A Motion Learning Method using CPG/NP

Fumio Nagashima[1]

[1]Fujitsu Laboratories, 10-1, Morinosato-Wakamiya, Atsugi 243-0197, Japan shiro@stars.flab.fujitsu.co.jp

1. Introduction

I propose a motion learning method for a robot using a Central Pattern Generator with Numerical Perturbation(CPG/NP). The Central Pattern Generator(CPG) is modeled as a sub circuit of Recurrent Neural Network(RNN). Numerical Perturbation(NP) determines coefficients for each perturbed order RNN, step by step. System inputs are a motion outline, the direction of perturbation and some pieces of advice. The experiments using HOAP-1 indicate that this method can generate a variety of motions, however, the calculation time dramatically decreases.

2. CPG model

I use 4 assumptions to make a RNN model, “delay”, “multiplication by constant”, “summation” and “switch”. The RNN based on these assumptions can create a variety of equations for control. Some RNN sub circuits can generate a group of functions. I call these RNN sub circuits “CPG” and the group of functions “CPG function group”.

3. NP method

The perturbation method is well known method for getting approximate solutions of non-linear equations system. I use an idea of perturbation method for obtaining motion. I assume the motion is a solution of non-linear equation system. I assume the solution is expressed by

$$y = \delta_0 y_0 + \delta_1 y_1 + \delta_2 y_2 + \delta_3 y_3 + \cdots, \qquad (1)$$

where y is the output of RNN, y_i is a function generated by CPG, δ_i is a coefficient of order i.

The strategy is simple. I use a numerical sum of linear equations’ solutions generated by CPGs to solve the unknown non-linear motion pattern equation system.

4. Learning system outline

The CPG can generate several type of functions. Some functions are orthogonal to each other. Some functions are useful to generate motion patterns.

The NP method can narrow down the number of coefficients that must be determined simultaneously, making it easier to determine coefficients.

Fig.1 shows an abstract algorithm for learning. System input are some pieces of advice and initial motion, and output is RNN circuit for motion.

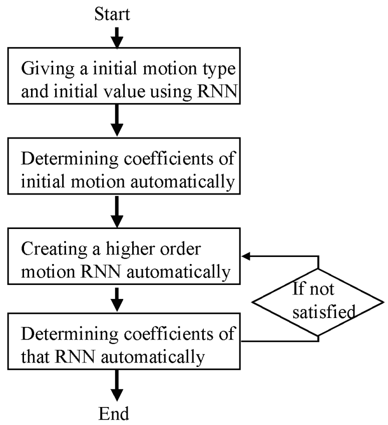

Figure 1: Outline Algorithm of the Proposed Method

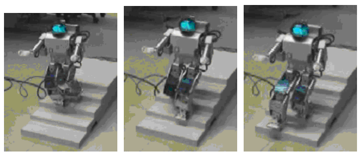

Figure 2: Experiment using HOAP-1

5. Experiments using HOAP-1

As a result, a 2nd order solution is sufficient for walking on a flat floor, though it is a little bit unstable. I call this motion “baby walking”. A 4th order solution is stable. It can almost do static walking. A 6th order solution can go up/down stairs. I tried testing using the lower order solution to go up/down stairs but it always fialed. Fig.2 shows the snapshot of experiment. The variation of motion increases as the NP order increases.