University of Zurich[UZH]

Zurich Open Repository and Archive
University of Zurich
Main Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2011

Finding structure in deadtime

Li, Tao ; Nakajima, Kohei ; Cianchetti, Matteo

Abstract: The dynamical coupling of the brain, the body, and the environment is essential to intelligent behaviors. However, we discuss the fact that most current robots still lack this coupling. We propose a methodology to realize such a coupling and demonstrate it in a soft robot experiment platform by releasing the deadtime of the high level controller. Some preliminary results, such as the splitting of the return map and the invariance of average errors, are reported and discussed.

Posted at the Zurich Open Repository and Archive, University of Zurich
ZORA URL: https://doi.org/10.5167/uzh-55688
Conference or Workshop Item
Published Version

Originally published at:
Li, Tao; Nakajima, Kohei; Cianchetti, Matteo (2011). Finding structure in deadtime. In: The 2nd International Conference on Morphological Computation (ICMC2011), Venice, 12 September 2011 - 14 September 2011, 48-50.

Finding Structure in Deadtime

Tao Li
Department of Informatics
University of Zurich
8050 Zurich, Switzerland
Email: taoli@ifi.uzh.ch

Kohei Nakajima
Department of Informatics
University of Zurich
8050 Zurich, Switzerland
Email: nakajima@ifi.uzh.ch

Matteo Cianchetti
The BioRobotics Institute
Scuola Superiore Sant'Anna
56026 - Pontedera, Italy
Email: m.cianchetti@sssup.it

***Abstract*—The dynamical coupling of the brain, the body, and the environment is essential to intelligent behaviors. However, we discuss the fact that most current robots still lack this coupling. We propose a methodology to realize such a coupling and demonstrate it in a soft robot experiment platform by releasing the deadtime of the high level controller. Some preliminary results, such as the splitting of the return map and the invariance of average errors, are reported and discussed.**

***Keywords*—embodiment, soft robotics, timing**

I. Introduction

Recent embodied intelligence research has revealed the importance of the reciprocal and dynamical coupling among the brain (control), body, and environment [1]. However, the coupling in the current robots still tends to be static. For example, it is often reported that living creatures adopt the division of functionality between the central and peripheral nervous systems as a simplification strategy for their behavioral control. Accordingly, robot designers often mimic this strategy by constructing corresponding modules, such as high and low level controllers, separately in a self-consistent manner. Although they are connected through the input-output interface, the function of high and low level controllers can still be defined independently, which means that they are not coupled at all. Our aim in this paper is to present a methodology to generate a mutual coupling between the high and low level controllers in a physical soft robot platform without any additional hardware.

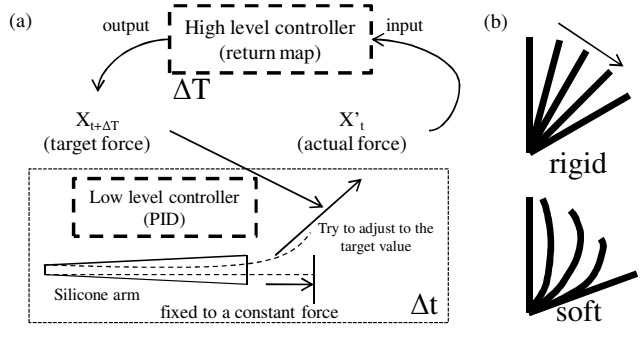

Fig. 1. (a) Schematic diagram explaining the relation between the high and low level controller through the processing time. See texts for details. (b) Soft materials take more time to relax compared with rigid materials. The two figures show the successive steps involved in rotating a rigid and a soft arm at the same angle velocity around a base, respectively.

In order to achieve this mutual coupling, we start by looking at the role of deadtime, the time step defined in the high level controller. In control systems, deadtime is a time delay from the time a control command is issued until the state variable begins to respond. The minimum deadtime is the time step for the controller. The high level controller step size, ΔT, is used as the deadtime for simplification in this study. For example, let us assume a simple closed-loop control system, as shown in Fig. 1(a), whose high level controller is equipped with a return map and low level controller is equipped with a PID controller. The return map takes the current state of the system as an input and outputs the target state, and its computational process is governed by ΔT as a unit time without finer time resolution. Meanwhile, the PID controller tries to tune the system state to the target state, and its computational process is governed by Δt which is smaller than ΔT. When looking into the construction of both controllers, we can find that each controller is governed by its own "clock", namely its time step size. Here, in order to successfully achieve the target using a high level controller, the low level PID controller needs a minimum time duration, $\Delta T_{threshold}$. Apparently, if $\Delta T_{threshold} < \Delta T$, the computational process for the high level controller is guaranteed. Otherwise, the input for the high level controller includes the perturbations originated from the error of the PID controller. Usually, $\Delta T_{threshold}$ is called the settling time of the PID controller. The settling time contains the relaxation process of the state to the target state and is governed by the parameters of the PID controller and also by the material property of the controlled system and its coupling with the environment. For example, if the controlled plant is a soft body, it would take much a longer time to relax compared with a rigid body [2], as shown in Fig. 1(b). In this paper, we will vary ΔT in incremental steps around this $\Delta T_{threshold}$. As explained above, if $\Delta T_{threshold} > \Delta T$, the input for the high level controller includes the perturbations originated by the error of the PID controller, which means that the high level controller would be affected and modulated by the implicit effect of the material properties of the controlled plant.

In the following sections, our experimental setup and conditions are explained, and the preliminary results are presented. Finally, we discuss the implication of the presented strategy for the biological system and future work.

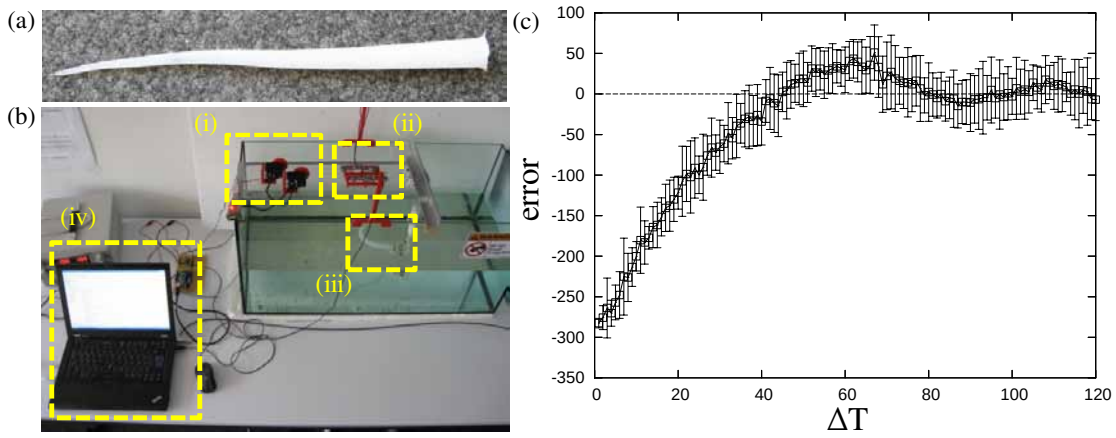

Fig. 2. (a) The silicone arm used in the experiment. (b) The experiment platform used in this paper. It is equipped with servo motors (i), force sensors (ii), a silicone arm (iii), and controllers (running on a laptop computer) (iv). (c) PID response curve in the current PID parameter setting. It was generated by taking the average and standard divisions of 10 independent response curves under the same setting. The initial force is set to 0, and the target force is set to 300.

II. Experiment Setting

To test the effect of the deadtime of the high level controller, we built an experiment platform to control a soft robot arm. Its setup, control algorithm, and experimental procedure are presented in this section.

A. Platform setup and control algorithm

The platform setup is shown in Fig. 2(b). It consists of a soft robotic arm, its actuation, sensing, and control system, and a water tank containing fresh water as the underwater environment. The soft robotic arm is constructed of commercially available silicone rubber (ECOFLEX™ 00-30). Using servo motors and force sensors, the soft arm is driven by regulating the tensions on cables embedded eccentrically along the arm [3]. Servo motors (Dynamixel™ AX-12A+) are used to provide and modulate cable tensions, which are measured by force sensors (KD24S from ME-Meβsystem GmbH). The force sensor signal is amplified and sent to a computer serial port through an Arduino™ UNO board, whose ADC outputs integer values between 0 and 1024 and corresponds linearly to forces in the range of 0 to 10N.

The control program running on a laptop computer consists of high and low levels, with time steps ΔT and Δt, respectively. ΔT is set much larger, tens to hundreds of times more than Δt and is fixed during each individual test. With the actual tension from the force sensor at that time as input, the high level controller sets the target cable tensions X'_T at the beginning of each high level time step ΔT by using the logistic map:

$$x_{n+1} = r x_n (1 - x_n), \tag{1}$$

where, x_n and x_{n+1} are the state variables of the current step and the next step, respectively. r is set to 4 to get a chaotic map. One exception is the target force for the first time step ΔT, whose setting is presented in detail in the next subsection. Since both the input and output of the logistic map are limited to the range of 0 to 1, the actual tension from the force sensor is linearly mapped to this range to be used as the logistic map input. And the output of the logistic map is linearly mapped back to the actual tension range, from 50 to 300, and used as the next target tension, X'_{t+T}. To ensure that the actual tension is correctly mapped to the range of x_n, we set a boundary: if the actual force is lower than or equal to 50, it is replaced by 51; if it is larger than or equal to 300, it is replaced by 299. And the low level is an standard PID controller, which aims to adjust the cable tension to the target X'_t during each high level time step ΔT , by adjusting motor rotation angle θ_t at each low level time step Δt. The equations of the PID controller are as follows:

$$e_t = X'_t - X_{current} \tag{2}$$

$$\Delta\theta_{t+1} = \frac{1}{k_p} e_t + \frac{1}{k_i} \int_0^t e_t + \frac{1}{k_d} \frac{\mathrm{d}\, e(t)}{\mathrm{d}\, t} \tag{3}$$

$$\approx \frac{1}{k_p} e_t + \frac{1}{k_i} \sum_j e_t + \frac{1}{k_d} (e_t - e_{t-1}) \tag{4}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_{t+1} \tag{5}$$

in which, e_t is the error between the target force X'_T and the current measured force $X_{current}$, and e_{t-1} is the error of the previous time step t. The PID controller calculates the motor position changes for each time step t. Its parameters are set as: $k_p = 50$, $k_i = 85$, $k_d = 16$. θ_{t+1} and θ_t represents the motor positions for the current and next low-level time step t, respectively.

B. Experiment procedure

The PID controller in the low level is firstly tuned to make sure that the cable tension can be adjusted to 300 within 80 low level time steps Δt . Then we start the actual experiment. The time step of the high level controller ΔT is set as 5, 10, 20, 30, 40, 50, 60, 70, and 80 times that of the low level time step Δt. For each of the high level time step ΔT settings, 13 different initial forces, from 60 to 300 with an interval of 20, are used as the first time step ΔT target forces F_T. Target

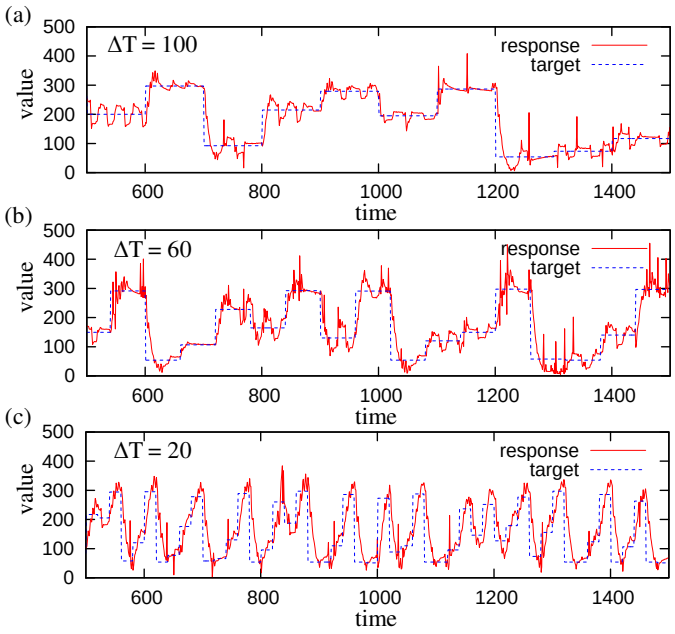

Fig. 3. Time series for different ΔT. (a) ΔT = 100, (b) ΔT = 60, and (c) ΔT = 20.

forces for the following time steps are generated by the logistic map (1) and has been elaborated in the previous subsection. We run 100 high level time steps ΔT for each initial force condition. The target forces and actual forces at each time step Δt are recorded and saved as text files for further analysis.

III. Result and Discussion

Fig. 2 (c) shows the averaged absolute error over 10 response curves under the same setting with the initial force 0 and target force 300. Its error bar shows the standard deviation. We can see that if the value of ΔT is smaller than 40, the actual force did not reach the target value. Also, around ΔT = 60, the overshoot can be observed, and by increasing ΔT, the error converges to 0. Fig. 3 shows the typical time series of the actual response and the target value when ΔT = 100, 60, and 20. When ΔT was small, we sometimes observed an oscillation-like trajectory. Fig. 4 (a) shows the return map structure constructed by the actual response. The modulation effects can be clearly observed, especially in the region that $50 < X'_t < 275$. Interestingly, we can also observe a splitting of the map in this region. This splitting means that although X'_t takes the same value, the resulting $X'_{t+\Delta T}$ could have different values. This would be caused by the context dependency of the underlying physical process, but should be clarified in future work. Fig. 4 (b) shows the averaged error when varying ΔT. The interesting point is that although we vary ΔT, the error tends to be constant. Since the response curve of the PID controller depends on both the current state and the target state, probability distribution of the state according to ΔT will help to clarify the mechanism in future work. It should be noted that the results presented here might contain noises besides the PID errors from the several physical factors in the experiment platform, for example, friction and inertia forces. These factors should also be clarified in future work.

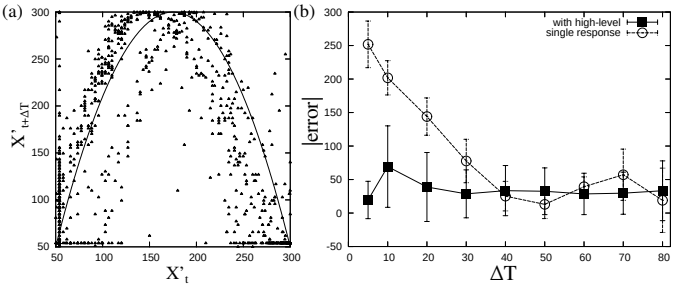

Fig. 4. (a) The return map plotted by using the actual responses when ΔT = 20. The line plot shows the scaled logistic map for the reference. (b) The averaged error according to ΔT. To calculate the averaged error, we run the system at each ΔT from different initial conditions (the first target forces). See experiment procedure for details

In this paper, we proposed a methodology to enhance the mutual coupling among control, body, and environment [1], demonstrated the concept by using a physical platform, and showed the preliminary results. Our focus was to release the deadtime and analyze the body dynamics underneath it, considering that the clock which guarantees the satisfying computational process is limiting the couplings. Then how do biological systems determine their clock? Recent findings suggest that high dimensional nonlinear dynamical systems can contain the ability to achieve the sense of time duration in the brain. Also, it is suggested that this is not realized as a static clock but rather a dynamic one which changes its property constantly according to external stimuli [4]. Since biological systems do not have an external tuner for their clocks but should adjust the clocks by themselves, if we consider how they self-organize and self-tune their clocks, we should take into account the interaction between the system and their environment through their bodies. This would be an interesting direction to be explored for autonomous control of robots [5].

Acknowledgment

This work was supported by the ICT-FET OCTOPUS Project (EU project FP7-231608).

References

[1] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *Science*, vol. 318, pp. 1088–1093, 2007.

[2] T. Li, K. Nakajima, M. J. Kuba, T. Gutnick, B. Hochner, and R. Pfeifer, "From the octopus to the octopus robot and back: a biologically inspired behavior control architecture for an octopus inspired soft robot," *Vie Milieu (unpublished)*, 2011.

[3] M. Cianchetti, A. Arienti, M. Follador, B. Mazzolai, P. Dario, and C. Laschi, "Design concept and validation of a robotic arm inspired by the octopus," *Mater. Sci. Eng. C*, vol. 31, pp. 1230–1239, 2011.

[4] U. R. Karmarkar and D. V. Buonomano, "Timing in the absence of clocks: Encoding time in neural network states," *Neuron*, vol. 53, pp. 427–438, 2007.

[5] K. Nakajima and T. Ikegami, "Dynamical systems interpretation of reversal of subjective temporal order due to arm crossing," *Adapt. Behav.*, vol. 18, pp. 189–210, 2010.