Adaptive Inertia Compensation Using a Cerebellar Model Algorithm

John Fiala, Ron Lumia

Robot Systems Division
National Institute of Standards and Technology
Bldg. 220, Rm B127
Gaithersburg, Maryland 20899, U.S.A.

Abstract

A method for adaptively compensating for inertia in the control of robotic manipulators is described. The method uses an adaptive network which can be related to a number of contemporary models of the cerebellum. Simulations of a three-link planar manipulator show that the algorithm compensates for inertia during a movement without a priori knowledge. The fact that motions need not be repetitive gives the algorithm significant practical value.

1. Introduction

Recent theories have proposed that the cerebellum is intimately involved in adjusting motor programs for the dynamics of motor acts [6,7,10,11]. Strong evidence exists that the cerebellum has such a role. Of particular interest is the study by Llinas, et al. [3], which found that when the climbing fiber corrections are eliminated from a rat's brain, the rat resorts to a "mudwalking" gait characterized by a loss of compensation for natural inertial forces [3,4]. This indicates that the cerebellum is performing some function for which a permanent program is not stored in the cerebellar cortex and that inferior olive input is necessary for performing this function. Further, the function is essential for the production of the normal, smooth movements associated with locomotion. It is conjectured here that one function of the cerebellum is to compensate for inertia of the body.

A recent theory by Kawato, et al. [11], also proposes that the cerebellum is involved with direct compensation for the dynamics of the motor system. In this approach, circuits associated with the cerebellum learn the dynamics of the system slowly, over hundreds of repetitions, similar to approaches used for robot control [2,5]. The point of departure for this paper is that adaption to task dynamics occurs not over many repetitions, but quickly during the initial performance of the movement. Clearly, long-term learning is important to motor performance, but humans seem to be able to obtain a rapid estimate of the task dynamics from the slightest perturbations. This paper examines the possibility of adaption based on small changes in ongoing movements. To place these ideas in a quantitative domain, control of a three-jointed robotic manipulator will be investigated in simulation.

2. Problem Statement

The position of a mechanical manipulator is given by a vector of joint angles as shown in Figure 1. The set of all possible joint angle vectors is called the *joint-space*. Application of a motor torque at the proximal end of each joint provides an ability to control the movements of the device. The dynamic equation of motion of the manipulator can be written

$$M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + f(\dot{\theta}) = \tau \tag{1}$$

where τ is the motor torque vector, θ is the joint position vector, $\dot{\theta}$ is the joint velocity vector, and $\ddot{\theta}$ is the joint acceleration vector, each of which is a function of time [1]. The vector $c(\theta, \dot{\theta})$ is

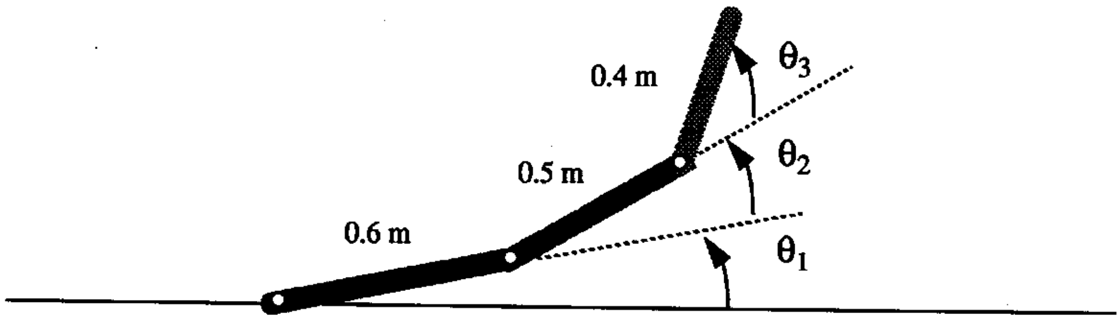

Figure 1. Three-link planar manipulator.

the torque due to centrifugal and Coriolis effects. The vector $f(\dot{\theta})$ is the frictional torque. It is assumed that the plane in which the manipulator operates is perpendicular to gravity such that gravity forces are not explicitly represented in (1). However, the presence or absence of gravity forces does not affect the adaption algorithm.

$M(\theta)$ is the *inertia tensor* of the manipulator [1,6]. This inertia is determined by the kinematic, inertial, and mass properties of the manipulator. For illustrative purposes, the device of Figure 1 is modeled as having lumped masses of 8.0, 5.0, and 3.0 kg, located at the distal end of the links attached to joints one, two, and three, respectively. The length of each link is shown in the figure. Note that $M(\theta)$ is a complicated function of joint position [1]. That is, $M(\theta)$ is a *field* defined over the joint-space since its value depends on the location θ in that space.

For the manipulator to be moved along a desired trajectory, it is at least necessary to provide a command sufficient to overcome the inertia [1]. Suppose that $c(\theta, \dot{\theta}) = 0$ and $f(\dot{\theta}) = 0$, and the manipulator is to move along a trajectory given by $\ddot{\theta}^*(t)$. Then, for a control input of

$$\tau(t) = M(\theta)\,\ddot{\theta}^*(t) \tag{2}$$

the acceleration of the manipulator will be exactly $\ddot{\theta}^*(t)$, as can be seen by substituting for τ in (1). Without including the inertia compensation in the control, the acceleration of the manipulator would be significantly different from what is desired. Unfortunately, $M(\theta)$ is not easily calculated. Determination of inertial and mass properties of a real mechanism can be difficult and these properties change as the manipulator handles unknown objects. A method of on-line estimation is proposed here to overcome these difficulties.

First, note that for (1)

$$\frac{\partial \tau}{\partial \ddot{\theta}} = M(\theta)$$

such that,

$$M(\theta)\,(\ddot{\theta}(t) - \ddot{\theta}(t-h)) \approx \tau(t) - \tau(t-h) \tag{3}$$

when h is small. By letting $M(\theta) = I + R(\theta)$, where I is the identity operator, (3) can be written

$$R(\theta)\,(\ddot{\theta}(t) - \ddot{\theta}(t-h)) \approx (\tau(t) - \tau(t-h)) - (\ddot{\theta}(t) - \ddot{\theta}(t-h))$$

which shall be denoted simply as

$$R\alpha = \delta\ . \tag{4}$$

3. Adaptive Compensation

To determine the inertia at a point in the state-space, an estimate of the locally-linear operator R can be obtained from acceleration and torque change sample vectors (α,δ) of relation (4). To accomplish this, an adaptive network as depicted in Figure 2 is used. Here, R is represented by a set of adjustable weights r_{ij}. These weights multiply the input acceleration changes α. The results are summed to produce the estimated torque/acceleration change difference $\hat{\delta}$. Note the similarities of this model to Fujita's adaptive filter model of the cerebellum [10], where the weights are implemented by Purkinje cells of the cerebellar cortex.

Adjustment of the weights for the network is made according to the rule

$$\Delta r_{ij} = \beta\,(\delta_i - \hat{\delta}_i)\,\frac{\alpha_j}{\alpha^t\alpha} \tag{5}$$

where δ_i is the actual value and β is a gain for the adjustment. This is a variation of a learning algorithm described in [13]. This is also a generalization of the learning algorithm used by Albus for the Cerebellar Model Articulation Controller (CMAC) [2,9]. In the CMAC algorithm, $\alpha=(1,..,1)$. In accordance with biological studies [3,7,8], the weight adjustments in the cerebellar cortex are made via climbing fibers from the inferior olive. With the current model, estimates $\hat{\delta}_i$ would be obtained at the cerebellar nuclear cells, which project to the inferior olive [14]. The inferior olive also obtains the actual values δ_i based on the muscle command change and the sensed acceleration change. From these values a correction for the Purkinje cells can be computed according to (5).

This adaptive compensation approach can be related to cerebellar models in a number of other ways. For example, other approaches are possible to the problem of estimating R from samples of (4), including inverse or least squares techniques [5]. A recursive form of the least squares approach is analogous to Kalman filtering which has been proposed as the functional role of the cerebellum by some authors [12]. In another model, called Tensor Network Theory [6,7], the cerebellum performs covariant-to-contravariant transformations according to an internal representation of the embedding metric tensor associated with coordinate transformation. In the adaptive compensation scheme presented here, the cerebellum also performs a covariant-to-contravariant transformation, in the form of equation (2). However, the internally represented tensor is the inertia [6].

In using the R estimate in the generation of control signals, the weights r_{ij} must multiply the

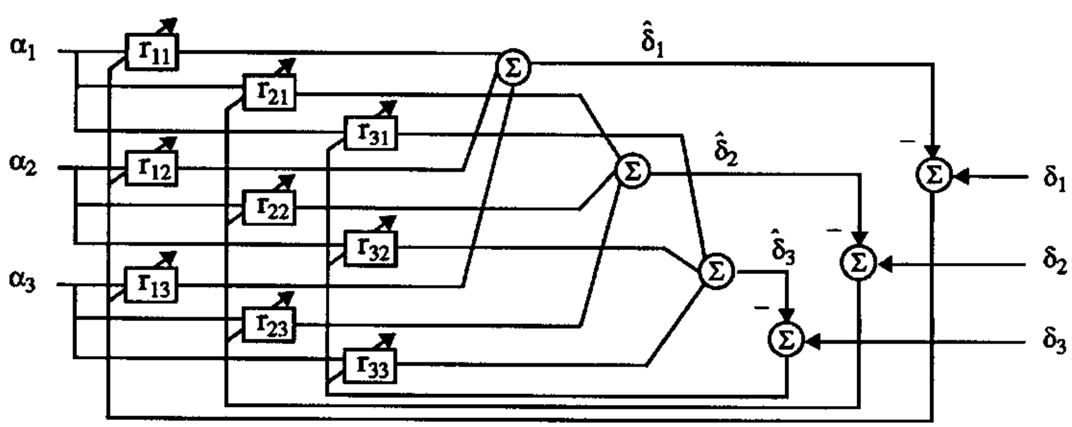

Figure 2. Adaptive filter for estimation of R.

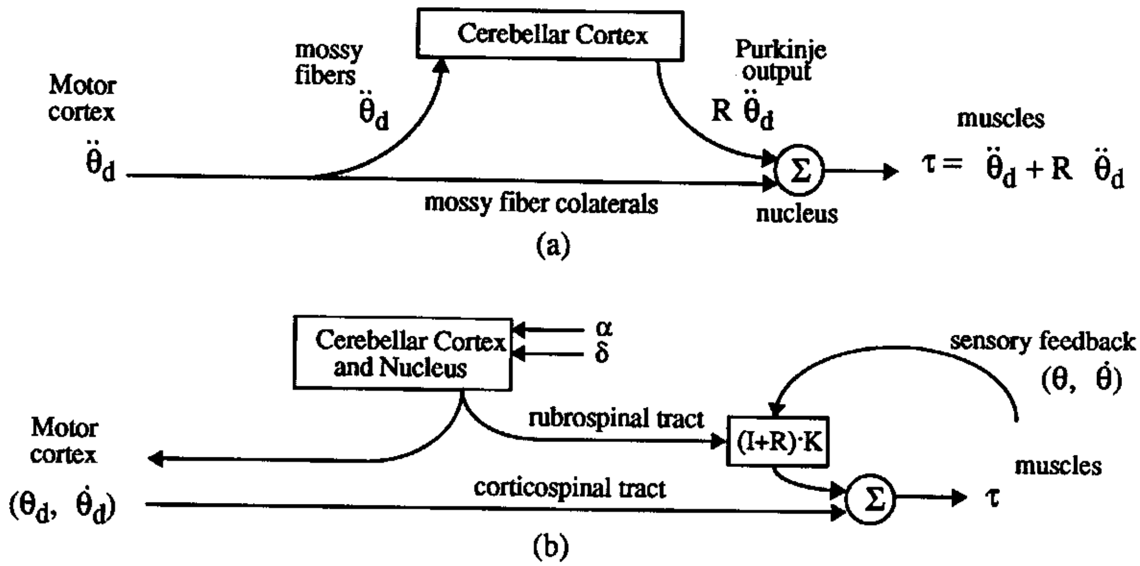

Figure 3. Models of cerebellar control paths.

acceleration command signal $\ddot\theta^*$. If this is to take place in the cerebellar cortex, then there must be a duplicate set of weights adjusted simultaneously with the set that receives the acceleration changes. This second set, identical to the first, could receive the desired acceleration command and form a sidepath model similar to that of the Tensor Network Theory [7] as depicted in Figure 3(a). Another method for utilizing R is to have the inertia compensation project to the reflex loops [4]. In this model, shown in Figure 3(b), the r_{ij} weights would more likely be represented by a combination of Purkinje cell and nuclear cell action. The resulting weight would modify the strength of lower level reflexes, such as the stretch reflex, as well as modify the command from the cortex.

4. Simulation Results

For the purposes of simulating the adaptive inertia compensation, this feedback control model is used.

$$\ddot\theta^* = K_p(\theta_d - \theta) + K_v(\dot\theta_d - \dot\theta) + \ddot\theta_d$$

$$\tau_{act} = \ddot\theta^* + R\ddot\theta^* \tag{6}$$

Mathematically, this algorithm represents either model of Figure 3.

The manipulator is controlled by a two-stage controller consisting of a trajectory generator and a feedback controller. The trajectory generator computes a sequence of desired position, velocity and acceleration vectors designed to move the joints to the appropriate goal position θ_{goal}. The trajectory generator is given by the dynamical system

$$\ddot\theta_d = (-32)\,\dot\theta_d - (640)\left(\frac{t^2}{1+t^2}\right)(\theta_d - \theta_{goal}) \tag{7}$$

The reference signals, $\ddot\theta_d$, $\dot\theta_d$, and θ_d, are updated every control cycle, which is repeated every 2.5 ms. Initially, the simulation starts with every $r_{ij} = 0$. Each control cycle, the weights are adjusted

using equation (5). The resulting weights multiply $\ddot{\theta}^*$ on the next cycle to produce the inertia compensated control of (6).

The values of friction used in the simulation are 4 Nm static Coulombic friction, 2 Nm dynamic Coulombic friction, and 7 Nm/rad/s viscous friction, for each joint. Values of 640 Nm/rad for K_p and 36 Nm/rad/s for K_v are used for each joint. To test the robustness of the adaptation, zero-mean noise of $\pm 10\%$ is added to the acceleration feedback used by the learning rule. A value of 0.2 is used for the learning gain β.

The results of a simulation are shown in Figure 4. The manipulator is moved from the initial joint position of (11°,20°,42°), (approximately the position shown in Figure 1,) to a goal position of (90°,60°,75°). Then, after a pause, the manipulator is moved to a goal position of (0°,0°,0°). The ideal trajectory shown in the figure is that of θ_d generated by (7). Also shown for comparison is the uncompensated control, (6) with R=0, and the model-based control which uses R as computed from the manipulator model. Note the closeness between the model-based method and the adaptive method which has no a priori information.

5. Conclusions

A method for adaptive inertia compensation for serial, articulated manipulators has been presented. The technique is simple enough for real-time application and does not require repetition of movement. An interesting aspect of the approach is that is incorporates features from a number of seemingly disparate theories of the cerebellum. Motor system dynamics are learned by the cerebellum as in Kawato's theory, but the part of the dynamics learned corresponds to a tensor as in the Tensor Network Theory. Also, the learning algorithm chosen is a generalization of the basic CMAC algorithm. Learning of this form might also be carried out by a recursive least squares approach analogous to Kalman filtering as in the Kalman Filter Theory of the cerebellum. This shows that these theories are in fact more closely related than they initially appear. One can speculate that the cerebellum's primary function is locally-linear estimation relevant to motor control. The simplicity of linear operators may account for the highly regular structure of the cerebellar cortex. Further, each distinct region of the cerebellar cortex may be dedicated to estimating a particular class of operators relevant to the motor behavior of the organism. Clearly, much future research is required to discern all the complexities associated with cerebellar function.

6. References

[1] Craig, J. J., Introduction to Robotics: Mechanics and Control, Addison-Wesley, Reading, MA, 1986.

[2] Miller, W. T., Glanz, F. H., Kraft, L. G., "Application of a General Learning Algorithm to the Control of Robotic Manipulators," Int. Jour. Robotics Research, Vol. 6, No. 2, 1987.

[3] Llinas, R., Walton, K., Hillman, D. E., Sotelo, C., "Inferior Olive: Its Role in Motor Learning," Science, Vol. 190, p. 1230, 1975.

[4] MacKay, W. A., Murphy, J. T., "Cerebellar Modulation of Reflex Gain," Progress in Neurobiology, Vol. 13, pp. 361-417, 1979.

[5] Raibert, M. H., "A Model for Sensorimotor Control and Learning," Biol. Cybern., 29: 29-36, 1978.

[6] Arbib, M., Amari, S., "Sensori-Motor Transformations in the Brain (with a Critique of the Tensor Theory of Cerebellum)," Jour. Theor. Biol., Vol. 112, p.123-155, 1985.

[7] Pellionisz, A., Llinas, R., "Tensor Network Theory of the Metaorganization of Functional Geometries in the Central Nervous System," Neuroscience, Vol. 16, No. 2, 1985.

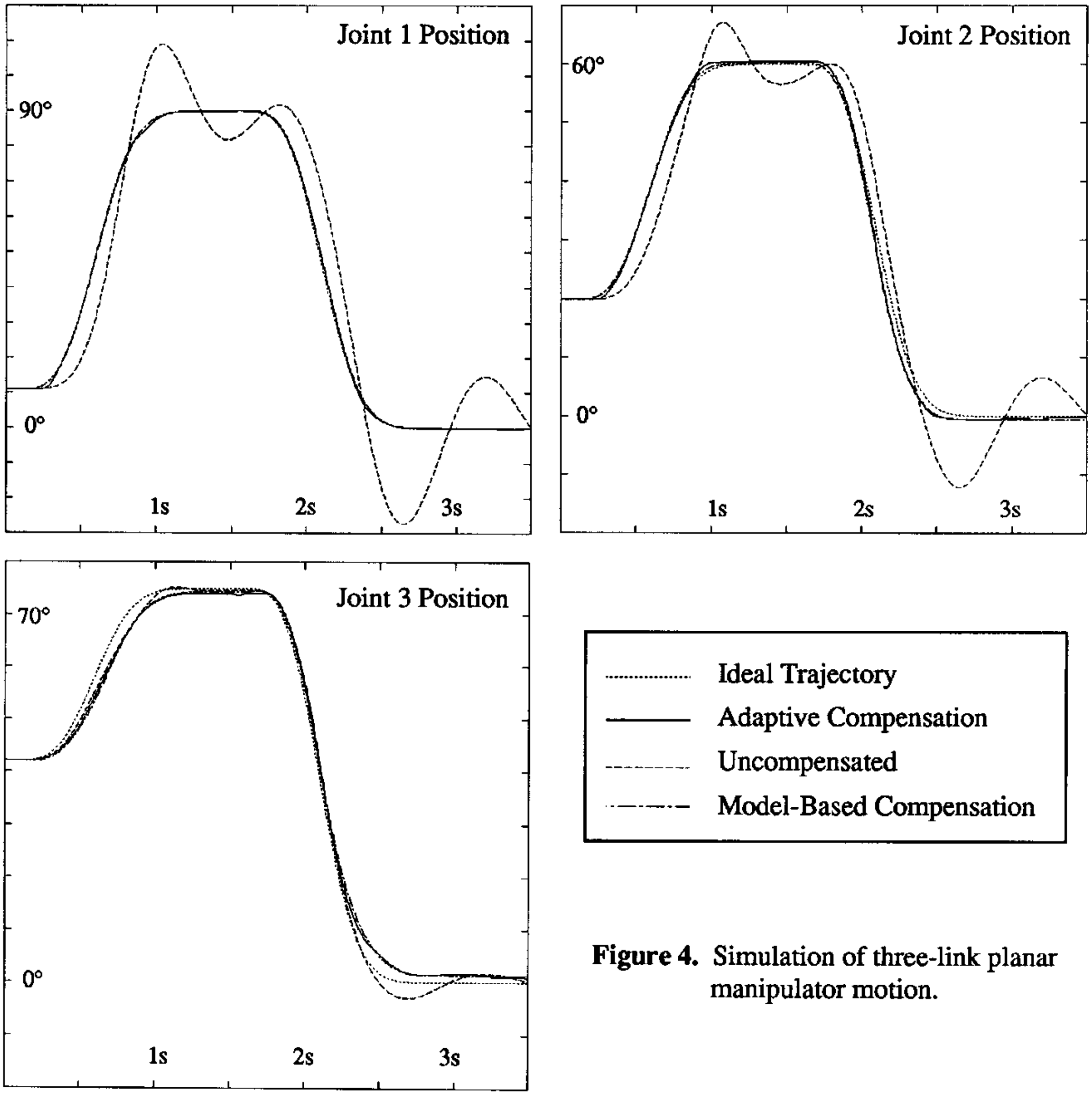

Figure 4. Simulation of three-link planar manipulator motion.

[8] Albus, J. S., "A Theory of Cerebellar Function," Math. Biosciences, 10: 25-61, 1971.

[9] Albus, J. S., "Data Storage in the Cerebellar Model Articulation Controller (CMAC)", Jour. Dyn. Sys., Meas. & Control, Sept., 1975

[10] Fujita, M., "Adaptive Filter Model of the Cerebellum," Biol. Cybern., 45: 195-206, 1982.

[11] Kawato, M., Furukawa, K., Suzuki, R., "A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement," Biol. Cybern., 56, 1987.

[12] Paulin, M., "A Kalman Filter Theory of the Cerebellum," in Dynamic Interactions in Neural Networks: Models and Data, Arbib, M., ed, Springer-Verlag, New York, 1989.

[13] Mycielski, J., "Can Mathematics Explain Natural Intelligence?" Physica, 22D: 366-375, 1986.

[14] Brodal, A., Kawamura, K., The Olivocerebellar Projection: A Review, Springer-Verlag, Berlin, 1980.