad \beta_{current,t+1} = \frac{2\pi\omega_{k_1-1}}{2m+1}$$

$$(3.17) \quad \beta_{right,t+1} = \frac{\beta_{right,t}}{2m+1} + \frac{2\pi\omega'_{k_1}}{2m+1}$$

$$(3.18) \quad \beta_{left,t+1} = (\beta_{left,t} - \frac{2\pi\omega_{k_1-1}}{2m+1})(2m+1)$$

With selection controllers, the system can change the rotational positions of $D_{current}$, D_{left} and D_{right} from $\beta_{current,t}$, $\beta_{left,t}$ and $\beta_{right,t}$ to $\beta_{current,t+1}$, $\beta_{left,t+1}$ and $\beta_{right,t+1}$ respectively according to the recursive equations described above.

3.4. Putting It Together. With mechanical components that can simulate the transition function and the tape of a universal TM, the frictional mechanical system that simulates the universal TM is almost immediate. The movements of the objects in this system are controlled by the rotations of the sequencer. In each cycle of rotation of the sequencer, the frictional mechanical system will finish the simulation of one step of the universal TM.

At the beginning of each cycle, the symbol at the current head location is decided depending on the rotational position of $D_{current}$. According to the current symbol as well as the current state recorded by D_{state} , the control component will make a sequence of moves and then decide (i) the next state, (ii) the symbol replacing the current symbol and (iii) the next move of the tape head. The tape component will then make a sequence of moves to change the rotational positions of $D_{current}$, D_{left} and D_{right} according to the symbol replacing the current symbol and the next move. This finishes the simulation of one step of the computation of the universal TM.

Initially, D_{state} is set at the orientation encoding the start state of the TM. The orientations of $D_{current}$, D_{left} and D_{right} also correspond to the initial status of the read-write tape. The simulation for the computation of the universal TM terminates when the rotational position of D_{state} is found to be corresponding to the accepting or rejecting state of the universal TM (we call these rotational positions *terminating orientations*) at the beginning of a rotational cycle of the sequencer.⁵

Herewith we have proved the following theorem:

THEOREM 3.1. *For any universal TM M , a frictional mechanical system can be constructed which has the property that, for any input string ω of M , the objects to the system can be set in a corresponding initial configuration so that a specified final configuration can be reached if and only if M accepts ω .*

As M is a universal TM, our frictional mechanical system can be used for general-purpose computing. Thus, it has the computational power of any conventional electronic computer.

4. A Frictional Mechanical System with Error. As mentioned in the introduction of this paper, our mechanical frictional system can simulate a universal TM without any error only if the system can be constructed and work exactly as it is specified. In the presence of errors, it is not possible to record the current configuration of the universal TM by the rotational positions of D_{state} , $D_{current}$, D_{left} and

⁵Observe that D_{state} might be at a terminating orientation momentarily as it rotates around. However, if at the beginning of a rotational cycle D_{state} stays at a terminating orientation, this means that the universal TM has reached an accepting or rejecting state.

D_{right} exactly as it is. Therefore, the computational power of the frictional mechanical system is restricted as the errors in the rotational positions of these discs may be accumulated significant enough to induce incorrect result in the simulation of the universal TM.

We use ϵ to denote the upper bound for the error in an angle which occurs in a single mechanical operation. ϵ is determined by the degree of accuracy in constructing mechanical devices as well as measuring rotational (and transitional) displacements.

4.1. A Constant Size Frictional Mechanical System in ϵ -error Model.

We first discuss a frictional mechanical system in this ϵ -error model; this system has exactly the same structure as the system in the exact model and therefore it only has a constant number of parts.

Let $\beta'_{state,t}$ denote the correct angle of D_{state} at the beginning of the t -th rotational cycle if the frictional mechanical system has no error. Further, let $\Delta_{state,t} = |\beta'_{state,t} - \beta_{state,t}|$, i.e., $\Delta_{state,t}$ is the error in $\beta_{state,t}$. $\beta'_{current,t}$, $\Delta_{current,t}$, $\beta'_{left,t}$, $\Delta_{left,t}$, $\beta'_{right,t}$ and $\Delta_{right,t}$ are defined accordingly.

In each rotational cycle, it takes a sequence of angle operations to generate each of $\beta_{state,t+1}$, $\beta_{current,t+1}$, $\beta_{left,t+1}$ and $\beta_{right,t+1}$ in the frictional mechanical system. Each operation involves in adding an angle to (or deducting an angle from) the rotational position of a disc, multiplying an angle by a constant factor, or a non-linear mapping. It is easy to see that there exists two constants c_1 and c_2 , such that, if the error in a rotational position is Δ , after any single operation, the error is bounded by $c_1\Delta + c_2\epsilon$. Further, as in each cycle the number of operations performed to get $\beta_{current,t+1}$, $\beta_{right,t+1}$, $\beta_{left,t+1}$ and $\beta_{state,t+1}$ is bounded by a constant, we can assume that there are two constants C_1 and C_2 such that: $\Delta_{state,t+1} < C_1\Delta_{state,t} + C_2\epsilon$, $\Delta_{current,t+1} < C_1\Delta_{current,t} + C_2\epsilon$, $\Delta_{right,t+1} < C_1\Delta_{right,t} + C_2\epsilon$ and $\Delta_{left,t+1} < C_1\Delta_{left,t} + C_2\epsilon$.

For simplicity, we assume $C_1, C_2 > 2$.

There are only m valid values for $\beta'_{current,t+1}$, (i.e., $0, 2\pi/(2m+1), \dots, 2(m-1)\pi/(2m+1)$) and the difference between any two valid values is at least $2\pi/(2m+1)$. Therefore, as long as $\Delta_{current,t+1} < \pi/(2(2m+1))$, a selection controller can be used to determine $\beta'_{current,t+1}$ from $\beta_{current,t+1}$. This means that $\beta_{current,t+1}$ can be corrected at the beginning of each cycle, given that $\epsilon < d_1/m$ for some constant $d_1 > 0$.

The same analysis applies to $\beta_{state,t+1}$: if $\epsilon < d_2/\sigma$ for some constant $d_2 > 0$, $\beta_{state,t+1}$ can be corrected at the beginning of each cycle. Therefore, only the errors in $\beta_{left,t}$ and $\beta_{right,t}$ may be accumulated to the next cycle. As at each step, $\beta_{left,t}$ and $\beta_{right,t}$ are only used to distinguish ω_{k_1-1} and ω_{k_1+1} , errors in $\beta_{left,t}$ and $\beta_{right,t}$ will not cause any incorrectness if they are less than $\pi/(2(2m+1))$.

Observe that if the read-write head reaches the left end-mark, the error in $\beta_{left,t}$ will be discarded. This is because when the head moves rightward from the end-mark, none of $\beta_{current,t+1}$, $\beta_{right,t+1}$ and $\beta_{left,t+1}$ will depend on $\beta_{left,t}$ as shown in (3.7), (3.8) and (3.9). Similarly, if the read-write head reaches the right end-mark, the error in $\beta_{right,t}$ will be discarded too. Therefore, if the head of the universal TM visits the left and right end-marks periodically during the computation, the errors in β_{left} and β_{right} will be corrected periodically.

We define another TM M' . M' will simulate the computation of M . The difference between M' and M is that, each time after M' finishes simulating M by a constant number K of steps of computation, (this constant will be specified later,) the head of M' will make a sweep of the tape. By ‘‘sweep,’’ we mean that it will first

move leftward until it reaches the left end-mark; then it will move rightward until it reaches the right end-mark; afterwards, it will move leftward again and return to the location where it was before the sweep. After it finishes the sweep, M' will start simulating the next K steps of the computation of M . It is easy to see that M' has the property that, at any time, if the current working space is s , the head of M' will reach the left and right end-marks within at most $2S + 2K$ steps.

Clearly, M' is equivalent to M in the sense that it decides exactly the same language as M does. Therefore, to simulate M , it suffices to construct a frictional mechanical system that can simulate M' . For any input string ω of M' , if M' decides ω in space S , the errors in β_{left} and β_{right} will be accumulated through at most $2S + 2K$ steps before they are discarded.

Thus, if the errors in β_{left} and β_{right} accumulated in $2S + 2K$ steps can be bounded by $\pi/(2(2m + 1))$ (otherwise incorrect values of ω_{k_1-1} and ω_{k_1+1} might be used), it will not induce any incorrect result in simulating the universal TM M' (and thus in simulating M) on input string ω .

Suppose at the t -th step the read-write head moves rightward from the left end-mark. As the accumulated error of β_{left} is discarded at this moment, $\Delta_{left,t}$ is bounded by a constant, say, $C_3\epsilon$. As $\Delta_{left,t+1} < C_1\Delta_{left,t} + C_2\epsilon$, we have

$$\begin{aligned} \Delta_{left,t+2S+2K} &< C_1^{2S+2K} \Delta_{left,t} + C_2 \frac{C_1^{2S+2K} - 1}{C_1 - 1} \epsilon \\ &< C_1^{2S+2K} C_3 \epsilon + C_2 C_1^{2S+2K} \epsilon. \end{aligned}$$

To bound Δ_{left} by $\pi/(2(2m + 1))$, it suffices to let

$$\epsilon < \frac{1}{2(2m + 1)(C_3 C_1^{2S+2K} + C_2 C_1^{2S+2K})} < d \cdot 2^{-c(S+K)}$$

for some constants c and d .

If we let K be 1, M' will sweep the tape after each step of simulating M . Therefore, for any input string ω , it will take at most $(2S + 1)T$ steps for M' to decide ω if M decides ω in time T and space S . Since $S \leq T$, the total time used by M' is bounded by $(2T + 1)T$. Hence, we have the following theorem:

THEOREM 4.1. *For any universal TM M , a frictional mechanical system with error can be constructed to simulate M . It has the property that, for any space bound S , if the single-operation error of the system, ϵ , is bounded by $\min\{d \cdot 2^{-cS}, d_1/m, d_2/\sigma\}$ for some constants c , d , d_1 and d_2 , then given any input string ω that M decides in space bound S , the frictional mechanical system will reach a distinguished final configuration from an initial configuration encoding ω if and only if M accepts ω . Further, the frictional mechanical system will take at most $(2T + 1)T$ cycles to finish the computation if M decides ω in T steps.*

If we let $K = S$, M' will simulate M by S steps between two consecutive sweeps. Thus, it will take at most three times as much time as M takes to finish the computation. The disadvantage is that now M' (and hence our frictional mechanical system) depends on S . For different value of S , a different frictional mechanical system needs to be constructed to simulate M .

With some additional reasonable assumption, we can show that a periodic sweeping is not necessary. Given that $\epsilon = O(2^{-cS})$, if M can decide an input string ω in space S , our frictional mechanical system will finish simulating M with the correct result. Due to the length of the proof, we will include it in the appendix.

4.2. A Frictional Mechanical System in ϵ -error Model with $\epsilon = \Omega(1)$.

The analysis of the above system implies that $1/\epsilon$ will have to increase exponentially as S increases to maintain the correctness of the frictional mechanical system. If in a frictional mechanical system $O(S)$ discs instead of three discs are used to encode the contents of the tape, each representing one cell, this frictional mechanical system in ϵ -error model can simulate the universal TM M with space bound S where $\epsilon = \Omega(1)$. This is because now each disc has only constant number of valid rotational positions and selection controllers can be used to distinguish the correct value represented by the disc. This is better than the previous result in the sense that $1/\epsilon$ does not increase as S increases. However, the frictional mechanical system is now dependent on S , as it needs S discs to encode the tape. In other words, this model of frictional mechanical systems is a digital computer, just like the Analytical Engine. If we want to increase the computational power of the frictional mechanical system by using larger space bound S , we will have to add more discs used to represent the tape.

More specifically, for the k -th cell of the tape, four discs are used to represent the current state of the cell: $D_{k,symbol}$ represents the symbol in the current cell; $D_{k,head}$ indicates whether the read-write head is located at this cell; $D'_{k,head}$ indicates whether the read-write head was located at this cell at the end of last step; and $D_{k,next}$ indicates the next move of the head. Also, for each cell, there is a set of non-linear mapping devices which implement the transition function. We call all the parts used to represent the k -th cell the k -th cell component and denote it by C_k . In addition, there is a single disc D_{state} which records the current state of the universal TM.

At the beginning of each rotational cycle of the power disc, both $D_{k,head}$ and $D'_{k,head}$ are set at $\pi/4$ if the head is located at the k -th cell or 0 if not. $D_{k,symbol}$ is set at $2i\pi/(2m+1)$ if the symbol in the k -th cell is i . And D_{state} is set at $j\pi/(2\sigma)$, indicating that the current state of the TM is q_j . Then, if the rotational position of $D'_{k,head}$ is $\pi/4$, the rotational position of $D_{k,symbol}$ will be changed to $2i'\pi/(2m+1)$ where $i' = \delta_2(q_j, i)$. If the rotational position of $D'_{k,head}$ is 0, $D_{k,symbol}$ will be clamped so that its rotational position will not be changed. Also, the rotational position of $D_{k,next}$ is set at 0 if the next move is to go leftward and $\pi/4$ if the next move is to go rightward.

The only thing remaining is to change each $D_{k,head}$ accordingly. More specifically, for each k , if the rotational position of $D_{k,head}$ is $\pi/4$, (i.e., the head is at the k -th cell,) $D_{k-1,head}$ (or $D_{k+1,head}$) will be changed to $\pi/4$ if $D_{k,next}$ is set at 0 (or $\pi/4$, respectively) while $D_{k,head}$ will be reset to 0; if the rotational position of $D_{k,head}$ is 0, $D_{k,head}$, $D_{k-1,head}$ and $D_{k+1,head}$ will be clamped.

The problem is that there will be some conflicts in the movements of different components. For example, suppose the current head is located at the k_1 -th cell. Then, k_1 -th component will try to change $D_{k_1-1,head}$ in case the next move is to go leftward. However, the (k_1-1) -th component itself will try to clamp $D_{k_1-1,head}$ as the rotational position of $D'_{k_1-1,head}$ is 0. To resolve this problem, we divide the S head components into three groups so that C_k belongs to the $(k \bmod 3)$ -th group. First, the components in group 0 will be activated to change all the $D_{k,head}$ s according to the rules described above. Then, the components in group 1 will be activated. And last are the components in group 2. By this, no two components that have conflict movement will be activated at the same time.

After all components have been activated, for each k , $D_{k,head}$ will represent whether the read-write head is located at the k -th cell at the beginning of the next step. Then, the rotational position of $D_{k,head}$ is copied to $D'_{k,head}$ so that each com-

ponent is ready for the simulation of the next computation of the universal TM.

It is easy to see that for a frictional mechanical system constructed as above, ϵ can be set to a constant regardless of the space bound S , as each disc only has a constant number of valid orientations and operations only constant number of operations are needed to finish one step of simulation.

5. Reduction from A_{TM} to the Frictional Mover's Problem. We prove the undecidability of the frictional mover's problem by a reduction from the A_{TM} problem. It suffices to show that every input of the A_{TM} problem, which is the description of a Turing machine M along with an input string ω of M , can be transformed to an input of the frictional mover's problem, which is a frictional mechanical system with initial and goal configurations such that (i) the transformation can be performed by a specified procedure that terminates in a finite period of time; and (ii) M accepts ω if and only if the frictional mechanical system can reach the goal configuration from its initial configuration.

Therefore, it is necessary that all the objects in the resulting frictional mechanical system and their positions can be described by rational coefficients. For example, for each disc, its radius as well as the location of its center need to be specified by rational numbers. Similarly, the length, orientation, and location of a bar need to be specified by rational numbers.

Recall that a universal frictional mechanical system only involves operations that add to a value (represented by the rotational position of a certain disc) a rational number, or multiply the value by a rational number. Therefore, we can easily specify all the discs (including partial discs), bars, and cylinders in the system by rational coefficients without changing the functionalities of the various devices constructed by these parts.

The only exception is the irregular discs used in the non-linear mapping devices. Each irregular disc has to be specified by some irrational coefficients, as indicated by Equation 2.1. Here we prove the following lemma:

LEMMA 5.1. *The possibly irrational coefficients specifying the irregular discs used in the non-linear mapping devices can be replaced by rational numbers without inducing any error in simulating the universal TM.*

Proof. Let D_1 be an irregular disc. It can be specified by the following two sets of parameters: $\{\theta_1, \theta_2, \dots, \theta_n\}$, $\{r_1(\theta_0), r_1(\theta_1), \dots, r_1(\theta_n)\}$. (Recall that $\theta_0 = 0$.) The boundary of D_1 inside the interval of $[\theta_i, \theta_{i+1}]$ is described by a linear function $r_1(\theta) = ((\theta - \theta_i) \cdot r_1(\theta_{i+1}) + (\theta_{i+1} - \theta) \cdot r_1(\theta_i)) / (\theta_{i+1} - \theta_i)$, whose coefficients are entirely determined by $\theta_i, \theta_{i+1}, r_1(\theta_i)$, and $r_1(\theta_{i+1})$. Recall that, in a universal frictional mechanical system, non-linear mapping devices are used to implement the transition function δ , which only takes rational numbers as input. Therefore, $\theta_1, \theta_2, \dots, \theta_n$ are all rational numbers. However, the exact value of each $r_1(\theta_i)$, denoted by $r_{1,i}$, could be an irrational number as the computation involves irrational number $\sqrt{2}$ as well as the function \tan . We replace each $r_{1,i}$ by a rational number $r'_{1,i}$ such that $|r'_{1,i} - r_{1,i}| < \epsilon'$, for some small $\epsilon' > 0$.

The non-linear mapping device constructed with these "rationalized" parameters may cause two types of errors, *mechanical errors* and *logical errors*. A mechanical error occurs when two mechanical parts lose contact with, or crush into, each other while they are supposed to maintain continuous contact. A logic error occurs when the error of the resulting angle α causes a false interpretation by the universal frictional mechanical system.

We first show that mechanical errors can be avoided. We let D_2 be the coupling

irregular disc of D_1 in the same non-linear mapping device. We need to show that D_2 can also be described by rational numbers while keeping $r_1(\theta) + r_2(\theta + \pi)$ a constant for all $\theta \in [0, 2\pi]$. For each $\theta_i, i = 0, 1, 2, \dots, n$, the radius of D_2 at angle $\theta_i + \pi$ is computed by $r_2(\theta_i + \pi) = \frac{|O_1O_2|}{\theta_{i+1}-\theta_i} - d - r_1(\theta_i)$. Since now $r_1(\theta_i)$ is chosen to be a rational number $r'_{1,i}$, and $|O_1O_2|$ and d are all rational numbers, $r_2(\theta_i + \pi)$ is a rational number. Therefore, the boundary of D_2 can be decomposed into $n+1$ sub-boundaries, corresponding to angular intervals $[\pi, \pi + \theta_1], [\pi + \theta_1, \pi + \theta_2], \dots, [\pi + \theta_{n-1}, \pi + \theta_n], [\theta_n - \pi, \pi]$, respectively. In each interval, the boundary can be specified by a linear function $r_2(\theta)$ of θ , just like the case for D_1 . Therefore, D_2 can be described by rational numbers. For any $\theta \in (\theta_i, \theta_{i+1})$,

$$\begin{aligned} r_1(\theta) + r_2(\theta + \pi) &= \frac{(\theta - \theta_i) \cdot r_1(\theta_{i+1}) + (\theta_{i+1} - \theta) \cdot r_1(\theta_i)}{\theta_{i+1} - \theta_i} + \frac{(\theta - \theta_i) \cdot r_2(\theta_{i+1} + \pi) + (\theta_{i+1} - \theta) \cdot r_2(\theta_i + \pi)}{\theta_{i+1} - \theta_i} \\ &= \frac{(\theta - \theta_i)(r_1(\theta_{i+1}) + r_2(\theta_{i+1} + \pi)) + (\theta_{i+1} - \theta)(r_1(\theta_i) + r_2(\theta_i + \pi))}{\theta_{i+1} - \theta_i} \\ &= \frac{|O_1O_2|}{\theta_{i+1} - \theta_i} - d \end{aligned}$$

This implies that irregular disc D_2 as described above still completely “complements” D_1 as defined in Subsection 2.6. Therefore, during each operation D_1 and D_2 maintain continuous contact with the horizontal bar but do not compress it, thus causing no mechanical error.

Next we show that any mechanical error can be corrected. With the “rationalized” non-linear mapping device, when the input angle is θ_i , the output angle is not exactly α_i , but a value α'_i very close to α_i . Since non-linear mapping devices are used to implement the transition function, each α_i is a rational number as it encodes either a symbol or a state. Further, there exists a constant ϵ'' such that $|\alpha_i - \alpha_j| > \epsilon''$ if $\alpha_i \neq \alpha_j$. Here $\epsilon = c \cdot \min\{\frac{1}{\sigma}, \frac{1}{m}\}$, where $c = \Omega(1)$. Therefore, there exists another constant $c' = \Omega(1)$ such that $|\alpha_i - \alpha'_i| < \frac{\epsilon''}{3}$ for all i , if $\epsilon' < c' \cdot \min\{\frac{1}{\sigma}, \frac{1}{m}\} \cdot R$. Hence, a selection controller is able to find out the correct value of α_i from α'_i , as α'_i is closer to α_i than to any other valid value.

This finishes the proof. \square

With this lemma, we have:

THEOREM 5.2. *The frictional mover’s problem is undecidable.*

6. Conclusion. In this paper we introduced frictional mechanical systems and proved that a universal frictional mechanical system can simulate the computation of a universal Turing Machine. We also gave some results for the case where there are limited errors for the mechanical parts in the system. Our work implies that the frictional mover’s problem is undecidable. It is, however, unclear to us what is the implication of our work to building nano-computers at the macro-molecule level when the nanotechnology further matures.

Acknowledgement. This research was supported in part by grants DARPA/AFSOR F30602-01-2-0561, NSF ITR EIA-0086015, NSF EIA-0218376, and NSF EIA-0218359.

REFERENCES

- [1] T. ASANO, D. KIRKPATRICK, AND C. K. YAP, *d₁-optimal motion for a rod*, in Proceedings of the 12th Annual ACM Symposium On Computational Geometry, 1996, pp. 252–263.
- [2] J. CANNY, *Some algebraic and geometric computations in PSPACE*, in Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 1988, pp. 460–467.
- [3] J. CANNY AND J. H. REIF, *New lower bound techniques for robot motion planning problems*, in Proceedings of the 28th Annual Symposium on Foundations of Computer Science, 1987, pp. 49–60.

- [4] P. C. CHEN AND Y. K. HWANG, *Practical path planning among movable obstacles*, in Proceedings of the 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 444–449.
- [5] B. DACRE-WRIGHT, J.-P. LAUMOND, AND R. ALAMI, *Motion planning for a robot and a movable object amidst polygonal obstacles*, in Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 1992, pp. 2474–2480.
- [6] M. ERDMANN, *On a representation of friction in configuration space*, International Journal of Robotics Research, 13 (1994).
- [7] J. E. HOPCROFT, D. A. JOSEPH, AND S. H. WHITESIDES, *Movement problems for 2-dimensional linkages*, SIAM Journal on Computing, 13 (1984), pp. 610–629.
- [8] J. E. HOPCROFT, J. T. SCHWARTZ, AND M. SHARIR, *On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the “warehouseman’s problem”*, International Journal of Robotics Research, 3 (1984), pp. 76–88.
- [9] J. H. REIF, *Complexity of the mover’s problem and generalizations*, in Proceedings of the 20th IEEE Symposium on Foundations of Computer Science, 1979, pp. 421–427.
- [10] J. H. REIF AND M. SHARIR, *Motion planning in the presence of moving obstacles*, in Proceedings of the 26th Annual Symposium on Foundations of Computer Science, 1985, pp. 144–154.
- [11] J. H. REIF AND Z. SUN, *Movement planning in the presence of flows*, in Proceedings of the 7th International Workshop on Algorithms and Data Structures, vol. 2125 of Lecture Notes in Computer Science, 2001, pp. 450–461.
- [12] J. H. REIF AND H. WANG, *The complexity of the two dimensional curvature-constrained shortest-path problem*, in Proceedings of the 3rd Workshop on Algorithmic Foundations of Robotics, 1998, pp. 49–57.
- [13] J. SELLEN, *Lower bounds for geometrical and physical problems*, SIAM Journal on Computing, 25 (1996), pp. 1231–1253.
- [14] P. SPIRAKIS AND C. K. YAP, *Strong NP-hardness of moving many discs*, Information Processing Letters, 19 (1984), pp. 55–59.
- [15] G. WILFONG, *Motion planning in the presence of movable obstacles*, in Proceedings of the 4th Annual ACM Symposium on Computational Geometry, 1988, pp. 279–288.

Appendix A. Error Bound for β_{right} and β_{left} under Additional Assumption.

We have proved that by forcing the read-write head to visit the left and right end-marks periodically, the errors in β_{left} and β_{right} can be discarded after certain number of steps so that they will not affect the result of simulating the universal TM. Here, we will show that, with the following assumption, there is no need for periodic sweeps of the tape.

ASSUMPTION 1. *For any operation that will multiply an angle by a constant factor C , if before the operation the error in the angle is Δ , after the operation, the error will become $C\Delta + k\epsilon$ for some constant k .*

Under this assumption, an error in an angle θ will be reduced if the operation is to reduce θ by a constant factor (i.e., $C < 1$).

Now we will examine how the error in β_{right} is accumulated in each step using this assumption. we decompose $\Delta_{right,t}$ into two components: $\Delta_{right,t+1}^1$ and $\Delta_{right,t+1}^2$. The first component is due to $\Delta_{right,t}$, and the other component is the error which occurs during the angle operations in this cycle. As in one cycle the number of operations to generate β_{right} is bounded by a constant number, we can assume $\Delta_{right,t+1}^2 \leq C_1'\epsilon$ for some constant C_1' . $\Delta_{right,t+1}^1$, however, is determined by $\Delta_{right,t}$ as well as the angle operation on $\beta_{right,t}$ to generate $\beta_{right,t+1}$.

If the next move is rightward, according to (3.8) and (3.11), $\beta_{right,t}$ is multiplied by a factor of $(2m+1)$ and then added to another angle to get $\beta_{right,t+1}$. Therefore, the error in $\beta_{right,t}$ is amplified by a factor of $(2m+1)$, i.e., $\Delta_{right,t+1}^1 = \Delta_{right,t} \cdot (2m+1)$. Similarly, if the next move is leftward, we have $\Delta_{right,t+1}^1 = \Delta_{right,t}/(2m+1)$, as in (3.14) and (3.17) $\beta_{right,t+1}$ has a component $\beta_{right,t}/(2m+1)$.

Therefore, in each cycle, the error in β_{right} which is accumulated in previous cycles is either amplified by a factor of $(2m+1)$ or reduced to $1/(2m+1)$ depending

on the move of the head. Hence we have the following recursive function:

$$(A.1) \quad \Delta_{right,t+1} \leq f(t)\Delta_{right,t} + \Delta_{right,t+1}^2$$

Here $f(t)$ is defined as follows:

$$f(t) = \begin{cases} 2m+1 & \text{if the move of the head in the } t\text{-th step is rightward} \\ \frac{1}{2m+1} & \text{if the move of the head in the } t\text{-th step is leftward} \end{cases}$$

$\Delta_{right,1}$ is the error in β_{right} at the beginning of the computation. This error is due to the inaccuracy in setting the starting orientation of D_{right} . We can assume that $\Delta_{right,1} \leq C'_2\epsilon$.

According the recursive equation, we can compute $\Delta_{right,t'+1}$ for any t' :

$$\begin{aligned} & \Delta_{right,t'} \\ & \leq f(t'-1)\Delta_{right,t'-1} + \Delta_{right,t'}^2 \\ & \leq f(t'-1)(f(t'-2)\Delta_{right,t'-2} + \Delta_{right,t'-1}^2) + \Delta_{right,t'}^2 \\ & \quad \vdots \\ & \quad \vdots \\ & \leq \Delta_{right,1}(\prod_{i=1}^{t'-1} f(i)) + \sum_{j=2}^{t'} \Delta_{right,j}^2 (\prod_{i=j}^{t'-1} f(i)) \end{aligned}$$

For any input string ω of M , we let S denote the working space M need to use to decide ω . It is easy to see that, for any t' and j , $\prod_{i=j}^{t'-1} f(i) \leq (2m+1)^S$ as the right moves could outnumber the left moves at most by S . Therefore, $\Delta_{right,t'}$ is bounded by

$$\Delta_{right,1}(2m+1)^S + \sum_{j=2}^{t'} \Delta_{right,j}^2 (2m+1)^S \leq C'_2\epsilon(2m+1)^S + 2^{cS}C'_1\epsilon(2m+1)^S$$

as $t' < 2^{cS}$ for some constant c .

To bound Δ_{right} by $1/(2(2m+1))$ (so that at each cycle correct value of ω_{k+1} could be retrieved from β_{right}), we only need

$$\epsilon \leq \frac{1}{2(2m+1)} \cdot \frac{1}{(\epsilon(2m+1)^S)(C'_2 + 2^{cS}C'_1)} < d' \cdot 2^{-c''S}$$

for some constants c'' and d' .

We have thus proved the following theorem:

THEOREM A.1. *For any universal TM M , a frictional mechanical system with error can be constructed to simulate M . It has the property that, for any space bound S , if the single-operation error of the system, ϵ , is bounded by $\min(d' \cdot 2^{-c''S}, d_1/m, d_2/\sigma)$ for some constants c'' , d' , d_1 and d_2 , then given any input string ω that M decides in space bound S , the frictional mechanical system will reach a distinguished final configuration from an initial configuration encoding ω if and only if M accepts ω . Further, the frictional mechanical system will take T cycles to finish the computation if M decides ω in T steps.*