

# Clustered Regression Control of a Biped Robot Model

Olli Haavisto and Heikki Hyötyniemi

Helsinki University of Technology, Control Engineering Laboratory  
Finland

## 1. Introduction

Controlling of a biped walking mechanism is a very challenging multivariable problem, the system being highly nonlinear, high-dimensional, and inherently unstable. In almost any realistic case the exact dynamic equations of a walking robot are too complicated to be utilized in the control solutions, or even impossible to write in closed form.

Data-based modelling methods try to form a model of the system using only observation data collected from the system inputs and outputs. Traditionally, the data oriented methods are used to construct a *global* black-box model of the system explaining the whole sample data within one single function structure. Feedforward neural networks, as presented in (Haykin, 1999), for example, typically map the input to the output with very complicated and multilayered grid of neurons and the analysis of the whole net is hardly possible. Local learning methods (Atkeson *et al.*, 1997), on the other hand, offer a more structured approach to the problem. The overall mapping is formed using several *local* models, which have a simple internal structure but are alone valid only in small regions of the input-output space. Typically, the local models used are *linear*, which ensures the *scalability* of the model structure: Simple systems can be modelled, as well as more complex ones, using the same structure, only the number of the local models varies.

In robotics, local modelling has been used quite successfully to form inverse dynamics or kinematic mappings that have then been applied as a part of the actual controller (Vijayakumar *et al.*, 2002). However, when trying to cover the whole high-dimensional input-output space, the number of local models increases rapidly. Additionally, external reference signals are needed for the controller to get the system function as desired.

To evaluate the assumption of simple local models, a feedback structure based on linear local models, *clustered regression*, is used here to implement the gait of a biped walking robot model. The local models are based on *principal component analysis* (see Basilevsky, 1994) of the local data. Instead of mapping the complete inverse dynamics of the biped, only one gait trajectory is considered here. This means that the walking behaviour is stored in the model structure. Given the current state of the system, the model output estimate is directly used as the next control signal value and no additional control solutions or reference signals are needed. The walking cycle can become *automated*, so that no higher-level control is needed.

This text summarizes and extends the presentation in (Haavisto & Hyötyniemi, 2005).

## 2. Biped model

### 2.1 Structure of the mechanism

The biped model used in the simulations is a two-dimensional, five-link system which has a torso and two identical legs with knees. To ensure the possibility of mathematical simulation, the biped model was chosen to be quite simple. It can, however, describe the walking motion rather well and is therefore applied in slightly different forms by many researchers (see e.g. Chevallereau *et al.*, 2003; Hardt *et al.*, 1999; Juang, 2000).

Fig. 1 shows the biped and the coordinates used. Each link of the robot is assumed to be a rigid body with uniformly distributed mass so that the centre of mass is located in the middle of the link. The interaction with the walking surface is modelled by adding external forces  $F$  to the leg tips. As the leg touches the ground, the corresponding forces are switched on to support the leg. The actual control of the biped gait is carried out using the joint moments  $M$ , which are applied to both thighs and knees.

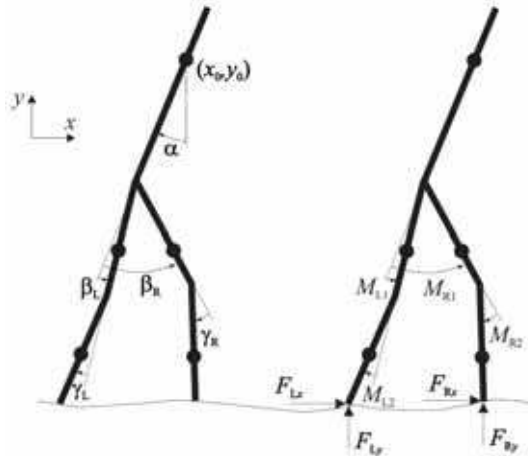


Fig. 1. The coordinates used (left) and the external forces (right).

As such, the system has seven degrees of freedom, and seven coordinates must be chosen to describe the configuration of the biped. The coordinate vector now is

$$q = (x_0 \ y_0 \ \alpha \ \beta_L \ \beta_R \ \gamma_L \ \gamma_R)^T. \quad (1)$$

Here the subindices L and R refer to the "left" and "right" leg of the biped, respectively. The dynamic equations of the system can be derived using *Lagrangian mechanics* (Wells, 1967) and the result has the following form:

$$A(q) \ddot{q} = b(q, \dot{q}, M, F), \quad (2)$$

where the "dotted variables" denote time derivatives of the coordinate vector. The exact formulas of the *inertia matrix*  $A(q)$  and the right hand side vector are quite complex and therefore omitted here (see Haavisto & Hyötyniemi, 2004). The minimum dimension  $n$  of the state vector representing the system state is 14 (the 7 coordinates and their time derivatives); however, because of the efficient compression carried out by the proposed data-based modelling technique, the dimension of the "extended state" could be higher, including measurements that are not necessarily of any use in modelling (redundancy among the measurements being utilized for elimination of noise).

## 2.2 Simulator

To simulate the biped walking on a surface, a Matlab/Simulink (MathWorks, 1994-2007) block was developed. It handles internally the calculation of the support forces  $F$  using separate PD controllers for each force component: The force opposing the movement is the stronger the farther the foot tip has gone beneath the ground surface (P term), and the faster the foot tip is moving downward (D term). By adjusting these controller parameters, different kinds of ground surface properties can be simulated (very hard surfaces, however, resulting in high force peaks, so that short simulation step sizes have to be applied). Also the knee angles are automatically limited to a range defined by the user. To obtain this, the simulation block adds an additional moment to the knee joints if the angle is exceeding or going under the given limits. This limitation can be used for example to prevent the knee turning to the "wrong" direction, that is, to keep the angles  $\gamma_L$  and  $\gamma_R$  positive.

The input of the simulation block is the four dimensional moment vector

$$M = (M_{L1} \quad M_{R1} \quad M_{L2} \quad M_{R2})^T, \quad (3)$$

and the output the 14-dimensional state of the system augmented with leg tip touch sensor values  $s_L$  and  $s_R$  is

$$\left( q^T \quad \dot{q}^T \quad s_L \quad s_R \right)^T. \quad (4)$$

If the leg is touching the ground, the sensor value equals to 1, otherwise 0.

The biped dynamics, ground support forces and knee angle limiters are simulated in continuous time, but the input and output of the block are discretized using zero order hold. This allows the use of discrete-time control methods. A more detailed description of the used biped model and simulation tool is presented in the documentation (Haavisto & Hyötyniemi, 2004).

## 2.3 Data collection

In general, the gait movement of a biped walker can be divided into two phases according to the number of legs touching the ground. In the *double support phase* (DSP) both legs are in contact with the ground and the weight is moving from the rear leg to the front leg. In the *single support phase* (SSP) only the *stance leg* is touching the ground, while the *swing leg* is swinging forward. Because the biped is symmetrical regarding the left and right leg, the whole gait can be described with one DSP and one SSP. In every other step the stance and swing leg signals are switched to model the left and right leg swing in turns. This means that it is necessary to model only one DSP and one SSP with, say, left leg acting as the rearmost or swing leg and right leg as the foremost or stance leg. Accordingly, the data collected during this work were transformed to cover only these DSP and SSP cases.

In order to collect sample data from the gait of the walking biped, a rough PD control scheme was developed. It was based on predetermined reference signals and four separate and discrete PD controllers, which produced the moment inputs for the biped. Fig. 2 shows the closed loop simulation model of the controlled biped. The *PD controller* block includes the separate PD controllers, whereas the *Biped model* block simulates the actual walking system and the surface interaction.

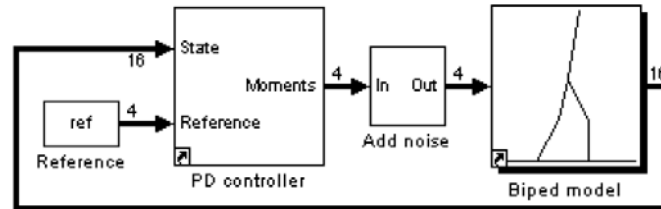


Fig. 2. The sample data were produced by simulating the PD controlled biped

The PD controller was able to control the biped so that a cyclic gait movement was attained. To reach more varying data, independent white noise was added to each moment signal component. The input and output data of the biped system were collected of a 40 seconds walk, during which the biped took over 40 steps. The sampling interval was equal to the discretization period  $h = 0.005$  sec.

The data were mean-zeroed and scaled to unit variance in order to make the different components equally significant. Also the  $x_0$  coordinate of the state vector was omitted because it is monotonically increasing during the walk instead of being cyclic. As a result, the modelling data had 8000 samples of the 15 dimensional (reduced) state vector and the corresponding four dimensional moment vectors. The points were located around a nominal trajectory in the state and moment space.

In the following, the *state vector* at time  $kh$  is denoted by  $u(k)$  and the *moment vector* by  $y(k)$  since these are the *input* and the *output* of the *controller*. Additionally, the state vector without the touch sensor values is denoted by  $\tilde{u}(k)$ .

### 3. Clustered Regression

The data-based model structure, *clustered regression*, that is applied here is formed purely by the statistical properties of the sample data. The main idea is to divide the data into *clusters* and set an *operating point* in the centre of each cluster. Every operating point, additionally, has its own local linear regression model which determines the local input-output mapping in the cluster region. A scheme of the model structure is presented in Fig. 3.

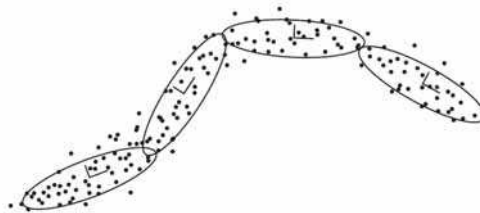


Fig. 3. The local models cover the whole data along the trajectory

#### 3.1 Local model structure

Each local model calculates a linear *principal component regression* (PCR) estimate of the given regression input. Principal components show the orthogonal directions of the highest variances in the input data. It is assumed here that variance is carrying information and the most important principal components are therefore relevant in the data, whereas the smaller

ones can be omitted as noise. The idea of the PCR is first to map the input data to the lower dimensional principal component subspace and then use multivariable linear regression to get the output estimate.

Let us denote the input signal for the local model  $p$  at the time instant  $kh$  as  $u^p(k)$ , and the corresponding output as  $y^p(k)$ . When the input data are mapped onto the principal components, one gets the latent variable signal  $x^p(k)$ . To simplify the regression structure the latent variable data are locally scaled to unit variance before the regression mapping.

Now the whole regression structure of the local model  $p$  can be stored in the following statistics, which are calculated using the data in the corresponding cluster:

- $C_u^p$  Expectation value of the input vector  $u^p$ .
- $C_y^p$  Expectation value of the output vector  $y^p$ .
- $P_{xx}^p$  Inverse of the latent variable data  $x^p$  covariance matrix.
- $R_{xu}^p$  Cross-covariance of the latent and reduced input data.
- $R_{yx}^p$  Cross-covariance of the output and latent variable data.
- $R_{uu}^p$  Covariance of the input data.

Note that in the following the statistics are calculated using the collected data, which gives merely the *estimates* of the real quantities. It is assumed, however, that the estimates are accurate enough and no difference is made between them and the real values.

The regression calculation itself is based on a Hebbian and anti-Hebbian learning structure (see 6.1). Assuming that the statistics correspond to the real properties of the data, the output estimate of the model  $p$  given an arbitrary input vector can be expressed in mathematical terms as

$$\hat{y}^p(k) = R_{yx}^p \left( P_{xx}^p \right)^2 R_{xu}^p \left( \tilde{u}(k) - C_u^p \right) + C_y^p. \quad (5)$$

Here  $C_u^p$  is the expectation value of the reduced state vector  $\tilde{u}^p$ . In (5) the input is first transformed to the operating point  $p$  centred coordinates by removing the input mean value; then the estimate is calculated and the result is shifted back to the original coordinates in the output space by adding the output mean vector.

A cost value for the estimate made in the unit  $p$  should be evaluated to measure the error of the result. The cost can depend for example on the distance between the cluster centre and the estimation point:

$$J^p(u(k)) = \frac{1}{2} \left( u(k) - C_u^p \right)^T H^p \left( u(k) - C_u^p \right), \quad (6)$$

where  $H^p$  is a constant weighting matrix.

### 3.2 Combination of the local models

The overall estimate of the clustered regression structure is calculated as a weighted average of all the local models. Assuming that the number of the operating points is  $N$ , one has

$$\hat{y}(k) = \frac{\sum_{p=1}^N K^p(k) \hat{y}^p(k)}{\sum_{p=1}^N K^p(k)}. \quad (7)$$

The weights should naturally be chosen so that the currently best matching local models affect the final results the most, whereas the further ones are practically neglected. Let us choose

$$H^p = \frac{1}{\sigma} (R_{uu}^p)^{-1}, \quad (8)$$

that is, the weighting matrix for each local model cost calculation equals the scaled inverse of the input data covariance matrix,  $\sigma$  being the scaling factor; assuming that the data in each local model are normally distributed, the *maximum likelihood* estimate for the combined output value is obtained when the weights in (7) are chosen as

$$K^p(k) = \frac{1}{\sqrt{(2\pi)^n |(H^p)^{-1}|}} \exp(-J^p(u(k))), \quad (9)$$

and  $\sigma = 1$ . The simulations, however, showed that a more robust walking behaviour is reached with larger scaling parameter values. This increases the effect of averaging in the combination of the local model estimates.

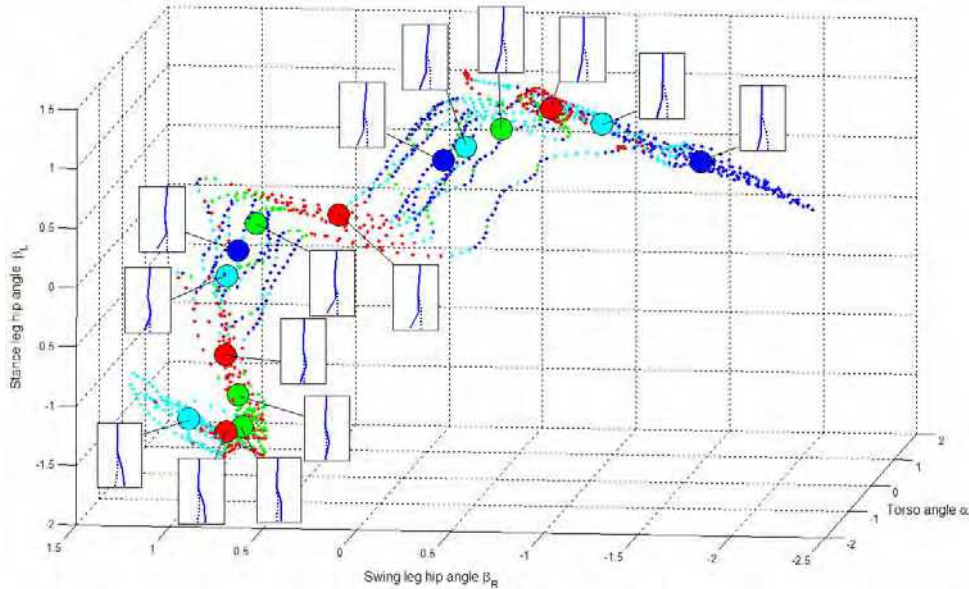


Fig. 4. Clustered teaching data and the operating points

## 4. Clustered Regression Control

### 4.1 Teaching the model

The clustered regression model was formed using the sample data collected from the PD controlled gait. The model input was the state vector  $u(k)$  and output the corresponding control signal  $y(k)$ . The data were first divided into  $N = 15$  clusters located along the nominal trajectory in the state and output space. Based on the data belonging to each cluster the estimates of the statistics listed in 3.1 were calculated with eight principal component directions in each model.

Fig. 4 shows part of the clustered teaching data projected to three input variables. The operating point centres are shown as large circles with the same shade as the data belonging to the corresponding clusters. Also the state of the system in the operating point centres is drawn. Clearly the start of a new DSP is located in the right part of the figure. Following the nominal trajectory the state of the biped then changes to a SSP after some four operating points. Finally, the step ends just before the beginning of a new DSP in the lower left corner of the figure, where the state of the system is almost identical with the original state, only the legs are transposed.

**4.2 Biped behavior**

To test the attained model, a Simulink block was developed to realize the estimation of the control related to the measured current state of the system. The block could now be used to control the biped instead of the PD controller module applied in the data collection. It appeared that the taught clustered regression model was able to keep the biped walking and produced a very similar gait as the original PD controller. Fig. 5 shows two steps of the both gaits.

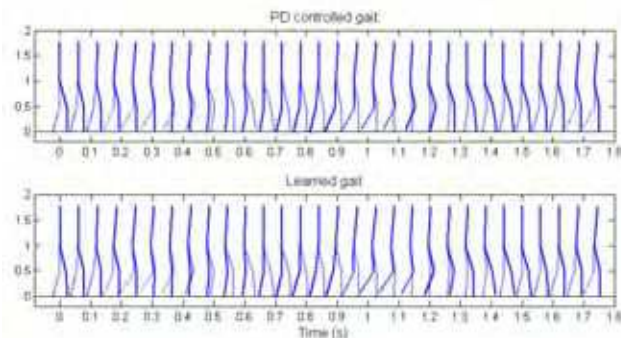


Fig. 5. The learned gait was qualitatively quite similar to the original one

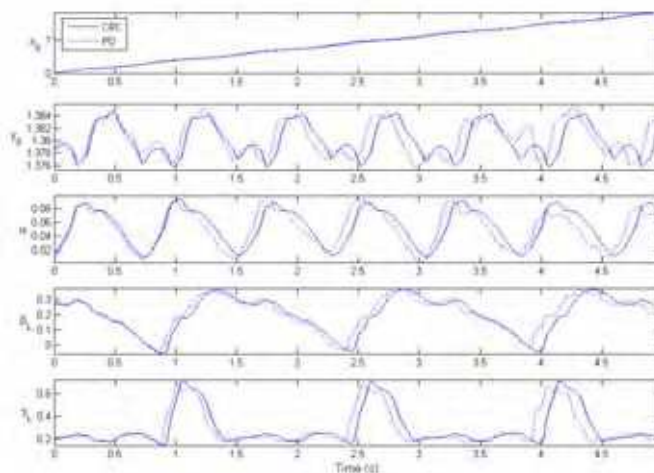


Fig. 6. The clustered regression controlled system (CRC) is functioning a bit slower than the PD controlled one.

The biggest difference between the two gaits is that the clustered regression controlled one is proceeding a little slower. When comparing the coordinate value changes, the lag of the clustered regression controlled system is clearly seen (Fig. 6). The variations in the PD controlled signals are due to the added noise in the teaching data generation. However, the clustered regression controller can reproduce the original behaviour very accurately.

## 5. Optimization

### 5.1 Dynamic programming

It was shown that the clustered regression controller is able to repeat the unoptimized behaviour used in the teaching. It would be very beneficial, however, to reach a more optimal control scheme so that the biped could, e.g., walk faster with smaller energy consumption; this would also be a strong evidence of the validity of the selected model structure when explaining biological systems. It would be interesting to see if the biped would learn new *modes* of moving: For example, if the speed were emphasized, would it finally learn to run?

The optimization procedure of the clustered regression structure could be compared with *dynamic programming* (Bellman, 1957) that is based on the *principle of optimality*. In this case this idea can be formulated as follows:

*An optimal control sequence has the property that whatever the initial state and initial control are, the remaining controls must constitute an optimal control sequence with regard to the state resulting from the first control.*

In general, a dynamic optimization problem can be solved calculating the control from the end to the beginning. Starting from the final state, the optimal control leading to that state is determined. Then the same procedure is performed in the *previous* state and so on. Finally, when the initial state is reached, the whole optimal control sequence is attained.

This means that one can form the *global* optimized behaviour using *local* optimization. When compared with the clustered regression structure, the local optimization should now be done inside every cluster or operating point. When the system enters the operating point region, the controller is assumedly able to guide it optimally to the next region.

Another fact from the theory of optimal control states that for a quadratic (infinite-time) optimization problem for a linear (affine) system, the optimal control law is also a linear (affine) function of the state. This all means that, assuming that the model is locally linearizable, a globally (sub)optimal control can be implemented in the proposed clustered regression framework.

### 5.2 Optimization principle

As the walking motion is cyclic, one cannot choose the "first" or "last" operating point. Instead, the optimization can be carried out gradually in each cluster. A trial-and-error based optimization scheme which was used successfully in the previous work (Hyötyniemi, 2002) is presented in the following.

As a starting point for the optimization an unoptimized but easily attained behaviour is taught to the clustered regression controller (the original PD control). Then the current cost of the control for one cycle is calculated. The cost criterion can be chosen rather freely so that in the minimum of the criterion the behaviour reaches the desired optimum – for example, low energy consumption and high walking speed.



When a stable and cyclic motion is obtained the control is slightly varied at one time instant, and the new data point is adapted to the clustered regression model. If the new model gives a better control, that is, the cost is now smaller, it is accepted. On the other hand, if the cost is higher, the new model is rejected. After that, a new change can be made and similarly evaluated. Repeating these steps the controller can hopefully learn a more optimal behaviour, although the process may be quite slow and nothing prevents it ending to a local minimum of the cost function. This random search is an example of *reinforcement learning*; now, because of the mathematically simple underlying model structure, the adaptation of the model can still be relatively efficient.

### 5.3 Adaptive learning

In order to optimize the control, it should be possible to update the clustered regression structure using new data. This adaptation, however, turned out to be quite hard to realize because of the properties of the data clusters.

It was detected during the PD controlled gait reproduction that at least eight principal component directions need to be considered in each local PCR model to reach an accurate enough control estimate. The relative importance of each principal component can be described by the variance of the corresponding latent variable component. Fig. 7 shows the averages of the variances of the local models.

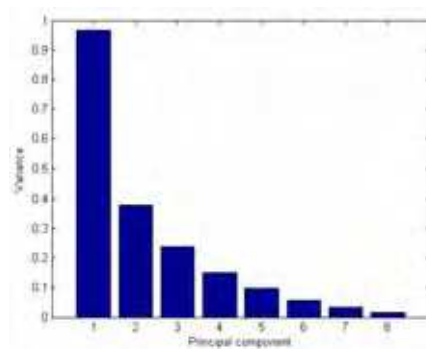


Fig. 7. Averages of the principal component variances in the local models

Clearly, the ratio of the first and last principal component variances is huge, which means that it is very hard to iteratively determine the last principal component directions. In practice, this means that when the training is repeated, always using the earlier model for construction of data for the next model, the model sooner or later degenerates. The tiny traces of relevant control information are outweighed by measurement noise.

## 6. Discussion

The presented scheme was just a mathematical model applying engineering intuition for reaching good control, and the main goal when constructing the model was simplicity. However, it seems that there is a connection to real neuronal systems.

It has been observed (Haykin, 1999; Hyötyniemi, 2004) that simple Hebbian/anti-Hebbian learning in neuronal grids results in a *principal subspace model* of the input data. This Hebbian learning is the principle that is implemented by the simple neurons (Hebb, 1949).

The principal subspace captures the principal components, and it is a mathematically valid basis for implementations of principal component regressions.

In (Hyötyniemi, 2006) the Hebbian learning principle is extended by applying *feedback through environment*. It turns out that when the nonideality is taken into account – exploiting signals also exhausts them – there is a negative feedback in the system; one can reach stabilization of the Hebbian adaptation without additional nonlinearities, and emergence of the principal subspace without complicated hierarchies among neurons. There can exist *self-regulation* and *self-organization* in the neuronal system, meaning that the adaptation of the global model can be based solely on the local interactions between individual neurons.

But as the biped structure is highly nonlinear, one needs to extend the linear model; here this was accomplished by introducing the clustered structure with submodels. How could such extension be motivated in a neuronal system, as it now seems that some kind of central coordination is necessary to select among the submodels and to master their adaptation? Again, as studied in (Hyötyniemi, 2006), there emerges *sparse coding* in a system with the Hebbian feedback learning. It can be claimed that in sparse coding the basis vectors are rotated to better match the physically relevant features in data – such behaviour has been detected, for example, in visual cortex (Földiák, 2002). In Hebbian feedback learning the correlating structures become separated, and they compete for activation; without any centralized control structures, the signals become distributed among the best matching substructures. As seen from outside, the net effect is to have “virtual” clusters, with smooth interpolation between them.

## 7. Conclusions

In this work, a clustered regression structure was used to model and control a walking biped robot model. It was shown that the purely data-based model is accurate enough to control the biped. The control structures can also be motivated from the physiological point of view.

The main problem is that to successfully reproduce the walking gait, the clustered regression controller should learn to keep the system well near the nominal trajectory. If the state of the system drifts too far from the learned behaviour, the validity of the local models strongly weakens and the system collapses. As a result, the robustness of the controller is dependent on the amount of source data variation obtained by additional noise. However, the clustered regression structure was unable to control the biped with the required noise level present in the PD controlled simulations. This complicates the iterative optimization process.

It was also shown that the management of no less than eight principal components is necessary; the “visibility ratio” between these principal components, or the ratio between variances, is over three decades. This also dictates the necessary signal-to-noise ratio. It seems that such accuracy cannot be achieved in biological systems; to construct a biologically plausible control scheme, the model structure has to be modified.

There are various directions to go. Principal components are always oriented towards the input data only, neglecting the output, or the outcome of the actions. This problem should somehow be attacked. One possibility would be to explicitly control the adaptation based on the observed outputs, so that rather than doing standard principal component analysis, some kind of *partial least squares* (PLS) (Hyötyniemi, 2001) approach would be implemented. Also, the data could be rescaled to emphasize the less visible PC directions.

It is also important to point out that the physical model of the biped robot is only a model, and some phenomena present in the data may be caused by the simulator. Especially the leg contacts with the ground may introduce some oscillations which probably would not appear in data collected from a real robot. This means that some of the less important principal component directions may as well describe these irrelevant effects thus making the modelling problem harder. In the future work, it would be interesting to analyze the principal component directions in each local model one by one and try to find out which of them are really connected to the actual walking motion.

## 8. References

- Atkeson, C. G.; Moore, A. W. & Schaal, S. (1997). Locally Weighted Learning. *Artificial Intelligence Review*, vol. 11, pp. 11-73.
- Basilevsky, A. (1994). *Statistical Factor Analysis and Related Methods*. John Wiley & Sons, New York.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.
- Chevallereau, C. & et al. (2003). RABBIT: A Testbed for Advanced Control Theory. *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57-79.
- Földiák, P. (2002). Sparse coding in the primate cortex. In Arbib M.A. (ed.). *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, Massachusetts (second edition).
- Haavisto, O. & Hyötyniemi, H. (2004). *Simulation Tool of a Biped Walking Robot Model*. Technical Report 138, Helsinki University of Technology, Control Engineering Laboratory.
- Haavisto, O. & Hyötyniemi, H. (2005). Data-based modeling and control of a biped robot. *Proceedings of The 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation - CIRA2005*, Espoo, Finland, June 27-30, pp. 427-432.
- Hardt, M.; Kreutz-Delgado, K. & Helton, J. W. (1999). Optimal Biped Walking with a Complete Dynamic Model. *Proceedings of The 38th Conference on Decision and Control*, Phoenix, Arizona, pp. 2999-3004.
- Haykin, S. (1999). *Neural Networks, a Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey (second edition).
- Hebb, D.O. (1949): *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, New York.
- Hyötyniemi, H. (2001). *Multivariate Regression: Techniques and Tools*. Technical Report 125, Helsinki University of Technology, Control Engineering Laboratory.
- Hyötyniemi, H. (2002). Life-like Control. *Proceedings of STeP 2002 - Intelligence, the Art of Natural and Artificial*, Oulu, Finland, December 15-17, pp. 124-139.
- Hyötyniemi, H. (2004). *Hebbian Neuron Grids: System Theoretic Approach*. Technical Report 144, Helsinki University of Technology, Control Engineering Laboratory.
- Hyötyniemi, H. (2006). *Neocybernetics in biological systems*. Technical Report 151, Helsinki University of Technology, Control Engineering Laboratory.
- Juang, J. (2000). Fuzzy Neural Network Approaches for Robotic Gait Synthesis. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 30, no. 4, pp. 594-601.

Wells, D. A. (1967). *Theory and Problems of Lagrangian Dynamics with a Treatment of Euler's Equations of Motion, Hamilton's Equations and Hamilton's Principle*. Schaum publishing, New York.

The MathWorks, Inc., *Matlab - The Language of Technical Computing*. See the WWW pages at [www.mathworks.com](http://www.mathworks.com).

Vijayakumar, S.; Shibata, T.; Conrardt, J. & Schaal, S. (2002). Statistical Learning for Humanoid Robots. *Autonomous Robots*, vol. 12, no. 1, pp. 55-69.



## **Humanoid Robots: New Developments**

Edited by Armando Carlos de Pina Filho

ISBN 978-3-902613-00-4

Hard cover, 582 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, June, 2007

**Published in print edition** June, 2007

For many years, the human being has been trying, in all ways, to recreate the complex mechanisms that form the human body. Such task is extremely complicated and the results are not totally satisfactory. However, with increasing technological advances based on theoretical and experimental researches, man gets, in a way, to copy or to imitate some systems of the human body. These researches not only intended to create humanoid robots, great part of them constituting autonomous systems, but also, in some way, to offer a higher knowledge of the systems that form the human body, objectifying possible applications in the technology of rehabilitation of human beings, gathering in a whole studies related not only to Robotics, but also to Biomechanics, Biomimetics, Cybernetics, among other areas. This book presents a series of researches inspired by this ideal, carried through by various researchers worldwide, looking for to analyze and to discuss diverse subjects related to humanoid robots. The presented contributions explore aspects about robotic hands, learning, language, vision and locomotion.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Olli Haavisto and Heikki Hyotyniemi (2007). Clustered Regression Control of a Biped Robot Model, Humanoid Robots: New Developments, Armando Carlos de Pina Filho (Ed.), ISBN: 978-3-902613-00-4, InTech, Available from:

[http://www.intechopen.com/books/humanoid\\_robots\\_new\\_developments/clustered\\_regression\\_control\\_of\\_a\\_biped\\_robot\\_model](http://www.intechopen.com/books/humanoid_robots_new_developments/clustered_regression_control_of_a_biped_robot_model)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.