Robotics and Autonomous Systems 101 (2018) 20–33

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Decentralized control of rhythmic activities in fully-actuated/under-actuated robots

M. Yazdani, H. Salarieh *, M. Saadat Foumani

School of Mechanical Engineering, Sharif University of Technology, Tehran, Iran

HIGHLIGHTS

- A new supervisory hierarchical control framework of robot has been proposed.
- The method can be applied to fully-actuated and under-actuated robots to control rhythmic activities.
- The method consists of two independent layers: the High-Level, and a network of Low-Level controllers.
- The proposed framework replicates the functionality of the CPGs in the animals and its learning capability.
- The high-level controller acts as a supervisor which control the system when the low-level controllers are not trained enough or they are unable to control the system stably.
- The low-level controllers are simple and local, designed to replicate the functionality of the CPGs.

ARTICLE INFO

Article history:
Received 10 January 2017
Received in revised form 8 October 2017
Accepted 5 December 2017
Available online 23 December 2017

Keywords:
Rhythmic activity
Online learning
Distributed control
Supervisory control
Biped robot

ABSTRACT

Rhythmic activities such as swimming stroke in the human body are learnable through conscious trainings. Inspiringly, the main objective of this study is to develop a control framework to reproduce the described functionality in the imitating robots. To do so, a two layer supervisory controller is proposed. The high-level controller, which acts as the conscious controller during trainings, is a supervisory dynamic-based controller and uses all system sensory data to generate stable rhythmic movements. On the other hand, the low-level controller in this structure is a distributed trajectory-based controller network. Each node in this network is an oscillatory dynamical system which has the ability to learn and reproduce the desired trajectory. Also, each node has a critic agent which evaluates the control eligibility of the low-level controllers for controlling the system. Then, based on the evaluation, these agents decide to assign the control of the system to the high-level controller or the low-level controllers. By using this structure, the system controller will act as simple and computing efficient as trajectory-based controllers and will perform as stably and robustly as dynamic-based controllers. At last, the applicability of this framework is demonstrated on a fully actuated robot and on an under-actuated biped robot.

1. Introduction

Rhythmic activities are one of the most prevalent motions in many robotic systems and biological ones and range from manipulatory tasks to locomotion. As they are periodic, specific control methods have been developed for these tasks, e.g. see [1–4].

In this field, some researchers developed dynamic-based methods to control these movements, e.g., see [5–9]. The main foci of these strategies are on the dynamics and kinematics of the system to generate rhythmic motion. Although such an approach needs extensive knowledge of the mechanical structure and high computational power, they perform more robustly than trajectory-based controllers under moderate disturbances.

In contrast, in the last decade, inspired by nature, many researchers have developed *Central Pattern Generators* (a.k.a CPGs) to control such movements [10–14]. The term CPG describes neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs [15]. Furthermore, the CPGs can be influenced by higher centers in the brainstem or cortex [16]. Inspiringly, it is suggested a two-layer hierarchical controller architecture in which the low-level controller contains CPGs and the high-level controller envisioned as the brainstem and provide information to promote the modulation of the reference trajectories, e.g., see [17]. Although the high-level controller has been added in these architectures, the reference trajectories are always computed by the CPGs and then

* Corresponding author.
E-mail addresses: masoudyazdani@mech.sharif.ir (M. Yazdani), salarieh@sharif.ir (H. Salarieh), m_saadat@sharif.ir (M. Saadat Foumani).

https://doi.org/10.1016/j.robot.2017.12.003
0921-8890/

these trajectories will be pursued by means of feedback controllers. Therefore, the system cannot recover from the moderate to heavy disturbances.

In this article, a new two-layer hierarchical controller is proposed. In contrast to previous works, in this architecture, the high-level controller is considered as a dynamic-based controller and the low-level controller is a new CPG model which is developed to make it more simple and more general. In another word, the high-level controller is a general feedback loop which uses all available sensory data from the system and the environment and generates stable rhythmic movement. On the other hand, the low-level controller is a network of simple controllers in which each component is corresponding to a controllable degree of freedom (a.k.a. DoF) in the system and responsible to generate command for its actuator. In this structure, these controllers will be trained to stabilize the system solely by using local information (position and velocity of the corresponding joint) and by retrieving synchronization information from the neighbors.

In this framework, the high-level controller is envisioned as a supervisor which takes actions only when the simple controllers are unable to control the whole system. In this situation, the whole system will be controlled by the central controller and the simple controllers will automatically switch to learning mode, trying to generate the trajectories based on the output of the controlled system. They will be held in this state until their performance meets minimum required criteria (the generated trajectories follow the desired ones within a maximum allowable error). Then, the central controller will be turned off and the system will be controlled by the learned simple controllers.

By using the proposed architecture, the distributed nature of low-level controllers makes it more robust to communication and hardware failure. Besides, it reduces the production costs and increases the modularity in the robots designs in comparison with the previous works. Also, by using a sophisticated high-level controller as a supervisory controller in this framework, the system will take advantage of the robustness and the stability of the high-level controller as well as the simplicity and computation efficiency of low-level controllers.

To implement this framework, the requirement of the high-level controller is presented in Section 2 and the low-level controllers and their subsystems are presented in Section 3. After discussing the control framework and its components, this framework is implemented on an under-actuated biped robot in Section 4 and on a fully-actuated manipulator robotic arm in Section 5. Finally, the conclusion is drawn in Section 6.

2. High-level controller

The high level controller is an arbitrary controller which could be an internal controller or it could even be an external supervisory system attached to the system. Either way, the selection of high level controller should be based on the following criteria:

- The output of the closed-loop system with this controller should be periodic.
- The closed-loop trajectory should be stable.

To have a periodic stable trajectory, it is trivial that the selected controller should meet these criteria. However, for the under-actuated system, the following criterion should be met:

- The controller should not be time-dependent and should be strictly in the form of state feedback.

This condition ensures the stability of the system when switching occurs. Generally, when the system is controlled by the low-level controllers, the synchronization between the states of the system and the absolute time, which has been put in place by the high-level controller, gets disturbed. Therefore, if the high-level controller is time-dependent, when the system switches back to be controlled by the high-level controller, it may get destabilized or perform poorly.

3. Low-level controller

The low-level controllers are simple identical entities connected in a network and each one is corresponding to one of the controllable DoFs of the system. The structure of each node and its interaction with the high-level controller is depicted in Fig. 1. As it is shown, each node contains a learning/imitating agent which has two modes: (1) Learning Mode and (2) Imitating Mode. When the system is in the learning mode, it is controlled by the high-level controller and the mission of this agent is to learn to generate the trajectory of the attached DoF. On the other hand, in the imitating mode, it reproduces the desired trajectory synchronously with the desired trajectories of the other DoFs. In this mode, the control agent will try to eliminate the difference between the system output and the produced trajectory of the corresponding DoF. Furthermore, the critic agent will be in charge of switching between learning mode and imitating mode based on the difference between produced trajectories and the outcome of the system. It should be noted that when low-level controllers switch to imitating mode, the high-level controller is bypassed and the system is controlled by the network of the low-level controllers.

These agents are studied in the rest of this section. First, control agent and its properties will be brought up in Section 3.1. Then, learning/imitating agent and its structure will be discussed in Section 3.2. At last, the critic agent will be introduced and discussed in Section 3.3.

3.1. Control agent

The main focus of this framework is on robotic systems. One of the most straight-forward methods to derive the governing dynamics equation for these systems is Lagrange method. By applying this method, their models can be written generally in the form of

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u}. \tag{1}$$

The matrix $\mathbf{D}$ is the inertia matrix; $\mathbf{C}$ is the Coriolis matrix; $\mathbf{g}$ is the gravity vector; $\mathbf{q} = (\mathbf{q}_a, \mathbf{q}_u)$ is the generalized coordinate of the system where $\mathbf{q}_a$ and $\mathbf{q}_u$ are corresponding to the actuated DoFs and the unactuated DoFs respectively. Furthermore, $\mathbf{u}$ denotes the inputs of the system and the matrix $\mathbf{B}$ maps inputs of the system to the generalized forces. Without loss of generality, it can be supposed that

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_{M\times M} \\ \mathbf{0}_{(N-M)\times M} \end{bmatrix}, \tag{2}$$

where M is the number of actuated DoFs and N is the total number of DoFs.

Suppose that the evolution of the generated trajectory by the imitating/learning agent is expressed as $(\mathbf{q}^*, \dot{\mathbf{q}}^*)$ and evolution of the desired input signals are denoted by $\mathbf{u}^*$. The mission of the control agent is to eliminate the difference between the desired trajectories and the actual ones. To do so, the structure of the low-level control agent for the ith actuated DoF is considered as follows,

$$u_i = u_i^* + \delta u_i, \tag{3}$$

where δu_i should stabilize the perturbed system from the desired trajectory. To design the controller, let us consider the perturbation from the desired trajectory is denoted by $(\delta\mathbf{q}, \delta\dot{\mathbf{q}})$. Therefore, the