

Investigation on Energetic Optimization Problems of Stochastic Thermodynamics with Iterative Dynamic Programming

Linchen Gong¹, Ming Li^{2,*} and Zhong-can Ou-Yang^{1,3}

¹ Center for Advanced Study, Tsinghua University, Beijing, 100084, China.

² College of Physical Science, Graduate university of Chinese Academy of Sciences, Beijing, 100190, China.

³ Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing, 100190, China.

Received 24 July 2009; Accepted (in revised version) 8 October 2009

Available online 6 January 2010

Abstract. The energetic optimization problem, *e.g.*, searching for the optimal switching protocol of certain system parameters to minimize the input work, has been extensively studied by stochastic thermodynamics. In this work, we study this problem numerically using iterative dynamic programming. The model systems under investigation are toy actuators consisting of spring-linked beads with loading force imposed on both ending beads. For the simplest case, *i.e.*, a one-spring actuator driven by tuning the stiffness of the spring, we compare the optimal control protocol of the stiffness for both the overdamped and the underdamped situations, and discuss how inertial effects alter the irreversibility of the driven process and thus modify the optimal protocol. Then, we study the systems with multiple degrees of freedom by constructing oligomer actuators, in which the harmonic interaction between the two ending beads is tuned externally. With the same rated output work, actuators of different constructions demand different minimal input work, reflecting the influence of the internal degrees of freedom on the performance of the actuators.

PACS: 05.40.-a, 82.70.Dd, 87.15.H-, 05.70.Ln

Key words: Iterative dynamic programming, stochastic thermodynamics, harmonic model.

1 Introduction

The past two decades have seen the rapid growth of researches on nano-machines. Various types of nano-machines have been proposed and synthesized for practical purposes [1, 2]. This progress presents an urgent appeal and also a great challenge for

*Corresponding author. *Email address:* liming@gucas.ac.cn (M. Li)

physicists to understand the working principle and energetics of nano-machines. For isothermal nano-machines including protein machines and their derivatives or mimics, researchers are able to formulate a conceptual framework, the Brownian ratchet, to understand in general how a *scalar* energy source (light, chemical reaction, thermal agitation, etc.) can facilitate a *vectorial* process. The ratchet mechanism has not only been used to explain many experimental observations on chemically-driven protein motors [3–5], but also been successfully employed to develop some micro-manipulation techniques (*e.g.*, a ratchet-like micro-device was designed for DNA segregation [6]). Within the ratchet framework, the energetics (or thermodynamics) has also been extensively investigated with multiple-state Langevin equations or Fokker-Planck equations (*e.g.*, see [3] for chemically-driven nano-machines, and [7] for externally-controllable nano-machines).

In this context, some general energetic topics of nano-machines can be put forward. Particularly, in analogy to the optimization theory based on macroscopic finite-time thermodynamics [8,9], the energetic optimization problems can be re-formulated for ratchet-type nano-machines. For instance, one may require the *minimal input work* (abbreviated as MIW) in a finite-time process to achieve least heat agitation [10] or maximum power [11] or highest efficiency [12]. To properly formulate these questions, choosing proper model systems is very important. While the multiple-state ratchet model may not be a proper candidate (especially, not easy for analytical treatment), Sekimoto and Seifert *et al.* initiated the study with a much simpler ratchet model. In this model, the machine performs Brownian motion on an energy landscape. And, a few parameters that characterize the landscape can be switched externally according to a deterministic protocol [11,15]. For such driven systems, stochastic thermodynamics has been firmly established, and the work and heat can be co-identified under the construction of first-law-like and second-law-like thermodynamic relations [15–17]. With the well-defined model systems and thermodynamics, a basic question is addressed, *i.e.*, what is the *optimal protocol* of the switching parameters to realize the *minimal input work* (abbreviated as OPMIW). For the simple cases that the Brownian motion of an overdamped or underdamped bead is controlled by a tunable harmonic potential, the authors were able to present analytical solutions for the OPMIWs (see [11,13] for overdamped case, and [14] for underdamped case). While their analysis focused on analytically solvable potentials, Then and Engel also presented a Monte Carlo numerical method to discuss similar optimization problems of more complicated systems [10].

In this work, we follow the same logic and introduce a numerical method called *iterative dynamic programming* (abbreviated as IDP) [18] to investigate the OPMIW problems. For systems with polynomial potential functions up to the second order, the OPMIW problem is equivalent to the conventional optimal control problem, thus can be solved numerically with IDP. Inspired by the controllable DNA nano-actuator converting chemical energy into mechanical work [19,20], we choose toy actuators as our model systems. The simplest construction is a two-bead actuator, in which the two beads are linked by a spring with externally tunable stiffness (a polymer can be regarded as such an actuator if its persistence length is tuned externally). Although this model has been

discussed by Seifert *et al.* [11], the promising numerical method enables us to study the more complicated case when the actuator is working against a pulling force, and consequently to study the relation between the output work and the input work. Furthermore, since IDP can efficiently treat systems with multiple degrees of freedom [18], actuators with different internal structures can be constructed and the related OPMIW problems can be discussed by this method. Besides, additional constraints can be easily specified (as will be clear in subsequent sections) to study some interesting issues on energetics.

In the following, we will first present the formulation of OPMIW problems for various toy models. In Section 3, after testing the numerical precision of IDP for two analytically solvable examples, we apply IDP to simple systems with overdamped or underdamped dynamics, and also to systems with multiple degrees of freedom. Some general remarks will be presented in Section 4. The details of IDP method can be found in the appendix.

2 Formulation for OPMIW problem

In this section, we will present the mathematical formulation of OPMIW problem for various model systems. We consider the simplest model of a one-spring actuator first, and then generalize the description to cases with multiple degrees of freedom.

2.1 Formulation for one-spring actuator

The one-spring actuator is basically the same model that Seifert and colleagues have considered before, in which a bead is trapped in a stiffening harmonic potential [11]. The actuator is illustrated in Fig. 1(a), where two beads are connected by a spring with externally tunable stiffness. To study the energetic problems, one bead is fixed at the origin while the other is pulled by a constant force.

We only focus on one-dimensional problems here, thus the potential energy of the one-spring actuator can be written as

$$V(x, \lambda) = \frac{1}{2} \lambda x^2 - f x, \quad (2.1)$$

where λ is the stiffness of the spring, x is the position of the movable bead, and f is a constant pulling force (suppose f is pointing to the positive direction of x -axis). For the overdamped case, the Langevin equation describing the motion of the movable bead is

$$\gamma \frac{dx}{dt} = -\lambda x + f + \sqrt{2k_B T \gamma} \zeta(t), \quad (2.2)$$

where γ is the frictional coefficient, k_B is the Boltzmann constant, T is the temperature of the thermostat, and $\zeta(t)$ is Gaussian white noise satisfying

$$\langle \zeta(t) \zeta(t') \rangle = \delta(t - t'). \quad (2.3)$$

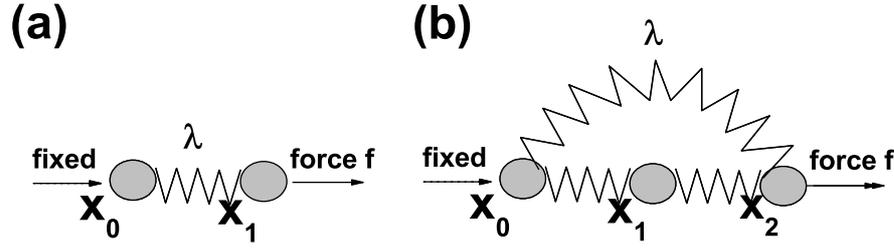


Figure 1: Illustration of the models in study. (a) The one-spring actuator with constant pulling force f . (b) The oligomer actuator. The $N=2$ case is portrayed.

In the following, γ and $k_B T$ are set equal to unity for simplicity. If the unit of another quantity is specified, *e.g.*, the force f , the dimensions of all the quantities involved in the model can be expressed by the dimensions of γ , $k_B T$ and f . The Fokker-Planck equation for the probability evolution of x reads

$$\frac{\partial g(x,t)}{\partial t} = \frac{\partial}{\partial x} \left[g(x,t) \frac{\partial V(x,\lambda)}{\partial x} + \frac{\partial g(x,t)}{\partial x} \right], \quad (2.4)$$

where $g(x,t)$ is the probability distribution function of x at time t .

According to stochastic thermodynamics [11], under the framework of Langevin equation [Eq. (2.2)], the work increment dw along the stochastic trajectory of x is

$$dw = \frac{\partial V}{\partial \lambda} \dot{\lambda} dt = \frac{1}{2} \dot{\lambda} x^2 dt. \quad (2.5)$$

Averaging with the Fokker-Planck equation [Eq. (2.4)], when λ is switched from an initial value λ_i to a final value λ_f within finite time, the expected input work W_{in} to the actuator can be expressed as

$$W_{in} = \int_{t_i}^{t_f} \frac{1}{2} \dot{\lambda} \langle x^2 \rangle dt, \quad (2.6)$$

where t_i and t_f are the beginning and ending time point of the switching process respectively. The dot over λ denotes the time derivative of λ . The angular brackets denote an instant ensemble average using the probability distribution $g(x,t)$. During the switching process, the average position $\langle x \rangle$ is changing, consequently the finite-time output work of the actuator is defined as

$$W_{out} = f \langle x \rangle |_{t=t_i} - f \langle x \rangle |_{t=t_f}. \quad (2.7)$$

There exist several differential constraints on the moments of $g(x,t)$ when optimizing W_{in} . Multiplying both sides of Eq. (2.4) by x or x^2 and taking average, we obtain the following relations

$$\langle \dot{x} \rangle = f - \lambda \langle x \rangle, \quad (2.8)$$

$$\langle \dot{x}^2 \rangle = 2 - 2\lambda \langle x^2 \rangle + 2f \langle x \rangle. \quad (2.9)$$

In IDP calculation, we usually consider W_{in} as a function of time by defining

$$W_{in}(t) = \int_{t_i}^t \frac{1}{2} \dot{\lambda} \langle x^2 \rangle dt, \quad (2.10)$$

which satisfies the following differential equation

$$\dot{W}_{in} = \frac{1}{2} u \langle x^2 \rangle. \quad (2.11)$$

Here, the time derivative of λ -protocol has been explicitly expressed as a temporal function u , *i.e.*,

$$\dot{\lambda} = u. \quad (2.12)$$

Eqs. (2.11) and (2.12), together with Eqs. (2.8) and (2.9), compose the formulation of the OPMIW problem for current system. The task now turns into searching for the optimal trajectory of u that minimizes $W_{in}(t_f)$. Once t_i , t_f , λ_i , λ_f and the initial values of $\langle x \rangle$ and $\langle x^2 \rangle$ are all specified [$W_{in}(0)$ is zero by definition], the problem can be solved with IDP method.

For underdamped one-spring actuator, while W_{in} is the same with Eq. (2.6), the set of differential constraints become

$$\langle \dot{x}^2 \rangle = 2 \langle xp \rangle / m, \quad (2.13)$$

$$\langle \dot{x}p \rangle = \langle p^2 \rangle / m - \langle xp \rangle / m - \lambda \langle x^2 \rangle + f \langle x \rangle, \quad (2.14)$$

$$\langle \dot{p}^2 \rangle = -2 \langle p^2 \rangle / m - 2\lambda \langle xp \rangle + 2f \langle p \rangle + 2, \quad (2.15)$$

$$\langle \dot{x} \rangle = \langle p \rangle / m, \quad (2.16)$$

$$\langle \dot{p} \rangle = -\langle p \rangle / m - \lambda \langle x \rangle + f, \quad (2.17)$$

where p and m are the momentum and mass of the movable bead, respectively.

2.2 Generalization to models with multiple degrees of freedom

In this subsection, we consider a nano-system with coordinates vector \vec{x} and time-dependent potential function $V(\vec{x}, \vec{\lambda}(t))$. The parameter vector, $\vec{\lambda}$, of the potential is externally tuned according to certain protocol. To model the actuation process, constant forces are loaded on the system, thus the potential function can be written as

$$V(\vec{x}, \vec{\lambda}) = V_0(\vec{x}, \vec{\lambda}) - \vec{f} \cdot \vec{x}, \quad (2.18)$$

where, $V_0(\vec{x}, \vec{\lambda})$ represents the internal energy of the system (*e.g.*, the elastic energy of the springs in the toy actuator). \vec{f} is not a normal three-dimensional force vector, instead, it is a high-dimensional vector with each element representing the force on the corresponding degree of freedom.

The dynamics of \vec{x} can be described by overdamped or underdamped Langevin equations. For example, in overdamped case, the Langevin equations read

$$\frac{dx_j}{dt} = -\frac{\partial V(\vec{x}, \vec{\lambda})}{\partial x_j} + \sqrt{2}\zeta_j(t), \quad j=1,2,\dots,d, \quad (2.19)$$

where d is the dimension of \vec{x} . For simplicity, we suppose that the random forces on different elements of \vec{x} are uncorrelated, and the frictional coefficients are all the same. Thus, the noise terms $\{\zeta_j\}$ ($j=1,2,\dots,d$) satisfy

$$\langle \zeta_i(t)\zeta_j(t') \rangle = \delta(t-t')\delta_{ij}. \quad (2.20)$$

It is worth noting that models without such restrictions can also be solved with IDP.

In the same spirit as Section 2.1, when $\vec{\lambda}$ is switched from its initial value $\vec{\lambda}_i$ to the final value $\vec{\lambda}_f$ in finite time, the input work to the system, W_{in} , can be formally expressed as

$$W_{in} = \int_{t_i}^{t_f} \dot{\vec{\lambda}} \cdot \langle \nabla_{\vec{\lambda}} V(\vec{x}, \vec{\lambda}) \rangle dt, \quad (2.21)$$

where $\nabla_{\vec{\lambda}}$ is the gradient with respect to $\vec{\lambda}$, and the brackets again denotes the average over the instant distribution function $g(\vec{x}, t)$ of the system. Besides, during the switching, the change of $\langle \vec{x} \rangle$ leads to an output work

$$W_{out} = \vec{f} \cdot \langle \vec{x} \rangle|_{t=t_i} - \vec{f} \cdot \langle \vec{x} \rangle|_{t=t_f}. \quad (2.22)$$

If $V(\vec{x}, \vec{\lambda})$ is a polynomial function up to the second order in \vec{x} , the number of differential constraints on the moments of $g(\vec{x}, t)$ would be finite, the OPMIW problem can be rigorously expressed as an IDP-olvable optimization problem.

To better illustrate the energetics of OPMIW, we only focus on the one-way switching processes of $\vec{\lambda}$ throughout this article. One can integrate these and other processes into a complete working cycle to construct a novel nano-machine, as illustrated in [13], the related optimization problem may need to be re-formulated. Besides, in present study, the systems are initially equilibrated under the potential function $V(\vec{x}, \vec{\lambda}_i)$. Consequently, the initial conditions of the differential constraints, *i.e.*, the initial values of the moments of $g(\vec{x}, t)$, are selected as the corresponding equilibrium values. However, both the initial and final values of the moments can be chosen arbitrarily in computation as long as the optimal protocol exists. Setting the final values of the moments of $g(\vec{x}, t)$ means that additional constraints are included in OPMIW problem.

2.3 Formulation for oligomer actuators

Under the general framework for systems with multiple degrees of freedom, we consider the one-dimensional oligomer actuators consisting of $N+1$ ($N=1,2,\dots$) beads in this

subsection. Instead of the linear oligomers without branches and cross-links, we consider a sub-family of structured actuators with the potential function

$$\begin{aligned} & V(x_1, x_2, \dots, x_N) \\ &= \frac{1}{2}x_1^2 + \frac{1}{2}\sum_{i=2}^N (x_i - x_{i-1})^2 + \frac{1}{2}\lambda x_N^2 - f x_N. \end{aligned} \quad (2.23)$$

Here, the beads are numbered from 0 to N , x_i denotes the position of the i th bead. While the adjacent beads are connected by a spring with elastic coefficient 1.0, x_N is connected to x_0 by an additional spring with tunable elastic coefficient λ . To study the energetic problems, x_0 is assumed to be fixed at the origin, a pulling force f is imposed on x_N . The actuators are driven by switching λ from the initial value λ_i to the final value λ_f in finite-time. The current model system is inspired by the recently designed DNA oligomer actuators [19,20]. In these DNA actuators, certain intra-molecular bonds can be modified by switching the light or chemical conditions, which results in considerable change of end-to-end distance (*i.e.*, a potential actuation process). Here, in the same spirit, a model actuator achieves an actuation when the 'intra-molecular bond' between bead 0 and bead N is tuned externally. Therefore, the $N = 1$ model is exactly the above-mentioned one-spring actuator in which the only bond is tuned, and thus only the last two energy terms in Eq. (2.23) need to be included in computation. The $N > 1$ models describe structured actuators with extra degrees of freedom, *e.g.*, the $N = 2$ actuator illustrated in Fig. 1(b).

In current article, overdamped Langevin equations are chosen to model the dynamics of the actuators. For $N = 2$ case, the optimal index, *i.e.*, the input work, W_{in} , is

$$W_{in} = \int_{t_i}^{t_f} \frac{1}{2} \dot{\lambda} \langle x_2^2 \rangle dt. \quad (2.24)$$

The differential constraints are

$$\langle \dot{x}_1^2 \rangle = -4\langle x_1^2 \rangle + 2\langle x_1 x_2 \rangle + 2, \quad (2.25)$$

$$\langle \dot{x}_1 x_2 \rangle = -(3 + \lambda)\langle x_1 x_2 \rangle + \langle x_1^2 \rangle + \langle x_2^2 \rangle + f\langle x_1 \rangle, \quad (2.26)$$

$$\langle \dot{x}_2^2 \rangle = -2(1 + \lambda)\langle x_2^2 \rangle + 2\langle x_1 x_2 \rangle + 2f\langle x_2 \rangle + 2, \quad (2.27)$$

$$\langle \dot{x}_1 \rangle = -2\langle x_1 \rangle + \langle x_2 \rangle, \quad (2.28)$$

$$\langle \dot{x}_2 \rangle = -(1 + \lambda)\langle x_2 \rangle + \langle x_1 \rangle + f. \quad (2.29)$$

For $N = 3$ case, W_{in} is

$$W_{in} = \int_{t_i}^{t_f} \frac{1}{2} \dot{\lambda} \langle x_3^2 \rangle dt. \quad (2.30)$$

The differential constraints are

$$\langle \dot{x}_1^2 \rangle = -4\langle x_1^2 \rangle + 2\langle x_1 x_2 \rangle + 2, \quad (2.31)$$

$$\langle \dot{x}_2^2 \rangle = -4\langle x_2^2 \rangle + 2\langle x_1 x_2 \rangle + 2\langle x_2 x_3 \rangle + 2, \quad (2.32)$$

$$\langle \dot{x}_3^2 \rangle = -2(1+\lambda)\langle x_3^2 \rangle + 2\langle x_2 x_3 \rangle + 2f\langle x_3 \rangle + 2, \quad (2.33)$$

$$\langle \dot{x}_1 x_2 \rangle = -4\langle x_1 x_2 \rangle + \langle x_2^2 \rangle + \langle x_1^2 \rangle + \langle x_1 x_3 \rangle, \quad (2.34)$$

$$\langle \dot{x}_2 x_3 \rangle = -(3+\lambda)\langle x_2 x_3 \rangle + \langle x_1 x_3 \rangle + \langle x_3^2 \rangle + \langle x_2^2 \rangle + f\langle x_2 \rangle, \quad (2.35)$$

$$\langle \dot{x}_1 x_3 \rangle = -(3+\lambda)\langle x_1 x_3 \rangle + \langle x_2 x_3 \rangle + \langle x_1 x_2 \rangle + f\langle x_1 \rangle, \quad (2.36)$$

$$\langle \dot{x}_1 \rangle = -2\langle x_1 \rangle + \langle x_2 \rangle, \quad (2.37)$$

$$\langle \dot{x}_2 \rangle = -2\langle x_2 \rangle + \langle x_1 \rangle + \langle x_3 \rangle, \quad (2.38)$$

$$\langle \dot{x}_3 \rangle = -(1+\lambda)\langle x_3 \rangle + \langle x_2 \rangle + f. \quad (2.39)$$

3 Numerical results and discussions

3.1 Numerical accuracy of IDP

To test the numerical accuracy of IDP, we first investigate the two examples that have been analytically solved by Schmiedl and Seifert [11]. Both of the two systems are one-dimensional systems satisfying overdamped Langevin dynamics. The two model potentials are $V_1(x, \lambda) = \frac{1}{2}(x - \lambda)^2$ (with tunable average position) and $V_2(x, \lambda) = \frac{1}{2}\lambda x^2$ (with tunable stiffness) respectively.

The comparison between the numerical results and the analytical solutions is shown in Fig. 2, where the potential functions, the initial and final values of λ , the switching time interval Δt and the MIW, W_{in} , are all listed. We only list one MIW value in either figure, since the numerical MIW is the same with the analytical MIW [Eq. (9) and Eq. (19) in [11]] up to the shown numerical accuracy. In either figure, the solid line is the analytical protocol of λ , and the hollow triangles denote the numerical results. Obviously, for the two examples studied here, the numerical results accurately follow the analytical curves. The important characteristics of the OPMIWs for the model systems, *i.e.*, the jumps both at the beginning and at the end of the optimal protocols, are successfully reproduced.

The triangles in Fig. 2 distribute non-uniformly along the time axis due to the employed method of variable step-length. In this scheme, the space between neighboring time points are also optimized. Hence, more time points will be gathered at the sharply changing parts of the λ -protocol, with less at the smoothly changing parts. As a result, higher accuracy can be obtained with limited discretization of $[t_i, t_f]$. Implementing the variable step-length method does not change the procedure of IDP (see the appendix).

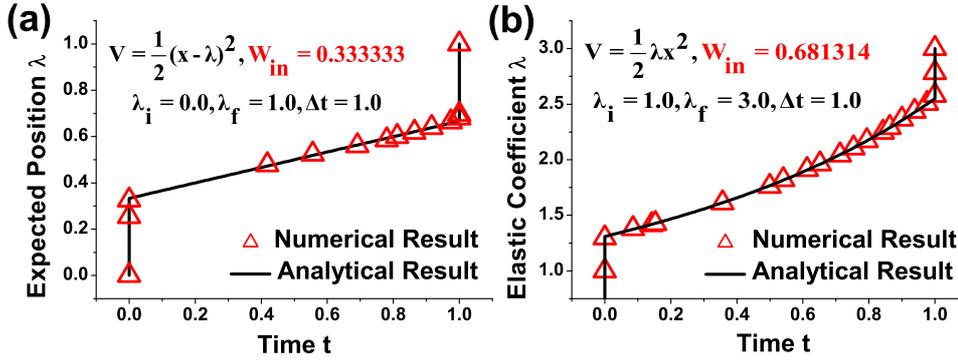


Figure 2: (color online). Comparison between numerical and analytical solutions [Eq.(9) and Eq.(19) in [11]] for overdamped systems with harmonic potentials. (a) Results for the system with tunable averaged position λ . (b) Results for the system with tunable elastic constant (spring stiffness) λ . The potential functions and related parameters are shown in the figure.

3.2 OPMIW for overdamped cases

In this subsection, we consider the overdamped one-spring actuator introduced in Section 2.1. Without the pulling force, this model turns into the analytical solvable model with tunable stiffness (see Section 3.1). The force complicates the problem, making it very hard to get an analytical solution. However, the model system is quite suitable for IDP. For all the processes studied in this subsection, λ is switched from 1.0 to 3.0, and the force is set as 1.0 if not further claimed.

We first solve the OPMIW problems for processes with different switching time intervals and the same λ_i , λ_f and f . Some of the OPMIWs are shown in Fig. 3(a). As can be seen, there are always jumps at the beginning and at the end of the optimal protocols, which is similar to the cases in the already studied systems [10, 13]. Intuitively, with infinite or large enough time interval, all the jumps should approach zero no matter how large the force is (or even whatever the potential function is), because the switching process is actually quasi-static or reversible. 'reversible' here means that the input work can be completely converted into the free energy of the system, without any heat dissipation. This is consistent with the second law of stochastic thermodynamics $W_{in} \geq \Delta F$, where ΔF is the free energy difference between the two equilibrium states that are determined by λ_i and λ_f respectively. This inequality holds for any protocols of control variables in a driven process including the OPMIW. To verify this, the MIWs for processes with different switching time are shown in Fig. 3(c). It can be seen that the MIW decreases monotonically with the increase of switching time, and asymptotically reaches ΔF when the switching time goes to infinity [the value of ΔF is indicated as the dashed line in Fig. 3(c)]. Consequently, the non-negative quantity $W_{in} - \Delta F$ can be taken as an indicator of the irreversibility of a driven process. In fact, it is exactly the total dissipation of the actuator and the environment [15]. In contrast to the input work, as shown in Fig. 3(c), the output mechanical work increases monotonically with the switching time and also

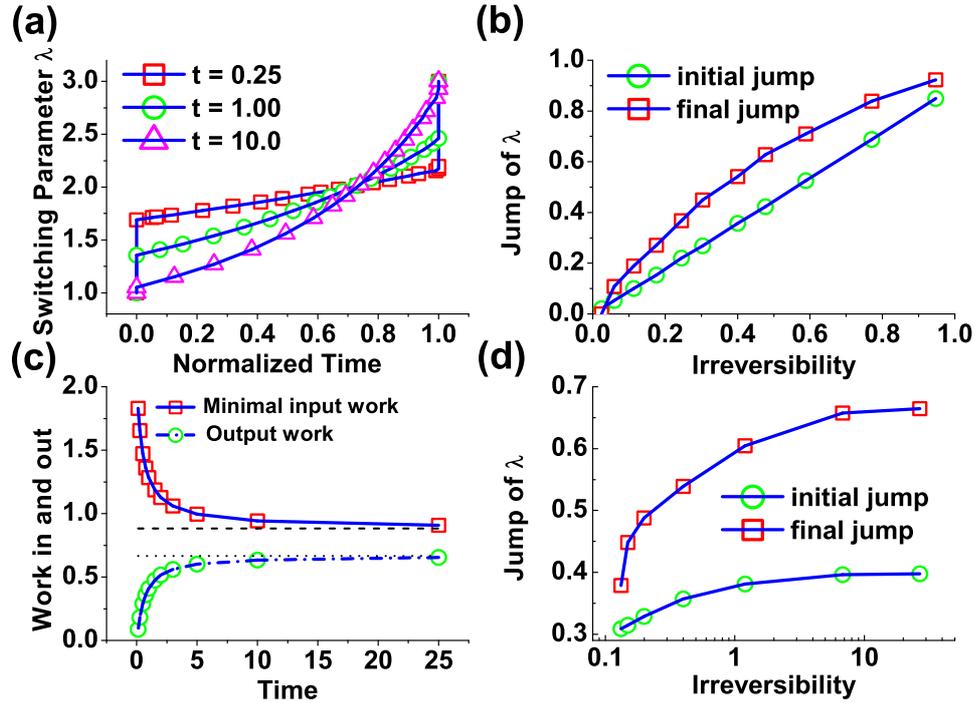


Figure 3: (color online). Numerical results for the overdamped one-spring actuator. (a) The optimal protocols with switching time 0.25 (squares), 1.00 (circles) and 10.0 (upward triangles). The switching time of the protocols have been re-scaled to $[0.0, 1.0]$ for better visualization. The stiffness of the spring, λ , is switched from 1.0 to 3.0. (b) Relation between the initial (circles) or final (squares) jump in OPMIW and the irreversibility, $W_{in} - \Delta F$. The data points correspond respectively to processes with switching times of 25.0, 10.0, 5.0, 3.0, 2.0, 1.5, 1.0, 0.75, 0.50, 0.25 and 0.10 (in the order of increasing irreversibility). (c) The minimal input works (squares) and the corresponding output works (circles) for processes with different switching time. The horizontal dashed line indicates the free energy difference between the two equilibrium states determined by λ_i and λ_f . The horizontal dotted line indicates the maximal output work of the system, which can only be realized with infinite switching time. (d) Relation between the initial or final jump in OPMIW and the irreversibility, $W_{in} - \Delta F$. The data points correspond respectively to processes with different pulling forces of 0, 0.25, 0.5, 1.0, 2.0, 5.0 and 10.0 (in the order of increasing irreversibility, the switching time is set as 1.0).

reaches an asymptotic value that is only determined by the initial and final equilibrium states. When the system finally reaches equilibrium, the change of average position $\langle x \rangle$ reaches its maximum with the largest output work. Thus, longer switching time results in a final state closer to equilibrium and a larger output work. The above results suggest that IDP can provide physically sound and numerically correct results for the OPMIW problems discussed here.

While the infinitely slow process without any jumps leads to the lower bound (zero) of dissipation, the upper bound of dissipation, *i.e.*, the highest irreversibility, occurs in the case of zero switching time (*i.e.*, λ is switched from λ_i to λ_f instantaneously). Between these two extremes are the processes with finite switching time. Thus, it is reasonable to expect that the shorter switching time corresponds to the larger jumps in OPMIW [see

Fig. 3(a)] due to the increased irreversibility. For switching processes with different time intervals, the relations between the magnitude of jumps (both of the initial and of the final jumps) and the irreversibility of the OPMIW are plotted in Fig. 3(b). The trends of these curves indeed conform to the above guess. We also notice that when the switching time approaches zero, *i.e.*, almost with the largest irreversibility, the OPMIW becomes stepwise and has very little advantage over the other control protocols: different λ -protocols result in almost the same input work since the switching is too fast to change the position of the bead.

Similarly, we also study the processes with the same switching time interval and various pulling forces. For current model system, larger force results in larger irreversibility, which, as expected, leads to larger jumps in the optimized λ -protocol [see Fig. 3(d)]. From Fig. 3(b) and (d), together with other results of changing the switching range of λ (data not shown) or the bead mass (the inertial effects will be discussed in the next subsection), we expect that the magnitude of either the initial or the final jump in OPMIW may always monotonically correlate to the irreversibility, no matter what the potential function is. Of course, this observation needs further validation in other systems.

3.3 OPMIW for underdamped cases

In this subsection, we study the underdamped one-spring actuator introduced in Section 2.1. Similar to the overdamped cases, the OPMIWs for the underdamped actuators also have initial and final jumps. Besides, for the system with tunable average position in Section 3.1, δ -function-like pulses have been analytically found in the optimized λ -protocol. Although the current model system can not be solved analytically, the δ -function-like pulses in OPMIW have also been inferred when the pulling force is zero [14]. For all the processes studied in this subsection, a constant force $f = 1.0$ is always loaded on the movable bead, λ is switched from 1.0 to 3.0 with switching time 1.0.

We first calculate the OPMIW for the one-spring actuator with mass $m = 0.02$. Three optimal protocols with different allowable range of $u \equiv \frac{d\lambda}{dt}$ are plotted in Fig. 4(a). In contrast to the overdamped cases, the optimized λ -protocols are no longer monotonic. There are always an upward peak at the beginning and a downward peak at the end of the OPMIW. The upward (downward) slope of either peak always matches the upper (lower) bound of u , indicating that the δ -function-like traces do exist in the OPMIW of current model system. The initial upward δ -peak enables an instant acceleration of the bead to ensure a nearly constant velocity during the process, and the final downward δ -peak enables an instant deceleration of the bead to ensure minimal dissipation [14]. Nevertheless, to exactly reproduce a δ -function trace of λ , the allowable range of u should be very large in computation, which is numerically hard (see the discussion in appendix) and is not the focus of this article.

To demonstrate the feasibility of IDP-based OPMIW study for underdamped systems, we turn to a subset of protocols in which λ grows monotonically with time. This subset of protocols might be easier to implement in practice. We calculate the OPMIWs for ac-

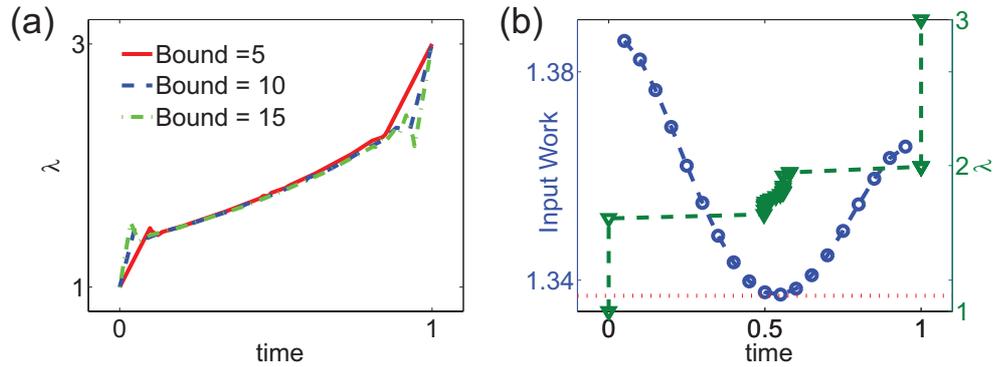


Figure 4: (color online). The OPMIWs of underdamped one-spring actuator. (a) The non-monotonic OPMIWs for the $m=0.02$ actuator with different bounds of $u \equiv |\frac{d\lambda}{dt}|$. The results obtained with the bound of 5 (solid line), 10 (dashed line) and 15 (dash-dotted line) are shown. (b) The input works for protocols with three jumps and two plateaus ($m=0.2$, see the main text for more details). The horizontal axis indicates the position of the intermediate jump. The left vertical axis shows the input work. The OPMIW (downward triangles) for this model is also shown for comparison. The right vertical axis shows the λ value. The dotted horizontal line indicates the MIW value for this model.

tuators with different bead mass m , with two OPMIWs ($m=0.2$ and $m=0.01$) illustrated in Fig. 5(a). The initial and final jump of λ value are found again in the OPMIWs. As expected, when m decreases, thus γ/m increases, the underdamped protocol will converge to the overdamped one. For finite m , however, plateaus appear immediately after the initial jump and before the final jump in an optimized λ -protocol. This phenomenon can be roughly explained as following. On one hand, after the first plateau stage, the bead has been accelerated to finite velocity. This velocity remains almost constant when λ starts to increase again until reaching the other plateau of λ -protocol. Therefore, from the perspective of minimal dissipation, the dissipation due to non-uniform velocity is minimized by including the first plateau stage. On the other hand, from the perspective of minimal input work, in the plateau stages with almost constant λ , no work is input into the system. Besides, in the final plateau stage, the established velocity of the bead is damped. Thus, the extra dissipation after switching, which would be payed by the input work, is reduced. In a word, the protocol between the initial and final jumps can be well interpreted either from the perspective of minimal dissipation or minimal input work. In the following, we will perform more detailed analysis on the numerical OPMIWs to provide in-depth perspective of inertial effect, as well as to further verify the correctness of the computational results.

As shown in Fig. 5(a), the OPMIW becomes stepwise when m increases. Taking the $m=0.2$ actuator as example, except for the initial and final jump, a narrow region with sharply increasing λ value exists approximately in the middle of the OPMIW. Actually, this short piece of increasing λ -protocol can be well approximated by a jump of λ value without prominent increase of input work. In contrast, the position of this narrow region is more prone to changing the input work. To illustrate this point, we compare differ-

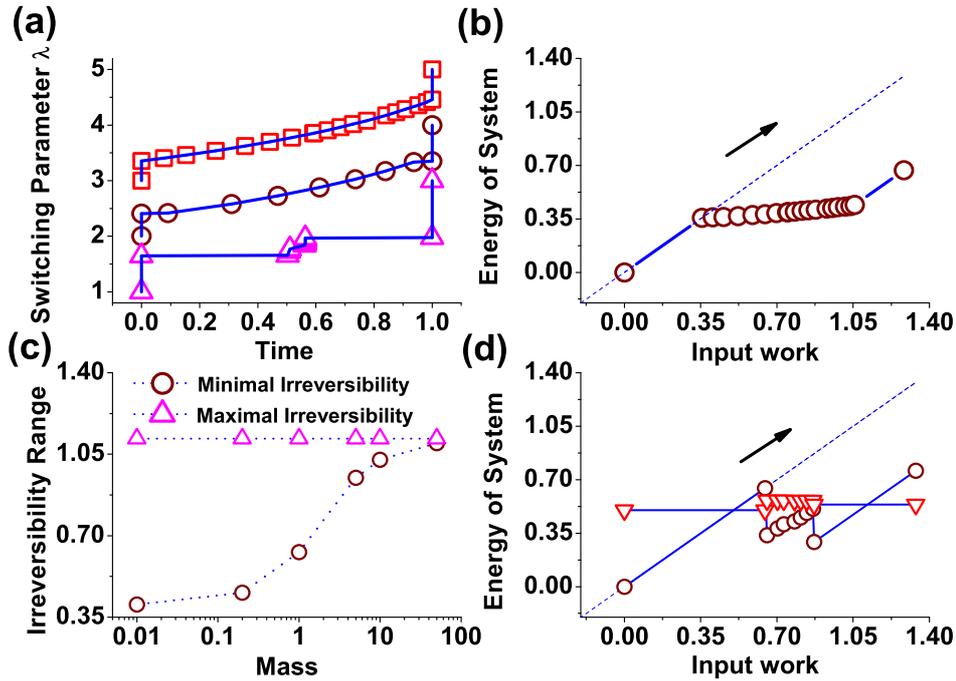


Figure 5: (color online). Numerical results for the underdamped one-spring actuators. (a) The OPMIW with different bead mass. The OPMIW for $m=0.01$ (hollow circles) and $m=0.2$ (hollow triangles) are shown. For comparison, the OPMIW for the overdamped case is also shown (hollow squares). The curves have been shifted for better visualization. The relations between the instant energy terms of the system and the cumulative input work are shown in (b) (overdamped case) and (d) ($m=0.2$), where the instant potential energy is denoted by hollow circles, the cumulative input work is denoted by dotted line, the kinetic energy is denoted by hollow downward triangles. The horizontal axis is the cumulative input work. The arrows in (b) and (d) indicate the direction of time evolution. (c) shows the variation range of irreversibility. The maximum and minimum irreversibility are plotted for different bead masses of 0.01, 0.5, 1.0, 5.0, 10.0 and 50.0.

ent stepwise protocols with three jumps and two plateau stages of λ value. The initial and final jumps of these protocols are set the same with the OPMIW of $m=0.2$ actuator (thus the magnitude of the intermediate jump is also determined), only the position of the intermediate jump is variable. For one-spring actuator with bead mass 0.2, the input works of these protocols are plotted in Fig. 4(b). As can be seen, the stepwise protocol with the smallest input work has its intermediate jump located within the intermediate increasing region of OPMIW, and its input work is only slightly larger than the MIW. This result strongly supports that the OPMIW obtained here is an optimum of the 'protocol sub-space' (*i.e.*, the space of monotonical λ -protocols). When m further increases, the input works of various λ -protocols will gradually converge to the same value. This can be seen from Fig. 5(c), where the possible variation range of irreversibility is shown for actuators with different mass. The variation range is defined as the maximal irreversibility minus the minimal irreversibility (or equivalently, the maximal input work minus the minimal input work). The former is obtained by instantaneous switching and is obvi-

ously identical for actuators with different mass, and the latter is obtained with OPMIW. Fig. 5(c) shows that the larger bead mass corresponds to the larger minimal irreversibility and thus the smaller variation range, which means that all the λ -protocols (including OPMIW) lead to nearly the same irreversibility and input work when m approaches infinity.

The inertial effect can also be perceived from the relation between input energy and system energy. For overdamped case, the potential energy of the system [*i.e.*, the instant ensemble average of Eq. (2.1)] increases monotonically with the cumulative input work as shown in Fig. 5(b). By contrast, as shown in Fig. 5(d), this relation is no longer monotonic for the underdamped cases (only the $m = 0.2$ case is shown here). In the first plateau stage of OPMIW, while no work is input, part of the potential energy transforms into kinetic energy and heat, which is indicated by the first downward jump on the potential energy curve and the first upward jump on the kinetic energy curve in Fig. 5(d). During the ascending stage in the middle of OPMIW, the kinetic energy remains almost constant, and the potential energy begins to increase with the cumulative input work. In the second plateau stage, both potential and kinetic energy are dissipated as heat, which is indicated by the second (downward) jump on the kinetic and potential energy curves. When m becomes larger, the second jump on either energy curve disappears first and then does the first jump (data not shown). When m approaches infinity, the bead is too heavy to be accelerated. Consequently, the whole system seems to be switched instantaneously from an equilibrium state to a non-equilibrium state. The input work is completely stored as potential energy and the kinetic energy remains unchanged.

3.4 Oligomer actuator: OPMIW for more degrees of freedom

In this subsection, we will investigate how the internal degrees of freedom can change the OPMIW and performance of an actuator. Concretely speaking, we try to study an optimal design problem, *i.e.*, can we construct a more efficient actuator that output the specified amount of mechanical work with least input work? To explore this question, we study the model systems introduced in Section 2.3, with N equals to 1, 2 and 3 respectively. In a fast enough actuation process, an actuator might be driven out of equilibrium, thus its internal structure may begin to substantially affect the performance of the energetics, which leaves room for optimal design.

Now, we can explicitly address the optimization problem. With the same variation range of $\lambda \in [\lambda_i, \lambda_f]$, the same time interval $t \in [t_i, t_f]$ and identical pulling force, as well as an extra requirement of equal amount of output work (*i.e.*, rated output work), the MIW values for different structured actuators can be compared. The one with the lowest MIW should outweigh the others. For illustration, we only consider the overdamped $N = 1, 2, 3$ models. In all the examples, λ is switched from 1.0 to 3.0, the switching time interval is $[0.0, 1.0]$, the pulling force is set as 1.0, and the rated output work is chosen as 0.3.

The OPMIWs for the $N = 1, 2, 3$ models are plotted in Fig. 6(a). While there are both initial and final jumps in the OPMIWs of the $N = 1$ and $N = 3$ models, the final jump in the OPMIW of the $N = 2$ actuator disappears, and a plateau region appears instead.

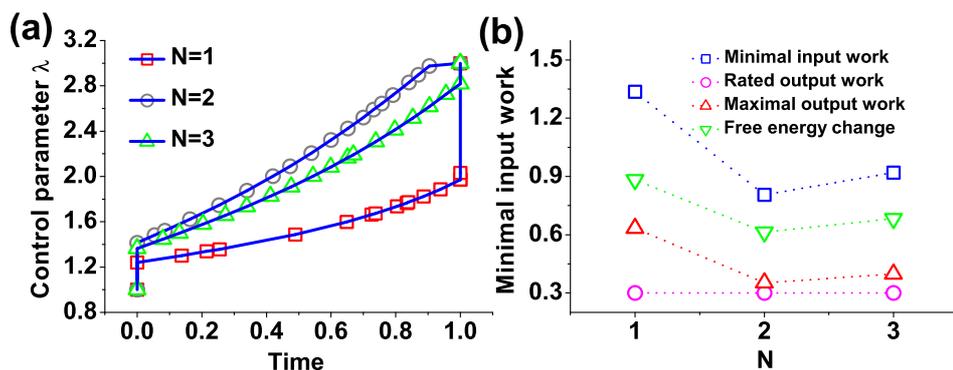


Figure 6: (color online). Numerical results for oligomer actuators. (a) The OPMIWs for the $N=1$ (squares), $N=2$ (circles) and $N=3$ (upward triangles) models. The switching time is 1.0, and λ is switched from 1.0 to 3.0. (b) The minimal input works (squares), the maximal output works in finite-time (upward triangles), the rated output work (circles), and the free energy differences (downward triangles) for the $N=1$, $N=2$ and $N=3$ actuators.

This is a consequence of constraining the output work. As shown in Fig. 6(b), among the three models considered here, the $N=2$ actuator possesses a maximal finite-time output work which is the closest to the rated output work. Since this maximal output work is realized by instantaneously switching λ from λ_i to λ_f at the beginning of the time interval (which is an infinitely fast process), the switching process of the $N=2$ actuator may also be relatively fast to realize the rated output work. Consequently, in contrast to protocols with final jump, λ_f is reached before t_f in the OPMIW of $N=2$ actuator. Actually, for any model system studied in this subsection, the OPMIW will become one-step-like when the rated output work approaches the maximal finite-time output work.

In spite of the fast switching OPMIW, the MIW of the $N=2$ model turns out to be the smallest among the three models [see Fig. 6(b)]. In other words, the $N=2$ actuator performs the best when generating a finite-time output work of 0.3, which demonstrates the possibility for an optimal design of oligomer actuators. Interestingly, the $N=2$ model also has the smallest irreversibility among the three models [see the gap between MIW and free energy change in Fig. 6(b)]. This observation inspires us to propose an premature conjecture that irreversibility and performance might be related for actuators with similar construction and different number of degrees of freedom. If such a relation does exist, the above discussion about optimal design could be generalized no matter the intra-molecular interaction is harmonic or not (for a real macromolecule, the intramolecular interaction is usually not harmonic), since it is the irreversibility that determines the energetic performance of an actuation device.

4 Conclusion and perspective

In current article, we study the OPMIW problems by introducing a numerical method, iterative dynamic programming (IDP). We first apply IDP to the systems with one degree

of freedom and simple potential functions. Both the overdamped and the underdamped systems are investigated. For these two situations, jumps of the control variable are always found both at the beginning and at the end of OPMIW. For the underdamped case, the δ -function-like traces in OPMIW are also found. With vanishing mass, the underdamped models indeed converge to the overdamped one, and with increasing mass, the OPMIW of the underdamped models become stepwise. However, with quite large mass, the OPMIW only marginally outweigh the other protocols in terms of the input work, which is similar to other cases (such as vanishing switching time) with large irreversibility.

Furthermore, we use IDP to study the optimal control and optimal design problems for the systems with multiple degrees of freedom. We consider a family of toy oligomer actuators in which the interaction between the two ending monomers can be tuned externally. The energetic performance of such actuators with different monomer number are compared, in the sense of outputting the same amount of mechanical work with as-least-as-possible input work. Our results indicate that properly choosing the number of degrees of freedom can indeed bring better energetic performance.

The above logic can be generalized to systems with more complex internal structures, *e.g.*, a spring network with different harmonic interactions between the bead pairs. This construction is in the same spirit with the Gaussian Network Model (GNM) widely used to describe the thermal fluctuations of folded proteins [21] or the Elastic Network Model (ENM) used to describe the large-scale motion of motor proteins [22]. While those works focus on the structure-function and dynamics-function relations of protein machines, our studies might suggest a new perspective on energetics-structure relations. For instance, if the binding or release of substrates can be understood as tuning the local interaction between the residues of the motor proteins, energetic optimization problems similar to those studied here could also be raised. To carry out such investigation, however, all the thermodynamic relations and optimization problems should be reformulated for chemically-driven protein machines. The stochastic thermodynamics of chemical reaction networks [25] and the structural details of the macro-molecules should be combined, leading to a possibly formidable framework (one potential candidate is the generic ratchet model of chemically-driven machines [3]) either for analytical or for numerical treatments. Therefore, at this preliminary stage, it may be beneficial to first construct some simple toy models to mimic the nano-machines. The toy actuators studied in this article can serve as the basic elements.

Acknowledgments

This work is supported by National Basic Research Program of China (973 Program) under the grant No. 2007CB935903 and NSFC grant No. 10847165.

A Iterative dynamic programming

In the main text, we have presented the numerical results of IDP for various OPMIW problems. Here we provide a brief introduction to IDP. More details can be found in [18].

Generally speaking, IDP can solve the optimal control problems of following style.

$$J = \int_{t_i}^{t_f} L(\vec{s}, \vec{u}, t) dt, \quad (\text{A.1})$$

$$\dot{\vec{s}} = r(\vec{s}, \vec{u}, t), \quad (\text{A.2})$$

where L is a function of the state variables \vec{s} , the control variables \vec{u} and time t . Since \vec{s} are determined by \vec{u} through the differential constraints Eq. (A.2), J is simply an objective functional (also named as optimization index) of the protocol $\vec{u}(t)$. The related optimization problem is to find the optimal trajectory $\vec{u}(t)$ that minimizes or maximizes J . Once the initial condition of the state variables and the initial time t_i are specified, the problem is well formulated. In principle, it is also possible to include more constraints on the final values of the state variables, the final time t_f and the control variables. Conventionally, this kind of problems is solved with Pontryagin principle [23], though it is usually hard to obtain analytical solutions. The dynamic programming method (DP) [24] is a conceptually sound numerical method for the optimal control problems. However, it also suffers from the difficulty of numerical interpolation and the restriction on dimensionality. In view of this, Luus developed IDP method to complement DP. With dramatic enhancement of speed, IDP is still promising for finding the global minimum, and achieves good numerical precision as shown in current article and also in Ref. [18]. Here, we only describe the main procedures of IDP, readers can learn the underlying principles from [18].

By introducing an additional state variable s^* satisfying

$$s^* = L(\vec{s}, \vec{u}, t), \quad (\text{A.3})$$

$$s^*(t_i) = 0, \quad (\text{A.4})$$

$$s^*(t_f) = J, \quad (\text{A.5})$$

the problem can be transformed into a set of differential equations shown in Eq. (A.6) to Eq. (A.8):

$$\dot{s}^* = L(\vec{s}, \vec{u}, t), \quad (\text{A.6})$$

$$\dot{\vec{s}} = r(\vec{s}, \vec{u}, t), \quad (\text{A.7})$$

$$J = s^*(t_f). \quad (\text{A.8})$$

As a result, the optimization index J is determined solely by the state variables at time t_f .

A.1 The procedure of IDP

The computational procedure of IDP is described in this subsection.

Before calculation, the time interval, $[t_i, t_f]$, should be discretized into P stages with identical length l , *i.e.*, $Pl = t_f - t_i$, with constant control variables in each stage. Thus, the protocol of control variables can be represented by a P -dimensional array, each element of the array stores the values of control variables in the corresponding time stage. In each time stage, an allowable range of control variables should be specified. The allowable range does not change in one cycle of computation (the definition of a computation cycle will be given below).

A reference P -dimensional array, U , of control variables pre-exists to initiate each cycle. The elements of U are denoted as U_i ($i = 1, 2, \dots, P$). The allowable range of control variables is determined by U , together with the size of the range, R . For the case of one-dimensional control, the allowable range in the i th time stage could be $[U_i - R, U_i + R]$. Of course, it is not necessary to set the reference control at the center of the allowable range. To start the first computation cycle, it is necessary to specify an initial P -dimensional array as the reference and also the initial size R of the allowable range. It should be noted that R could be either independent or dependent on time. We only discuss the former case here, since no substantial difference exists between the two cases.

In calculation, a matrix of state variables should be first constructed as the mesh grid for dynamic programming. For simplicity, we only discuss the case of one-dimensional control in the following.

First, The allowable interval of control variable at each time stage is uniformly divided into $N_s - 1$ parts, with the N_s nodes of this partition taken as the allowable values of control. All the largest allowable values in each time stage are grouped as the first P -dimensional array of control (*i.e.*, the first control protocol). The second largest ones are grouped as the second P -dimensional array, etc. Finally, N_s P -dimensional arrays of control variable (N_s control protocols) are obtained. Suppose the arrays are labeled with m , the elements of one array are labeled with n , then all these control values can be put into a $N_s \times P$ matrix, with its elements denoted as u_{mn} .

Given the matrix of control variable, we can begin to construct the matrix of state variables. The idea is to use each P -dimensional array as the control protocol to calculate the corresponding state variables in different time stages. For the arbitrarily selected j th array, the calculation starts from the initial state S_0 , which is determined by the initial conditions. An updated state S_{j1} can be obtained by taking u_{j1} as the control variable and integrating Eq. (A.6) and Eq. (A.7) from t_i to $t_i + l$. S_{j1} is stored as an element into the matrix of state variables. After that, starting from S_{j1} , another updated state S_{j2} can be obtained by taking u_{j2} as the control variable and integrating from $t_i + l$ to $t_i + 2l$. With similar procedure, the j th row of the matrix can be constructed by integrating along the j th control protocol. After all the integrations are finished, we get a $N_s \times P - 1$ -dimensional matrix of the state variables as shown in Fig. 7. Subsequently, the optimal control value for each element of the matrix can be estimated by the backward iteration method of dynamic programming.

The backward iteration starts from the states S_{iP-1} ($i = 1, 2, \dots, N_s$), *i.e.*, the states at time $t_f - l$. T trial values of control variable should be provided *in priori*. Starting from

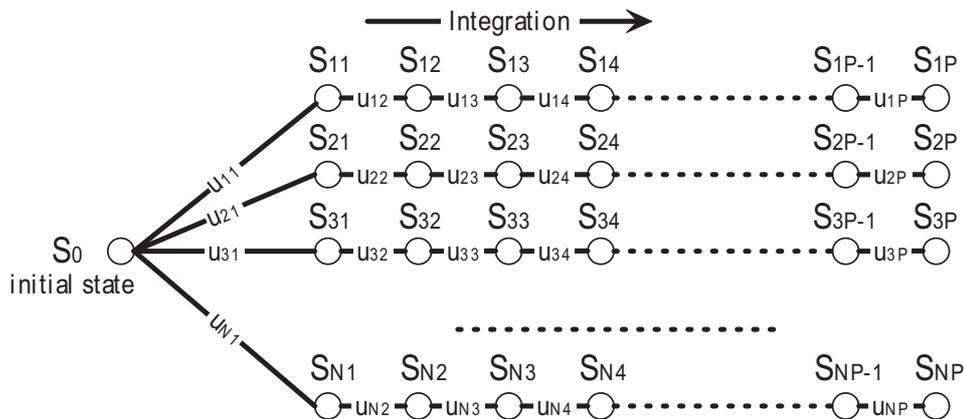


Figure 7: The process for building the matrix of state variables.

certain S_{iP-1} , taking respectively the T trial values as the control variable and integrating Eq. (A.6) and Eq. (A.7) from $t_f - l$ to t_f , we get T states, S_{iP-1}^j ($j=1,2,\dots,T$), at $t=t_f$, which correspond to different J values. The trial control value leading to the minimal J (here we only consider the minimization problems) is chosen as the optimal control variable for S_{iP-1} , and will be stored together with S_{iP-1} .

Subsequently, for each S_{iP-2} ($i=1,2,\dots,N_s$), by one round of integration from $t_f - 2l$ to $t_f - l$, T states, S_{iP-2}^j ($j=1,2,\dots,T$), at $t=t_f - l$ are generated. To proceed, we need to start from these states and integrate from $t_f - l$ to t_f . Then, since the resulted T states are at $t=t_f$, their corresponding J values can be compared to judge the optimal control value for S_{iP-2} . Similarly with the preceding paragraph, the trial control value leading to the minimal J should be selected as the optimal control of S_{iP-2} . The control values of S_{iP-2}^j ($j=1,2,\dots,T$) can be determined by the aid of the matrix of state variables. The key is to find the nearest state to S_{iP-2}^j among S_{iP-1} ($i=1,2,\dots,N_s$). If the nearest state to S_{iP-2}^j is S_{mP-1} , the control value of S_{iP-2}^j should be set equal to the optimal control of S_{mP-1} .

The same procedure applies to other elements S_{ij} ($i=1,2,\dots,N_s; j=P-3,P-4,\dots,1$) of the matrix of state variables. One cycle is finished while the above procedure has been performed for S_0 . After a cycle, we get the matrix with renewed optimal control values u_{ij}^* ($i=1,2,\dots,N_s; j=1,2,\dots,P-1$) for states S_{ij} ($i=1,2,\dots,N_s; j=1,2,\dots,P-1$), as well as the optimal control value u_0^* for S_0 .

Before the next cycle, it is necessary to refresh the reference array of control variable. The procedure is as following. Starting from S_0 , Eq. (A.6) and Eq. (A.7) are first integrated from t_i to $t_i + l$ with the optimal control u_0^* , with a new state S_0^* obtained. u_0^* is stored as the first element of a P -dimensional array. Then, within the states S_{1j} ($j=1,2,\dots,N_s$), the one nearest to S_0^* is picked out. Suppose the nearest one is S_{1m} , starting from S_0^* , and integrating Eq. (A.6) and Eq. (A.7) from $t_i + l$ to $t_i + 2l$ with the optimal control value u_{1m}^* of S_{1m} , we get a new state S_1^* . Once again, u_{1m}^* is store in the P -dimensional array as the

second element. A P -dimensional array of control variable can be generated by repeating the above procedure step by step. This protocol will be taken as the new reference array of control variable. Besides, the size of the allowable range, R , of control variable should be contracted before the new cycle. The new reference array and R together determine the allowable range of control in the next cycle.

A pass is constituted by prescribed number of cycles. Several passes are included in each calculation. At the beginning of a new pass, R is restored partially to its initial value, and the optimal control protocol generated in last pass is used as the initial reference array of control variable.

A.2 Variable step-length IDP

The IDP algorithm with variable step-length is almost the same with normal IDP method. Only slight re-formulation is needed. Here the time interval $[t_i, t_f]$ is scaled to $[0, 1]$. As in normal IDP method, the $[0, 1]$ interval is now uniformly divided into P segments. Furthermore, each segment i corresponds to a real time length of v_i , and the real time length in each segment is introduced into IDP algorithm as an extra control variable. Due to the scaling of time, Eq. (A.6) to Eq. (A.8) for the i th time segment should be expressed as below,

$$\frac{ds^*}{dt'} = L(\vec{s}, \vec{u}, t') v'_i, \quad (\text{A.9})$$

$$\frac{d\vec{s}}{dt'} = r(\vec{s}, \vec{u}, t') \frac{dt}{dt'} = r(\vec{s}, \vec{u}, t') v'_i, \quad (\text{A.10})$$

$$J = s^*(1), \quad (\text{A.11})$$

where v'_i satisfies

$$v'_i \equiv \frac{dt}{dt'} = v_i P, \quad (\text{A.12})$$

t' is the scaled time. The sum for all v_i should be equal to $t_f - t_i$. This constraint can be incorporated by substituting the origin optimization index J by a new one I , which is the sum of J and a penalty function for violating the constraint. For example, I could be chosen as below

$$I = J + \theta (\sum_i v_i - t_f + t_i)^2, \quad (\text{A.13})$$

where θ is a positive constant, v_i is the value of v at the i th time segment. The other kinds of constraints can be implemented in similar way.

B Computation details

As pointed out in the main text, the time derivative of λ -protocol is expressed as a temporal function u under current formulation. When searching for the OPMIW, u should be confined within a finite interval at all time. Consequently, larger range of u is necessary

to describe the abrupt changing of λ , *e.g.*, a noncontinuous jump or a δ -function trace of λ value. However, if in a preliminary calculation the optimized λ -protocol is found to be monotonic, and some parts of the protocol are jump-like (for example, the u values in these parts are always equal to the allowable upper or lower bound), it is possible to accurately estimate the OPMIW in the following way. Instead of $u = \frac{d\lambda}{dt}$, we can consider the reversed case, $u' = \frac{dt}{d\lambda}$, in calculation because of the monotonicity of t versus λ , and re-formulate all the functions of t to functions of λ . Since there is usually no need to specify large range for u' , the calculation is simplified. This situation happens to be true for all the overdamped examples studied in current article. Because the initial calculation without restriction almost ensures the monotonicity of λ , the above-mentioned scheme should be promising for finding the global minimum. For demonstration, the re-formulated Eqs. (2.11), (2.12), (2.8) and (2.9) of the overdamped one-spring actuator are shown below,

$$\frac{dW_{in}}{d\lambda} = \frac{1}{2}\langle x^2 \rangle, \quad (\text{B.1})$$

$$\frac{dt}{d\lambda} = u', \quad (\text{B.2})$$

$$\frac{d\langle x \rangle}{d\lambda} = fu' - \lambda \langle x \rangle u', \quad (\text{B.3})$$

$$\frac{d\langle x^2 \rangle}{d\lambda} = 2u' - 2\lambda \langle x^2 \rangle u' + 2f \langle x \rangle u'. \quad (\text{B.4})$$

It is worth noting that although we adopt the formulation with explicitly expressed time derivative of λ , the other strategies that may be more natural for noncontinuous λ -protocols, *e.g.*, approximating the λ -protocol with a stepwise function, can also be implemented with IDP.

In current article, we usually study each model (and parameter settings) with two IDP calculations. In the first calculation, the initial reference protocol is selected as a straight line connecting the two points, (t_i, λ_i) and (t_f, λ_f) (with constant $u \equiv \frac{d\lambda}{dt}$ or $u' \equiv \frac{dt}{d\lambda}$). After that, the resulted control protocol is selected as the initial reference in the second calculation. According to our experience, this procedure can ensure the successful application of IDP in all the examples here.

In computation, we apply the Runge-Kutta method in Ref. [26] for integration. The error tolerance in integration is set as 10^{-9} for the two examples in Section 3.1, and 10^{-7} for the others. According to our experience, the MIW calculated with 10^{-9} tolerance are precise up to the seventh digit after the dot, the ones calculated with 10^{-7} tolerance are precise up to the fifth digit after the dot. Usually, the number of stage, P , is selected within the range of [15,20]. In the matrix of state variables, the number of state at each time point is selected within [10,20]. The number of trials along each dimension of control variable is about 10, leading to around 100 trials for two-dimensional control variables (including $u \equiv \frac{d\lambda}{dt}$ and the length of time steps). When new cycle begins, the contraction factor for the range of control variables is selected as 0.92. When new pass begins, the

restoration factor for the range is selected as 0.88. Most of the calculations can be finished within 3 hours by single-thread computation with Intel(R) Q6700 CPU.

References

- [1] T. J. Huang and B. K. Juluri, Biological and biomimetic molecular machines, *Nanomedicine*, 3 (2008), 107-124.
- [2] E. R. Kay, D. A. Leigh, and F. Zerbetto, Synthetic molecular motors and mechanical machines, *Angew. Chem. Int. Ed.*, 46 (2007), 72-191.
- [3] A. Parmeggiani, F. Julicher, A. Ajdari, and J. Prost, Energy transduction of isothermal ratchets: generic aspects and specific examples close to and far from equilibrium, *Phys. Rev. E*, 60 (1999), 2127-2140.
- [4] A. B. Kolomeisky and M. E. Fisher, Molecular motors: a theorist's perspective, *Annu. Rev. Phys. Chem.*, 58 (2007), 675-695.
- [5] H. Wang and G. Oster, The stokes efficiency for molecular motors and its applications, *Europhys. Lett.*, 57 (2002), 134-140.
- [6] J. S. Bader, R. W. Hammond, S. A. Henck, M. W. Deem, G. A. McDermott, J. M. Bustillo, J. W. Simpson, G. T. Mulhern, and J. M. Rothberg, DNA transport by a micromachined Brownian ratchet device, *Proc. Natl. Acad. Sci. USA*, 23 (1999), 13165-13169.
- [7] D. Suzuki and T. Munakata, Rectification efficiency and Brownian motor, *Phys. Rev. E*, 68 (2003), 021906.
- [8] B. Andresen, P. Salamon and R. S. Berry, Thermodynamics in finite time, *Phys. Today*, 9 (1984), 62-70.
- [9] P. Salamon and R. S. Berry, Thermodynamic length and dissipated availability, *Phys. Rev. Lett.*, 51 (1983), 1127-1130.
- [10] H. Then and A. Engel, Computing the optimal protocol for finite-time processes in stochastic thermodynamics, *Phys. Rev. E*, 77 (2008), 041105.
- [11] T. Schmiedl and U. Seifert, Optimal finite-time Processes in stochastic thermodynamics, *Phys. Rev. Lett.*, 98 (2007), 108301.
- [12] I. Derenyi, M. Bier, R. D. Astumian, Generalized efficiency and its application to microscopic engines, *Phys. Rev. Lett.*, 83 (1999), 903-906.
- [13] T. Schmiedl and U. Seifert, Efficiency at maximum power: An analytically solvable model for stochastic heat engines, *Europhys. Lett.*, 81 (2008), 20003.
- [14] A. Gomez-Marin, T. Schmiedl and U. Seifert, Optimal protocols for minimal work processes in underdamped stochastic thermodynamics, *J. Chem. Phys.*, 129 (2008), 024114.
- [15] K. Sekimoto and S. Sasa, Complementarity relation for irreversible process derived from stochastic energetics, *arXiv:cond-mat/9708032* (1997).
- [16] K. Sekimoto, Langevin equation and thermodynamics, *Prog. Theor. Phys. Suppl.*, 130 (1998), 17-27.
- [17] U. Seifert, Entropy production along a stochastic trajectory and an integral fluctuation theorem, *Phys. Rev. Lett.* 95 (2005), 040602.
- [18] R. Luus, *Iterative Dynamic Programming*, Chapman and Hall/CRC, London, 2000.
- [19] H. Liu, Y. Xu, F. Li, Y. Yang, W. Wang, Y. Song and D. Liu, Light-driven conformational switch of i-motif DNA, *Angew. Chem. Int. Ed.*, 46 (2007), 2515-2517.
- [20] D. Liu and S. Balasubramanian, A proton-fuelled DNA nanomachine, *Angew. Chem. Int. Ed.*, 42 (2003), 5734-5736.

- [21] T. Haliloglu, I. Bahar and B. Erman, Gaussian dynamics of folded proteins, *Phys. Rev. Lett.*, 79 (1997), 3090-3093.
- [22] W. Zheng and S. Doniach, A comparative study of motor-protein motions by using a simple elastic-network model, *Proc. Natl. Acad. Sci. USA*, 100 (2003) 13253-13258.
- [23] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, New York, 1962.
- [24] R. Bellman, S. Dreyfus. *Applied dynamic programming*, Princeton University Press, 1962.
- [25] T. Schmiedl and U. Seifert, Stochastic thermodynamics of chemical reaction networks, *J. Chem. Phys.*, 126 (2007), 044101.
- [26] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007.