LUND UNIVERSITY

A Neuron-Based Pulse Servo for Motion Control

DeWeerth, Steve; Nielsen, Lars; Mead, Carver; Åström, Karl Johan

1990

[Link to publication](#)

Total number of authors:
4

A Neuron-based Pulse Servo
for Motion Control

Steve DeWeerth
Lars Nielsen
Carver Mead
Karl Åström

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden	Document name Report
	Date of issue March 1990
	Document Number CODEN: LUTFD2/(TFRT–7446)/1-7/(1990)

Author(s) Steve DeWeerth, Lars Nielsen, Carver Mead, Karl Åström	Supervisor
	Sponsoring organisation

Title and subtitle
A Neuron-based Pulse Servo for Motion Control

Abstract

Paper at the 1990 IEEE International Conference on Robotics and Automation, May 13-18, Cincinnati, Ohio.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title		ISBN

Language English	Number of pages 7	Recipient's notes
Security classification		

A Neuron-based Pulse Servo
for Motion Control

Steve DeWeerth*, Lars Nielsen**, Carver Mead*, Karl Åström **

*Department of Computer Science, California Institute of Technology, Pasadena, Ca 91125, USA
**Department of Automatic Control, Lund Institute of Technology, Box 118, 221 00 Lund, Sweden

Abstract Sensory control based on biological computational paradigms can be implemented using low-power analog very large-scale integrated circuitry. We describe a design frame and a set of circuit elements with which generic motor controllers can be implemented. We then embed a simple proportional–derivative motor controller in the design frame, and describe its performance advantages over a more traditional controller. We also discuss the merits of biologically inspired systems used in applications related to robotics.

1.   Introduction

In recent years, the engineering community has demonstrated a great deal of interest in biological systems. One aim is complete artificial systems whose organizing principles are inspired by those found in biology. Such a system must be implemented in a technology that can incorporate neural-like processing. Current modeling of neural systems using analog VLSI (Mead, 1989) offers practical solutions to engineering problems with complex functionalities. Both systems (biological and artificial) use the properties of physical devices as computational primitives, and both technologies are limited by the interconnections rather than by the computations themselves. The result can be a vast amount of computation in a few square millimeters, and a few milliwatts.

Robotics is one field in which this biological inspiration could obviously be exploited. Biological motor systems can perform very gentle and precise motions even though the individual components, neurons and muscles, are imprecise. This is accomplished by using highly redundant systems that greatly improve accuracy and fault tolerance. Available robotic manipulators, on the other hand, are usually built from high precision components, but can only mimic the dexterity of a human hand in restricted cases.

In this paper, we describe a first achievement in this direction in which a new design methodology is used for the control of a friction limited servo. The results suggest that the size and reliability of the motion controller should be applicable to tasks such as the control of a robot hand built from low precision components like actuators designed with high friction lead screws.

2.   The Control Problem

Human motion control is a computational process, and so is conventional motion control in robotics. We will now discuss some aspects of these computational metaphors that have inspired the current effort.

Computation and control

Traditionally, technology has strongly influenced how engineers think about control systems. Implementations using relays, operational amplifiers, and microprocessors all have individual requirements that affect their uses. The microcomputer revolution, for example, has had a major impact on control system implementations. Because of shrinking cost and size combined with increases in performance, even single loop controllers are now implemented with microprocessors. As a consequence, control theory has developed in new directions, and perhaps more important, more advanced concepts like adaptive control have become popular (Åström and Wittenmark, 1984, 1989).

Computational metaphors may thus be suited or tailored for different types of control. When considering the general control problem, it is clear that the control research community spends more time on formulation than on solution. More general problems, like control principles, are seldom discussed as a major topic (Åström and Wittenmark, 1984, Chapter 7). Our attempt to investigate new metaphors for control computations is general in scope and could be applied to different control objects. However, the efforts were originally inspired by a robotics application, and from now on in the paper it will be assumed that the system to be controlled to is a robot.

Biological systems

As with other implementation media, neurobiology influences the computation used to solve a given control problem. The inherent imprecision in neurons and neural communication pathways requires system-level organization

that is quite different than traditional engineering implementations. Vertebrate nervous systems take advantage of parallelism, aggregation, and redundancy to make up for low-level imprecision. By using this type of high-level organization neural systems attain excellent system precision.

Biological organization characteristics put requirements on any physical medium capable of neural-like computations. First, the physical size of individual computational devices must be small to facilitate highly parallel systems. Also, the power generated by each device must be sufficiently small, allowing the system to dissipate the power generated by densely packed devices. Finally, since neurons aggregate large numbers of signals, there must be some mechanism to keep the resulting signals within operational range. Another important aspect of biological systems is the fact that information is transferred via nerve pulses. It is generally accepted that the pulse representation of the information improves reliability in the transfer of information. However, nature also makes a virtue out of using pulses when controlling muscles. Muscles are damped systems with a fair amount of internal friction (Månsson, 1989). We will later point out the advantages of pulses when driving friction limited systems.

A novel approach

If we are to use neural computational paradigms to implement control systems, we must first have a suitable technology in which to implement these systems. Mead (1989) has introduced a new highly-parallel low-power analog computing paradigm. We will in the following see that, by using this paradigm, it is possible to make regulators with a small number of circuit elements. These systems require a silicon surface which is many orders of magnitude smaller than a conventional system with A/D, CPU, Memory, and D/A. The system can also be made very reliable e.g. by duplicating the circuit elements, so that a component failure will only lead to a slight degradation in the system performance. Beside these advantages related to cost and function, we believe that the view on computational metaphors for control will be influenced by the possibilities inherent in the technology.

3. System Implementation

In the previous section, we described three criteria necessary for any technology capable of implementing highly redundant systems: small physical size, low power, and large signal range (highly compressive functionality). Mead(1989) has presented a design paradigm using analog integrated circuits which operate primarily in the subthreshold (weak inversion) region of transistor operation. This paradigm meets our three criteria quite well: First, systems can be designed with subthreshold MOS devices which use orders of magnitude less power than there counterparts using either standard (superthreshold) analog or digital design techniques. Also, these systems can be designed using a very small amount of silicon area with large,

highly redundant systems fitting on a fraction of a standard CMOS die. Finally, due to their exponential nature, subthreshold MOS transistors operate with signals over many orders of magnitude, allowing for the result of the aggregation of large numbers of signals to remain in the operating range of the devices.

Another advantage of this design methodology is the fact that both currents and voltages are used as signal types, giving more varied computational metaphors than standard analog systems which primarily use voltages. These facts make it especially easy to aggregate signals through current summing. In fact our approach is inspired by biological neurons, in which there is significant analog processing and aggregation leading to the possible pulse firing from the cell body. Using neurally inspired ideas and a small number of basic circuit elements, we have developed a design methodology for the design of fault-tolerant analog control systems. This section presents this "design frame" along with the necessary circuits for its implementation.

The Design Frame

There are two restrictions on the design frame into which we embed our systems. First, it must be flexible enough to allow us to implement a number of different types of regulators (i.e., multiple input, cascaded, etc.). It must also allow for the implementation of large-scale redundancy. These characteristics can be accomplished using a model for aggregation which is implemented as a weighted linear combination of differential inputs. This aggregation can be described by

$$y = \sum_n k_n (r_n - x_n)$$

where k_n is the proportionality constant and $r_n - x_n$ is the differential input for the nth stage. Since the sections of a standard regulator usually consist of a functional element multiplied by a constant, this system allows implementation of any elementary control element as long as its signal can be turned into a differential input. (In the next section, we will demonstrate how this can be accomplished for both the trivial case of a proportional element and the slightly more complex case of a derivative element.) Our system also allows for simple replication of the individual functional elements.

For the implementation described in this paper, the input signals are differential voltages. Also, for the purpose of aggregation, an intermediate current representation is used. Finally, as described earlier the output is encoded as pulses to be used to drive the actuators. The following subsections will describe the circuit elements necessary to implement this system, and then describe the actual circuit implementation of the design frame.
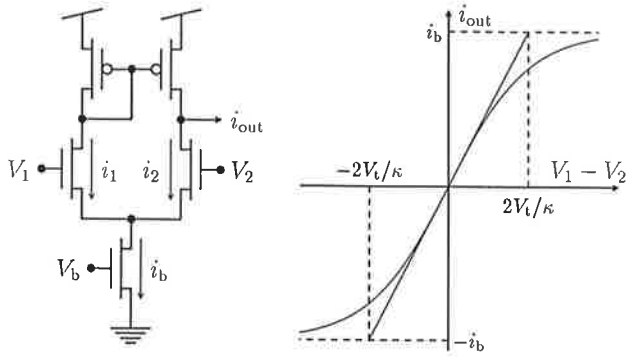
Figure 1. Transconductance amplifier

Transconductance amplifiers

The transconductance amplifier (Vittoz, 1985), shown in Figure 1, is a differential stage that produces an output current proportional to its differential input voltage $V_1 - V_2$ multiplied by its bias current i_b. This circuit consists of a bias transistor that sets the maximum output current of the amplifier, a differential pair of transistors, and a current mirror. The voltage V_b set the bias current i_b. This current is divided into i_1 and i_2 by the differential pair transistors according to input voltages V_1 and V_2. The current i_2 is directly subtracted from the output node. The current mirror replicates i_1 and adds it to the output node. Thus, the output current is the difference between the currents in the two branches of the differential pair: $i_{out} = i_1 - i_2$. The transconductance amplifier is generally used with very small bias currents, and operates well with all transistors in the sub-threshold regime. The output current for this circuit (shown in Figure 1) when the bias current is sub-threshold is

$$i_{out} = i_b \tanh\left(\frac{V_1 - V_2}{2}\right)$$

with voltages given in units of V_t/κ where V_t is the thermal voltage and κ is the gate voltage efficiency. Saturation behavior is clearly exhibited for large differential input voltages ($|V_1 - V_2| > 2V_t/\kappa$), at which point nearly all of the bias current is flowing through one of the transistors in the differential pair. For small differential voltages, the transfer equation can be approximated as

$$i_{out} \approx g(V_1 - V_2)$$

where $g = i_b \frac{\kappa}{2V_t}$ is the amplifier transconductance. We consider the linear approximation valid when $|V_1 - V_2| < 2kT/q\kappa \approx 100\text{mV}$.

The functionality of the transconductance amplifier can be extended by mirroring both differential pair currents (i_1 and i_2) to form a wide-range transconductance amplifier. The traditional reason for this extension is the isolation the output node from both differential pair transistors allowing a wider range of output voltages. However, this extension also facilitates multiple output currents. A wide-range amplifier with two outputs is shown in Figure 2. The output currents are complements of each other
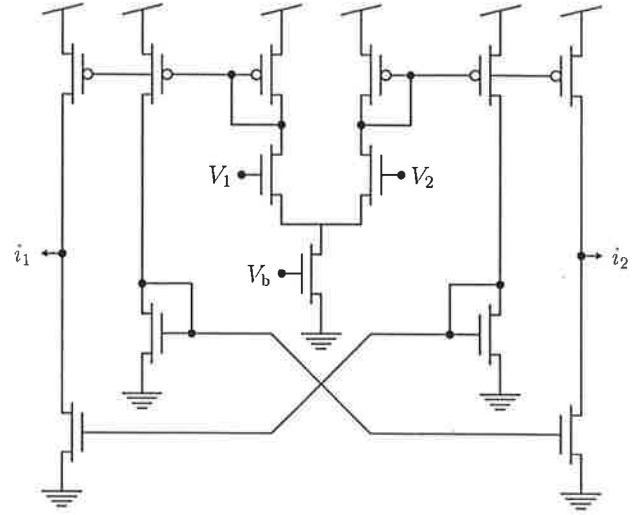

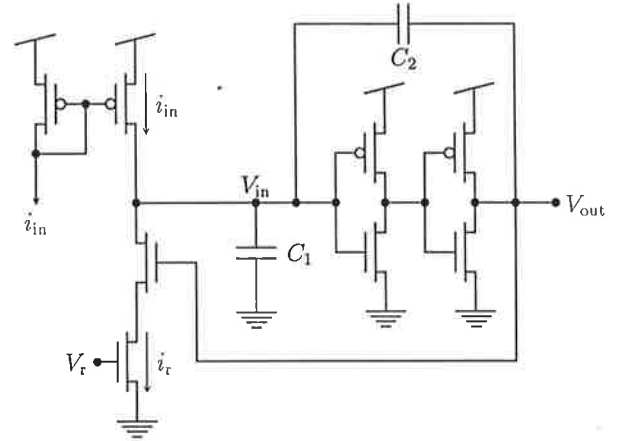**Figure 2.** Wide-range amplifier with complementary outputs

Figure 3. Neuron circuit

with $i_1 \approx g(V_1 - V_2)$ and $i_2 \approx g(V_2 - V_1)$. This variation on the transconductance amplifier is used in the implementation of the design frame.

The neuron circuit

Figure 3 shows a circuit that converts an input current into a series of pulses whose frequency is a monotonic function of the input current (Mead 1989). This circuit is named the neuron after the cells that generate nerve pulses in biological systems. The circuit consists of an input current mirror, a non-inverting high gain stage (two standard CMOS inverters), a pair of reset transistors, and two capacitors.

Figure 4 shows the response of the input and output voltages during pulse generation. The current mirror sources the current i_{in} onto the input node. When the input capacitance integrates enough charge to reach V_{inv} (the switching threshold of the input inverter), the inverters switch and the output goes high. This transient is fed back to the input and capacitively divided be-
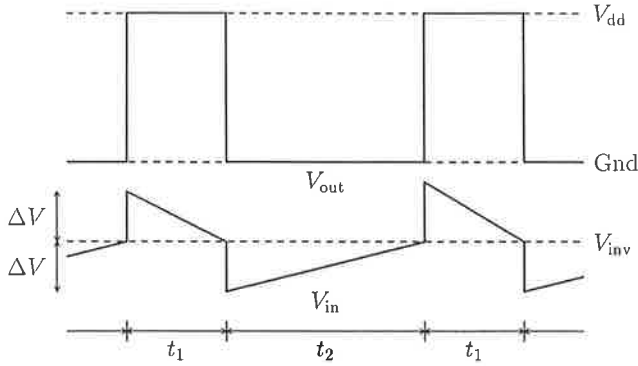
Figure 4. Neuron circuit pulses

Figure 5. Design frame

Figure 6. Differentiator circuit

tween C_1 and C_2, causing the input to rapidly increase by $\Delta V = \frac{C_2}{C_1 + C_2} V_{dd}$. While the output is high, the reset current i_r discharges the input capacitor. When the voltage on the input node falls below V_{inv}, the output switches low causing the input voltage to rapidly decrease by ΔV (thereby resetting the system).

Since the systems being designed are intended to be approximately linear, it is important that the neuron's conversion from the input signal representation (current) to the output representation (pulses) is also linear. Since torque is generated in a motor only when the pulse output is high, the duty cycle (the fraction of the time that the output is high) is the important output measure. This linearity can be proven from charge conservation at the input node. The sources of current at the input node of the circuit are the input current, the reset current, and the capacitive feedback. During one cycle of the circuit (the time from the end of one pulse to the end of the next pulse) the charge due to the capacitive feedback is zero (the charge from the up- and down-going pulse edges cancel). For one cycle, the charge added to the input node due to the input current is $(t_1 + t_2)i_{in}$, and the charge subtracted from the same node due to the reset current is $t_2 i_{reset}$. Since the input voltage is the same at the beginning and the end of the cycle, these two charges must be equal. By setting them equal and rearranging, we get the following equation:

$$\frac{t_2}{t_1 + t_2} = \frac{i_{in}}{i_{reset}}$$

Since i_{reset} is a constant, the duty cycle ($\frac{t_1}{t_1 + t_2}$) is proportional to the input current, and thus the circuit is linear.

The completed system

The design frame implementation is shown in Figure 5. The systems consists of N input devices implemented as complementary-output amplifiers. Each amplifier has a differential input voltage $V_1 - V_2$ and a gain set by V_b. The output currents from these amplifiers are summed onto a pair of wires. These aggregated currents are half-wave rectified and converted into pulses by two neuron circuits. This system converts an arbitrary number of weighted differential voltages into a pair of pulse trains. The dual-rail complementary nature of the system guarantees non-
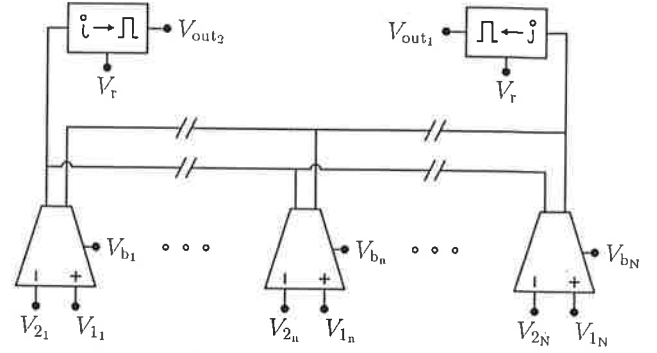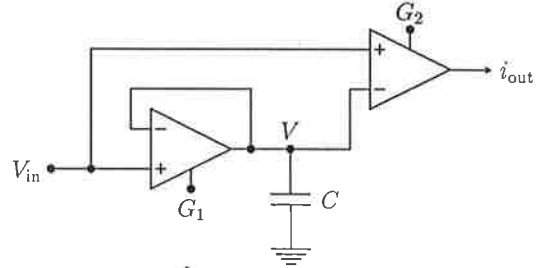
overlapping pulses. The sign of the aggregated currents determines which output (V_{out_1} or V_{out_2}) is firing. By using the pulse outputs from the chip as inputs to the two terminals of a D.C. motor, we can bidirectionally control the motor.

4. The P–D Regulator

In this section we will describe an embedding of a P–D (proportional–derivative) regulator into our design frame. We will first describe the differentiator circuit used in the derivative section of the regulator. Then we will describe the implementation of the complete regulator.

The differentiator circuit

Figure 6 shows the schematic of a differentiator circuit consisting of two transconductance amplifiers and a capacitor. Kirchoff's Current Law at node V requires that the charging current supplied by the amplifier with transconductance G_1 must equal the current flowing into the capacitor, thus

$$G_1(V_{in} - V) = C\frac{dV}{dt}$$

By rearranging this equation, we can derive the following equation for V:

$$V = \frac{1}{\tau s + 1} V_{in}$$

where $\tau \equiv C/G_1$ and s is the Laplace transform of $\frac{d}{dt}$. This circuit is known as a follower–integrator (Mead 1989) and has the same transfer function as an R–C integrator.
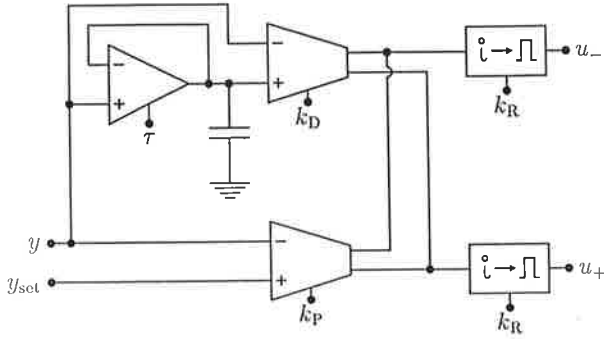
Figure 7. P–D regulator

The derivative is computed by the second transconductance amplifier which takes the difference between the output and the input voltages of the follower–integrator. Using the transfer functions for the follower-integrator and the transconductance amplifier in its linear regime gives the following transfer function for this circuit:

$$\frac{i_{\text{out}}}{V_{\text{in}}} = \frac{G_2 \tau s}{\tau s + 1}$$

For $\tau s \ll 1$ the right side of this equation can be approximated by $G_2 \tau s$ which is the transfer function of a perfect differentiator.

The completed regulator

Figure 7 shows the completed P-D regulator. The proportional section is implemented by using a complementary-output amplifier. The derivative section is implemented using the differentiator circuit with the output amplifier replaced by a complementary-output amplifier. The inputs to the system are a the operating point represented as a voltage y, and a set-point reference voltage y_{set}. The constant τ sets the time constant of the differentiator. The constants k_P, k_D, k_R are used to set the gains for the proportional and derivative sections of the regulator. Given that $\tau s \ll 1$, the outputs of the regulator are

$$u_+ = \frac{k_P}{k_R}(y_r - y) - \frac{k_D \tau}{k_R}\frac{dy}{dt}$$

$$u_- = \frac{k_P}{k_R}(y - y_r) + \frac{k_D \tau}{k_R}\frac{dy}{dt}$$

5. Experimental results

In this section, we describe an experiment in which we compare the operation of our pulsed-output controller with a standard analog D.C. controller. Before going into the details it is relevant to mention something about robot control principles. Recently, nonlinear controllers utilizing the physics of robots has been proposed in general terms (Slotine, 1988), where classical concepts in mechanics like e.g. energy conservation, can be used both in stability
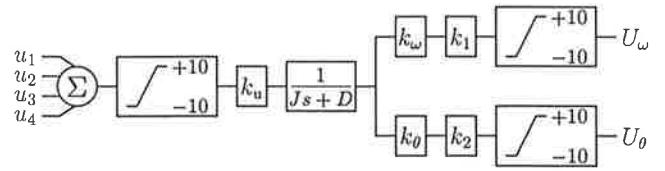
$$k_u = 0.060\,\text{N m V}^{-1} \qquad k_\omega = 5.0 \cdot 10^{-2}\,\text{V rad}^{-1}\,\text{s}$$
$$J = 2.45 \cdot 10^{-3}\,\text{kg m}^2 \qquad k_\theta = 1.01\,\text{V rad}^{-1}$$
$$D = 2.94 \cdot 10^{-4}\,\text{kg m}^2\,\text{s}^{-1} \qquad 0 < k_1, k_2 < 2.0$$

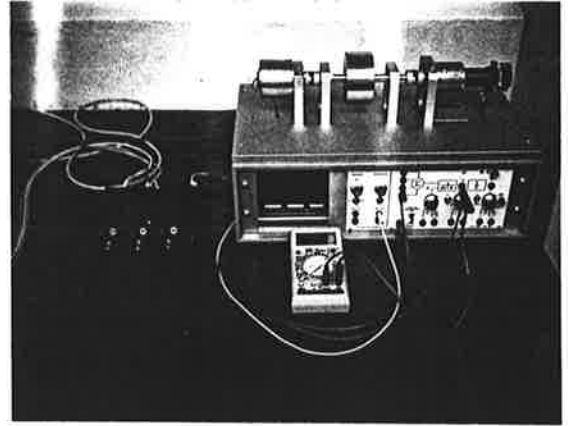**Figure 8.** Servo implementation

Figure 9. Laboratory apparatus

analysis and in the design of trajectory controllers. One specific factor of physical nature is friction, which has become a popular phenomenon of study, since its handling in control is important and nontrivial (Canudas et al, 1987). In the following, this latter aspect is significant.

We use a simple robot model as our physical plant. A robot can, from a motion control point of view, be regarded as a physical setup consisting of a electric motor, a drive system, and an inertial load. In our case (Figure 8) we have a 18 V DC-motor, a gear-box with ratio 1:8, a coupling with some backlash, and an inertial load. The position and speed is measured and available on the front panel. Our experimental setup is shown in Figure 9. The neuron chip was installed in the box to the left of the D.C. servo. The voltmeter is used for coarse speed measurements.

At medium and high speeds both controllers work well. However, at very slow speeds, the pulse controller continues to drive the system while the analog controller fails. The analog controller needs a fairly high reference signal to cause the motor to break free of the static friction inherent in the system. When the reference velocity is decreased, the system has a tendency to "stick". With the pulse servo, however, each pulse contains enough energy for the motor to overcome the static friction. Thus, instead of stopping when the reference velocity is small, the motor continues to move. At high speeds the time-averaged output of the pulse controller operates as the control signal, and the duty cycle controls the speed as expected. At low velocities, however, each pulse generates

a discrete movement of the motor, similar to the operation of a stepper motor.

The behavior has not been fully explored in experiments, but a simple measurement of the speed indicates the following. The conventional controller can hardly function in speeds below 2 rad/s. At these speeds a small disturbance can stop the motion. The neuron servo on the other hand easily can work reliably at below 0.2 rad/s.

6. Conclusions

We have presented a biologically inspired controller applicable to robotics applications. We have described the basic circuitry and design frame for such an implementation and showed how they can be applied to a particular example, namely a P–D regulator. We have also shown the advantages of using a neural-like pulsed output representation to improve slow-speed operation of a friction limited system. Finally, we discussed the utility of exploiting parallelism, aggregation, and redundancy to improve system-level performance given imprecise low-level components.

One advantage of the analog VLSI technology used in this implementation is the ease of combining sensing, intermediate processing, and control (both conventional and adaptive). Sensory systems (visual and auditory) that can be used as the input sensor for feedback systems have been designed (Mead, 1989). Since the control circuitry is implemented using VLSI it can easily be integrated on the same chip as these sensory systems to form integrated sensory systems. We have designed a system combining a visual system that computes the centroid of the 2-D visual field (DeWeerth and Mead, 1988) with two P–D regulators to be used for 2-D visual tracking. We are also working on the extension of the neurally inspired designs to the modeling of high-level biological sensory–motor systems. We are concentrating much of this effort on oculomotor systems and have completed a simplified adaptive model of the vestibulo-ocular reflex (DeWeerth and Mead, 1990).

Speed and accuracy are major issues in the design and control of robots. However, the robot characteristics when it comes to flexibility, dexterity, and intelligence could be even more important. Therefore it is important to consider the function of and interfaces to higher levels of the design when designing the basic motion control. On this higher level adaptivity is desirable both in its classical form and more recent versions of neural networks (Åström and Wittenmark, 1989). Also here, adaptivity can be integrated on the same chip using the proposed design frame. We see this technology as having applications in many areas. A very promising set of applications comes from biology itself. The control of artificial motor systems to mimic the behaviors of animals should be much more attainable if the low-level computational structures are also biologically related.

7. References

K.J. ÅSTRÖM AND B. WITTENMARK (1984): *Computer Controlled Systems*, Prentice Hall, Englewood Cliffs N.J.

K.J. ÅSTRÖM AND B. WITTENMARK (1989): *Adaptive Control*, Addison-Wesley, Reading, Mass.

C. CANUDAS, K.J. ÅSTRÖM, K. BRAUN (1987): "Adaptive Friction Compensation in DC-motor Drives," *IEEE Journal of Rob. and Aut.*, **RA-3**, No. 6, 681-685.

S.P. DEWEERTH AND C. A. MEAD (1988): "A Two-Dimensional Visual Tracking Array," Advanced Research in VLSI, Fifth MIT Conference, March 1988.

S.P. DEWEERTH AND C. A. MEAD (1990): "A VLSI Model of Adaptation in the Vestibulo-Ocular Reflex," *Advances in Neural Information Processing Systems*, *Vol. 2* (Ed.) *D.S. Touretzky*, Morgan Kaufmann, San Mateo, CA.

A. MÅNSSON (1989): "A Study of Muscle Mechanics," Ph. D. thesis, Department of Pharmacology, Lund University, Sweden.

C.A. MEAD (1989): *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, Mass.

J-J. E. SLOTINE (1988): "Putting Physics in Control - The Example of Robotics," *IEEE Control Systems Magazine*, **8**, No. 6, 12-18.

E.A. VITTOZ (1985): "Micropower Techniques," *Design of MOS VLSI Circuits for Telecommunications* (Ed.) *Y. Tsividis and P. Antognetti*, Prentice Hall, Englewood Cliffs N.J, pp. 104-144.