Control of pneumatic robot arm dynamics by a neural network

Patrick van der Smagt*+ Klaus Schulten+
+Department of Physics and Beckman Institute
University of Illinois, Urbana, Illinois 61801

*Department of Mathematics and Computer Science
University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam
The Netherlands

July 25, 1993

Abstract

The trajectory control of a pneumatically driven robot arm resembling a skeletal muscle system is studied. The arm dynamics have been shown to be hysteretic and significantly changing in time due to external influences (Hesselroth et al., IEEE Systems, Man and Cybernetics, in press) thus requiring an adaptive controller.

A highly adaptive feedback algorithm is suggested and shown to control accurately trajectory following tasks.

1 Introduction

When a robot system is designed, the focus generally is a design such that friction, gravity, and payloads can be practically neglected. Therefore, robots are built extremely stiff (i.e., non-compliant) and are equipped with joint actuators which are strong enough to overcome threshold friction, position-dependent gravity, and payloads. The merit of such an approach is that relatively simple control algorithms can be used to position the end-effector with high accuracy.

However, apart from the high cost of such robot systems, their large strength makes their use in environments where humans operate, such as hospitals and household environments, too dangerous. In our research, we concentrate on a new type of robot whose actuators are rubber tubes called *rubbertuators*, which are modeled after skeletal muscle systems. The rubbertuators have a high force-to-weight ratio and are very compliant, such that the robot is safe for operation in direct contact with human operators. In a previous publication [1], we have investigated a visuo-motor coordination system to control the end-effector positioning of this robot arm. Although accurate positioning is possible, the trajectory of the robot arm which connects one endpoint to another is uncontrolled and oscillatory, and completion takes a relatively long time, i.e., about 30 s.

In this article, we introduce a neural feedback system to train the robot to follow a prescribed trajectory in real time. The system coarsely learns to follow the trajectory within a few trials, and reaches accurate positioning after several tens of trials.

In: Proceedings of the World Congress on Neural Networks, Portland, OR, July 11–15, 1993, pp. III-180–III-183.

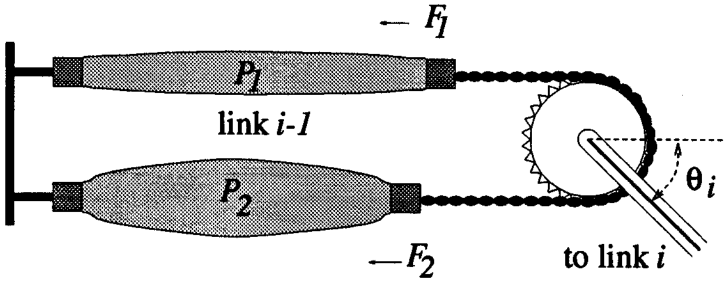

Figure 1: An agonist–antagonist rubbertuator pair connected via a chain, controlling the rotation of a joint. For joints 1, 2, and 3, each agonist as well as antagonist actuator consists of a pair of rubbertuators.

2 The robot system

The dynamic behaviour of any n degree of freedom robot arm can be described by the equation [2]

$$\tau = M(\Theta)\ddot{\Theta} + B(\Theta)\left[\dot{\Theta}\dot{\Theta}\right] + C(\Theta)\left[\dot{\Theta}^2\right] + F(\Theta,\dot{\Theta}) + G(\Theta) \tag{1}$$

where τ is an n-vector of torques exerted by the links, and Θ, $\dot{\Theta}$, $\ddot{\Theta}$ are n-vectors denoting the positions, velocities, and accelerations of the joints. $\left[\dot{\Theta}\dot{\Theta}\right]$ and $\left[\dot{\Theta}^2\right]$ are vectors

$$\left[\dot{\Theta}\dot{\Theta}\right] = \left[\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \ldots, \dot{\theta}_{n-1}\dot{\theta}_n\right]^T, \qquad \left[\dot{\Theta}^2\right] = \left[\dot{\theta}_1^2, \dot{\theta}_2^2, \ldots, \dot{\theta}_n^2\right]^T, \tag{2}$$

$M(\Theta)$ is the matrix of inertia, $B(\Theta)$ is the matrix of Coriolis coefficients, $C(\Theta)$ is the matrix of centrifugal coefficients, $F(\Theta,\dot{\Theta})$ a friction term, and $G(\Theta)$ the gravity working on the joints.

Industrial robots are generally designed to eliminate the interdependence between the joints, such that the robot arm can be regarded as n independent moving bodies. In that case, M and C are diagonal matrices and B is zero. This reduces the $3n$-valued vector field as described by eq. (1) to n independent functions of three variables for which the coefficients have to be found. Also, the link actuators are usually made so powerful that M, C, F, and G can be considered independent of Θ. Thus, eq. (1) is reduced to n independent equations for which simple control methods suffice [3].

However, such simplifications cannot be made for the rubbertuator robot arm. The M, B, C, F, and G are now functions of Θ, and due to the physical properties of rubber change considerably in time. For that reason, we concentrate on designing an adaptive neural method to control the robot in a feedback loop.

2.1 The rubbertuator SoftArm

The robot we use, which has been manufactured by Bridgestone Corporation of Tokyo, is a four-link anthropomorphic manipulator with five degrees of freedom. Its pneumatic actuators, consisting of two or four rubbertuators, are arranged in agonist-antagonist pairs. The rubbertuators are relatively light, such that the arm only weighs 12 kg and can lift 3 kg. Figure 1 shows a rubbertuator agonist-antagonist pair for controlling one joint.

The manufacturer specifies the force F_j exerted by a rubbertuator j at pressure P_j and elongation ε_j as

$$F_j = P_j D_j^2 \left(a_j(1-\varepsilon_j)^2 - b_j\right). \tag{3}$$

Here, D_j is the diameter of the rubbertuator before displacement, and a_j and b_j are tube-specific constants.

The robot can be controlled in *position control mode* or *pressure control mode*. In position control mode, an internal PID controller [2] is used for positioning the joints of the robot. Obviously, this can only give very coarse positioning, and our measurements show that the feedback makes all joints of the arm oscillate with an amplitude of about 1°, never reaching the desired position accurately. In pressure control mode, each rubbertuator can be given a desired pressure. An internal feedback mechanism will then realise this pressure.

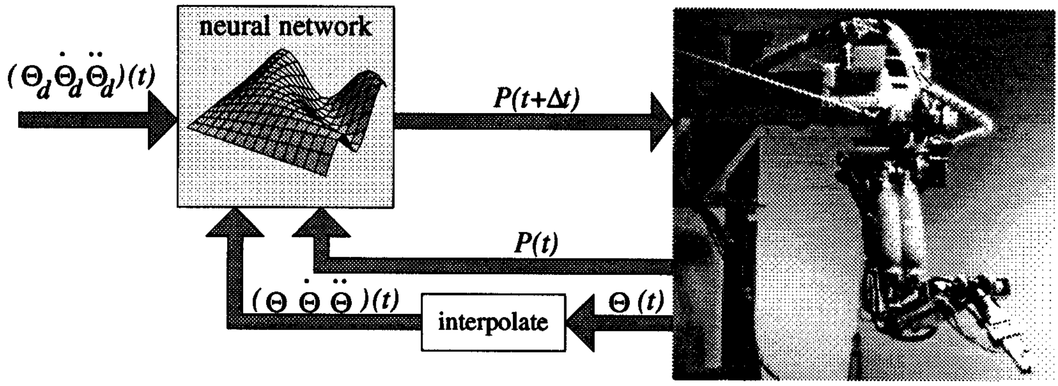

Figure 2: The neural robot control system in a feedback loop with the robot.

However, from eq. (3) it can be seen that the resulting force not only depends on the new pressure, but also on the previous diameter; therefore, the pressure–force and hence pressure–position relation is hysteretic.

Secondly, the use of rubber tubes in a pneumatic system results in non-negligible temperature effects. Rubber is a good insulator, such that temperature changes due to the repeated contraction and expansion, as well as temperature influences from the environment, are only gradually accommodated to. The temperature of the tube has a strong effect on the expansion of the air enclosed in it. In accordance with these observations, we measured a joint rotation in the order of 0.5–1° after the desired pressure has been attained, over a period of 200 seconds.

From these observations, it is clear that an adaptive controller is required to obtain accurate positioning with the robot arm.

3 Controller structure

When two rubbertuators with property (3) are combined into one actuator, the torque τ exerted on the sprocket is

$$\tau = \gamma_1(P_1 - P_2)\theta^2 + \gamma_2(P_1 + P_2)\theta + \gamma_3(P_1 - P_2) \tag{4}$$

where θ is the current joint angle and γ_1, γ_2, and γ_3 are constants depending on $a_{1,2}$, $b_{1,2}$, $D_{1,2}$, and the sprocket radius. The dependency on D means that D must be a parameter of the system. Since we do not have D readily available, we choose $P_{1,2}$ before displacement, which, together with θ, contains the same information.

The task of the robot controller is to generate pressures $P_1(t)$ for the first muscle of a joint, such that a specified trajectory $\left(\theta_d(t), \dot{\theta}_d(t), \ddot{\theta}_d(t)\right)$ is followed. The 'stiffness' $P_1 + P_2$ is always kept constant, such that the pressure from the second rubbertuator can be derived from the first.

The robot control system, which is depicted in figure 2, receives values $\theta(t)$ from the robot at intervals of approximately 20 ms. In order to obtain a estimates of $\dot{\theta}$ and $\ddot{\theta}$ which are not too much noise-sensitive, these values are fitted to orthonormal polynomials following an incremental algorithm described in [4, 5]. Thus we can, with some accuracy, find θ, $\dot{\theta}$, and $\ddot{\theta}$ at each desired time.

The measured pressure $P_1(t)$, $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$, and the desired $\theta_d(t)$, $\dot{\theta}_d(t)$, and $\ddot{\theta}_d(t)$ are input to the neural network. The network then generates a target pressure $P(t + \Delta t)$ which is sent to the robot. The obtained rotation, after the pressure change has been applied, is used as a new learning sample.

3.1 Network structure

The whole control system consists of two programs running on two processors. One program gathers the data from the robot and calculates the joint velocity and acceleration. These data are transmitted to the neural network. The neural network is a feed-forward network trained with conjugate gradient optimisation

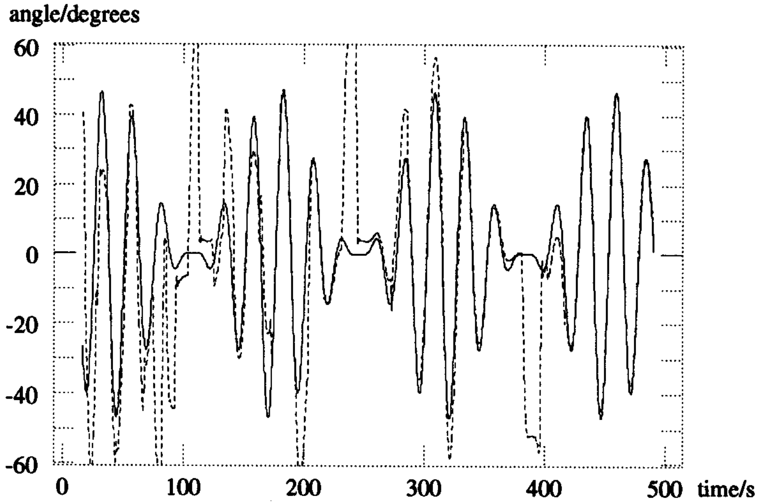

Figure 3: Training the system on a $\sin(t)\cos^2(11t)$ wave. The solid line is the target trajectory, the dotted line the realised trajectory. Initially, the trajectory is reasonable followed where velocity is constant; after only 16 trials, the whole trajectory is followed with an average error of 0.1°.

with Powell restarts [6]. Newly generated samples are continuously added to the bin of available samples, upon which minimisation is performed.

4 Results

The system, with no a priori knowledge, is trained on a trajectory $\sin(t)\cos^2(11t)$. Initially, the trajectory is only followed very coarsely. After 16 trials, however, accurate trajectory following is obtained, with an error of 1° near the extrema, and less than 0.1° on the slope (see figure 3).

5 Conclusions

Is has been demonstrated that the Bridgestone rubbertuator SoftArm can be controlled with an adaptive feedback mechanism to do trajectory following. First results, when the robot is controlled at a moderate speed, show that the network is capable of following the trajectory accurately after a short period of learning.

To follow trajectories at a higher speed, more testing is necessary. Also, we have started the investigation into the simultaneous control of multiple joints of the SoftArm.

Acknowledgments

The authors express their gratitude to the Carver Charitable Trust for support. This work has been partly sponsored by the Dutch Foundation for Neural Networks.

References

[1] T. Hesselroth, K. Sarkar, P. van der Smagt, and K. Schulten. Neural network control of a pneumatic robot arm. Technical Report UIUC–BI–TB–92–15 (*IEEE Systems, Man, and Cybernetics*, to be published), Theoretical Biophysics, University of Illinois at Urbana/Champaign, August 1992.

[2] John J. Craig. *Introduction to Robotics*. Addison-Wesley, Reading, MA, 1986.

[3] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company, New York, 1987.

[4] G. E. Forsythe. Generation and use of orthogonal polynomials for data-fitting with a digital computer. *J. Soc. Indust. Appl. Math.*, 5(2), June 1957.

[5] J. G. Hayes. Curve fitting by polynomials in one variable. In J. G. Hayes, editor, *Numerical Approximation to Functions and Data.* University of London, The Athlone Press, London, 1970.

[6] P. P. van der Smagt. Minimisation methods for training feed-forward networks. Technical Report UIUC–BI–TB–92–17 (submitted to *Neural Networks*), Theoretical Biophysics, University of Illinois at Urbana/Champaign, November 1992.