Explicit Computation versus Mechanics

Satyajit Rao

May 17, 1993

Abstract

Given the task of moving your arm and picking up an object, the conventional approach requires explicit computation of the inverse dynamics. A recent proposal "The Equilibrium point hypothesis" offers an alternative approach that does not explicitly compute torques but rather, lets the mechanics of the actuators generate the torques. The purpose of this area exam paper is to compare the two approaches. The paper is organized in the following manner:

1. A description of the conventional approach to arm movement - gathered from Hollerbach[5] Hollerbach[6] Winters[11] Hogan[4] and Uno[9].
2. A description of the Equilibrium point approach - gathered from Bizzi[3] Bizzi[1].
3. A final comparison of the two approaches.

1 The conventional approach to arm movement

Consider the task of picking up a cup and drinking from it while you hold a plate in the other hand. The conventional approach to this problem first involves locating the cup in space. This information is passed on to a trajectory planner that generates a plan for picking up the cup. The planner must produce a trajectory profile to reach for the cup, a strategy for holding the cup and a trajectory profile to get it to the mouth. This rough plan is then executed by a controller whose responsibility it is to produce the appropriate forces and torques to achieve the desired trajectory while compensating for any perturbations caused by unmodelled aspects of the environment. The separation of planning from control is central to the conventional approach to arm movement. However as we shall see, once dynamic variables enter the planning process and feedback is used to alter plans, the distinction between planning and execution becomes blurred.
We now consider the Trajectory planning process in more detail. Figure 1 summarizes the issues in Trajectory planning in the form of an AND-OR tree.

1.1 Trajectory Planning

Consider the two link manipulator in figure 2. The position of the end point can be described by a pair of Cartesian coordinates (x y). The position of the arm can be specified by two variables ($\theta1$ $\theta2$).
These are called kinematic variables, they describe the geometry of the arm. ($\theta1$ $\theta2$) are called intrinsic kinematic variables because they are directly controlled by the system, whereas (x

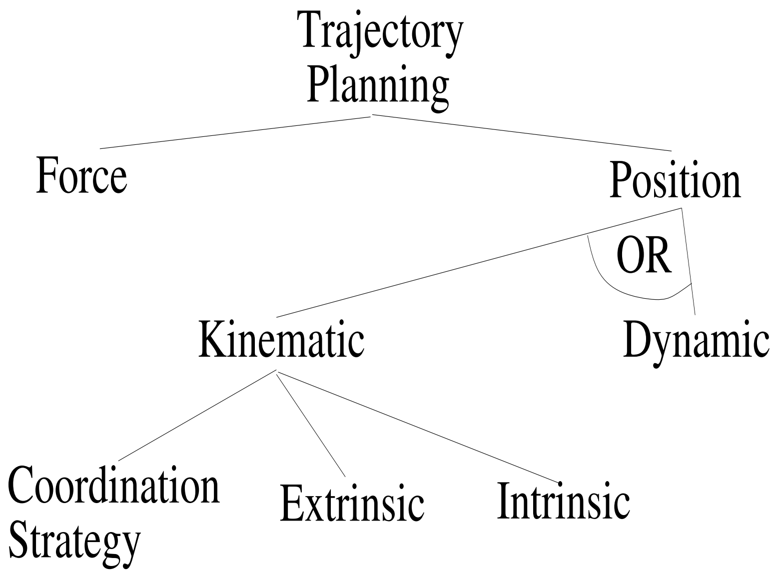


Figure 1: Issues in trajectory planning

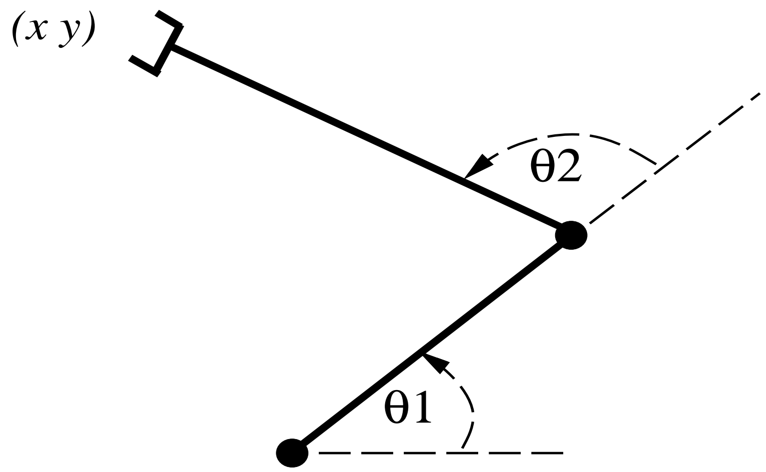


Figure 2: The kinematic variables

y) are extrinsic kinematic variables, they describe the coordinates of the end point. Clearly there is a direct mapping from intrinsic to extrinsic variables because the end point position is directly determined by the arm position, but the converse is not true because there may be more than one arm position that achieves the same end-point position. Kinematic redundancy while making inverse kinematics difficult can be an advantage when obstacles are present because of the multiple ways to achieve the same end effector position. One goal of Trajectory planning is to determine some time evolution of the kinematic variables that will get the end effector from the initial to the final position. This takes care of the position degrees of freedom, but we must also plan for the force degrees of freedom. For example when writing on the blackboard the hand has 5 degrees of position freedom and one degree of force freedom (into the board), the planner must generate some time profile for the force degree of freedom as well [5]. The following discussion will consider only the position aspect of Trajectory planning.

Trajectory Planning: kinematic versus dynamic variables

One approach to trajectory planning takes only the kinematics of the manipulator into account.

$$\int_0^d \left| \frac{d\ddot{\mathbf{r}}}{dt} \right|^2 dt \tag{1}$$

Where

d = Duration of the movement
$\mathbf{r}$ = End effector position

A cost functional is defined that depends solely on the geometry of the manipulator. The external path $\mathbf{r}(t)$ that minimizes this functional is chosen as the desired path. For example the minimum jerk hypothesis by Hogan[4] finds a path r(t) from the initial to final position by minimizing the rate of change of acceleration. Notice that the cost function takes only the geometry of the arm into account, it does not consider the dynamic properties of the actuators or any external forces, for example from a load that the arm might be carrying. This view is in accordance with traditional view of separating planning from control. The advantage is its simplicity - it does not have to deal with dynamics. However, ignoring the dynamics also puts it in peril because the plan may be inappropriate given the forces involved.
An alternative view of trajectory planning is that the planner must consider external forces and actuator dynamics to determine a trajectory. The trajectory for bringing the cup to the lips must take into account the weight of the cup, and the fact that you are balancing a plate in the other hand. Thus a cost functional is defined in terms of dynamic variables, and the external path $\mathbf{r}(t)$ that minimizes this functional is found. For example Uno, Kawato, and Suzuki [9] define a functional that minimizes the rate of change of torque.

$$\int_0^d \sum_i \left| \frac{d\tau_i}{dt} \right|^2 dt \tag{2}$$

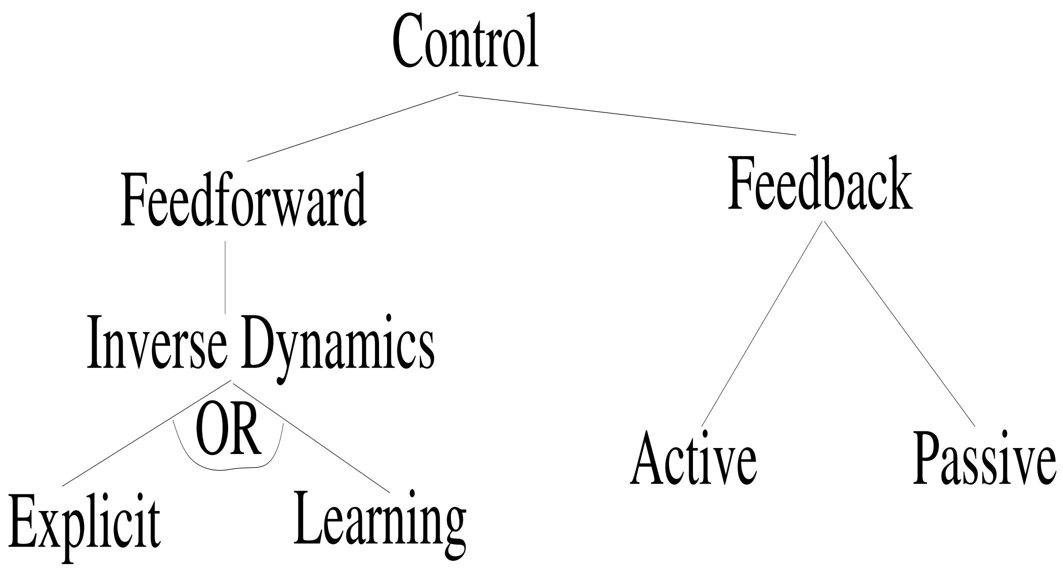


Figure 3: Issues in control.

Trajectory Planning: Extrinsic versus Intrinsic coordinates
Trajectory planning in terms of the extrinsic end-point coordinates allows external constraints to be very easily captured in coming up with a trajectory, because we can directly see the obstacles in terms of extrinsic coordinates. The problem is that the input to the controller must be in terms of intrinsic coordinates, because those are the variables directly under its control. This means that the external path profile must be converted to an intrinsic joint profile. As noted earlier, this may not be a unique transformation.
The advantage of planning the trajectory in intrinsic coordinates (joint space) is that any continuous path that connects start position with end position in joint space is a valid solution, this was not true in extrinsic space because a continuous path in real space may not be realizable as a sequence of arm positions. The disadvantage of planning in joint space is that the all the obstacles in the external environment must be represented in joint space by using inverse kinematics to transform their coordinates.

Trajectory Planning: Coordination Strategies for kinematic variables
We have seen that the time profile for the kinematic variables can be derived by optimizing some kinematic or dynamic cost function. However we can arbitrarily pick some coordination strategies. A very simple coordination strategy is “Straight line interpolation”. The path is simply a straight line in either extrinsic or intrinsic coordinates between the initial and final positions, the velocity profile is chosen as desired. This is just one of a multitude of coordination strategies possible, for example there is no need for the velocity profiles of the two joints to start and end at the same time, the Staggered joint interpolation coordination strategy includes a delay and a scaling factor in the velocity profile of one of the joints.
We now consider the next stage after trajectory planning, namely Control. Figure 3 summarizes the possibilities for Control in the form of an AND-OR tree.

2 Control

The function of the controller is to generate the appropriate torques to achieve the desired trajectory.

Feedforward control
Dynamics tells you the effect of torques and forces on the kinematic variables. Inverse dynamics tells you the torques needed to produce a desired joint angle profile. Sometimes there is more than one way to produce a torque at a joint, in such cases we run into the "Force distribution" problem where one has to decide how the responsibility of producing the given torque at a joint is going to be distributed among the various actuators at that joint. Computing the torques given the desired joint profiles does not completely solve the control problem because:

- The inverse dynamics model may be inaccurate. If the dynamics of the real system is given by R and faulty model of the system by $\hat{R}$. Then the deviation of the actual trajectory from the desired one is:

  $$\epsilon(t) = \theta_d(t) - R\hat{R}^{-1}(\theta_d(t)) \tag{3}$$

  One reason that the dynamics model of the system may be inaccurate is that it is being learned by the system. There are clear advantages for a robot to learn its arm dynamics rather than explicitly compute it. One reason suggested by Hollerbach [5] and Hogan [4] is that inverse dynamics is extremely hard to compute, its complexity increases dramatically with the number of degrees of freedom of the arm. While it is true that inverse dynamics computations are cumbersome, Lathrop [7] has shown a way to extract inherent parallelism in the computations so that the computations increase in only the log of the number of joints. I believe the real reason for not using inverse dynamics computations is that the robot must be adaptive. For example if one of the actuators malfunctions or a piece of the arm breaks off, the robot must be capable of learning the new dynamic model.

- An inaccurate initial state or unexpected perturbations from the environment can cause a deviation from the planned trajectory.

To handle deviations of the actual trajectory from the planned trajectory, it seems that it is insufficient to simply specify the joint torques in a feedforward manner, but some mechanism is needed that senses the *state* of the arm and takes corrective measures to correct the error between the actual and desired trajectory profiles.

Feedback control
The most general form of feedback takes the state of the arm, the desired trajectory, and the dynamic model of the system into account to generate the corrective torques.

$$\tau(t) = f(\theta_i, \dot{\theta}_i, \theta_{di}, \dot{\theta_{di}}, \hat{R}) \tag{4}$$

An important consequence of feedback control is that it relieves the feedforward mechanism of some of its complexity. Now we can ignore some of the interaction torques,

thus making the dynamics simpler but more inaccurate, and have feedback generate compensatory torques, thus effectively treating the unmodelled torques as perturbations.
Active Feedback involves sensing the state of the arm, comparing it to some desired state and sending some signal based on this comparison. A simple example is Positional Derivative Control.

$$\tau(t) = -K_p(\theta - \theta_d) - K_v\dot{\theta} \tag{5}$$

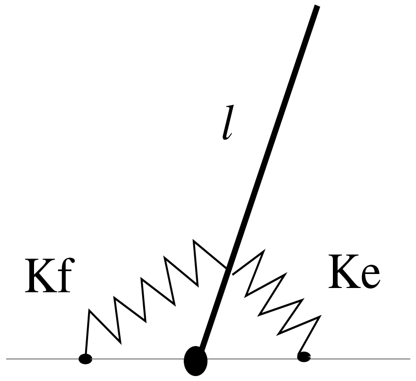
The first term tries to reduce the difference between actual state and desired state, the second term penalizes large velocities. All active feedback mechanisms will have a gain and a delay. Large gains make the effect of the system dynamics negligible. However we also want short delays in order to avoid oscillations. Negative feedback loops with large gains and short delays are very useful, but difficult to make. The human arm has a number of active feedback loops involved in the control of the arm.
Passive Feedback is like active feedback, except that there is no explicit sensing of the control variable, or its comparison with some desired value, the feedback is implicit in the mechanics of the system. We will see that this is one of the central ideas behind the Equilibrium point approach.
Task-Level Feedback While biological mechanisms certainly have the above two types of feedback they display much more sophisticated behavior that takes into account the task being performed. For example a perturbation of the hand while lifting the cup, may cause the controller to change the positions of the leg and torso so that the cup doesn't fall, Showing that the feedback mechanism is much more sophisticated then a local reflex.

3 Equilibrium Point Control

As the name indicates this is primarily a suggestion for the control stage. The assumption is that trajectory planning has already been done, and we have some trajectory profile to follow. Bizzi and colleagues point out the weakness of the inverse dynamics computation for control, namely that it is unrealistic to expect the system to have an accurate model of its dynamics at all times because even a 5% error in the estimation of the inertial parameters leads to instability. The key idea behind Equilibrium point control is that the controller doesn't bother about explicitly specifying the torques, but generates movements by shifts of posture. The torques are automatically generated because of the springlike nature of the muscles. Whereas the inverse dynamics approach considered actuators to be perfect torque producers irrespective of how they are implemented, the Equilibrium point approach makes use of the mechanical properties of the actuators to simplify computation. Bizzi, Hogan, Mussa-Ivaldi, and S. Giszter make two claims about the advantages of the Equilibrium point control over the inverse dynamics approach.

- By using the springlike properties of the muscle to produce restoring torques to reach equilibrium points, and specifying the trajectory as a sequence of equilibrium points, the computations for control are drastically simplified.

- Equilibrium point control is good not only for position control but also force control. In robotics the force control is done by sensing the force of contact and then using that in a feedback loop to adjust the force. Not surprisingly this active feedback loop can have oscillations resulting in "contact instability". The claim is that the



$$\Theta_e = \frac{Kf}{Kf + Ke} \Pi$$

Figure 4: A single link manipulator at its equilibrium position

equilibrium point approach avoids this because of the springlike property of the actuator.

In what is to follow we will examine the above claims by first considering the control of posture and movement in a single link manipulator, this simple example should make the fundamental idea behind equilibrium point control clear. Then we will consider the control of posture and movement (slow and fast) for a multiple link manipulator. Then we consider the experimental evidence offered in support of "Equilibrium Point Control" as a mechanism in biological systems. Finally we summarize the advantages and disadvantages of using equilibrium point control instead of doing inverse dynamics.

The control of posture and movement for a single joint:
The basic idea underlying equilibrium point control can be easily understood by considering the problem of controlling a single link manipulator. See figure 4.
In the absence of external loads a limb is in static equilibrium (in a particular posture) when the torques generated by the opposing muscles cancel out. Perturbing the limb from this equilibrium position will cause the springlike muscles to automatically generate restoring forces that take it back to the equilibrium position. In fact the form of the restoring torque is like Positional derivative Control in equation 5.
The condition for the equilibrium point to be a stable one is very simple, the stiffness constants K_p and K_v in the above equation should be positive. The important point here is that the above feedback is *passive*, it happens instantaneously because of the mechanical properties of the muscle, it does not happen via some mechanism that senses the current position, compares the difference and generates a signal (such an active feedback mechanism may also be at work in parallel but we will come to that later). In this simple one joint case, given some desired joint angle, the controller adjusts the stiffness and rest lengths of the flexor and extensor muscles so that the the length tension curves intersect at the desired equilibrium point. Note that different Muscle tensions can achieve the same equilibrium position.
What happens in the presence of gravity? The final position will not be exactly equal to the equilibrium position, because there must be a non-zero net torque applied by the springs to counteract the gravitational torque, however this deviation from equilibrium can be made arbitrarily small by increasing the stiffness of the springs. Thus the ability to

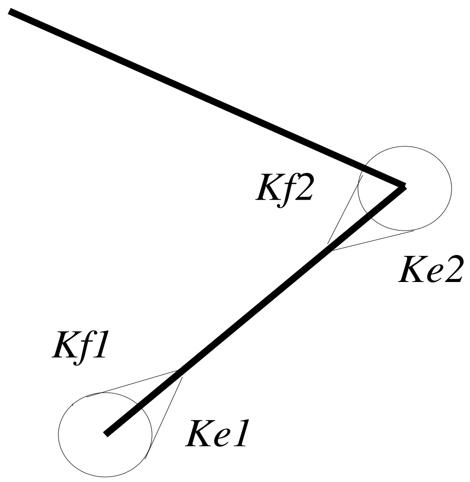


Figure 5: The two joint arm with four springs.

change the mechanical behavior of the system by modulating the stiffness can be used to come close to the desired position without having to explicitly precompute the effect of the gravitational force.

$$\theta + \frac{lmg}{2(K_f + K_e)} \cos\theta = \frac{K_f}{K_f + K_e}\pi \tag{6}$$

When we increase the stiffness of the springs we get.

$$\theta_{equilibrium} \approx \frac{K_f}{K_f + K_e}\pi \tag{7}$$

Movement of the link is brought about by defining a moving equilibrium point - this trajectory is called the *virtual trajectory.* The restoring forces try to make the arm follow the virtual trajectory, but the actual trajectory may be different.

Multijoint posture and movement:

Posture in multiple joints:
As before we can set the spring stiffnesses at both joints to achieve some desired position, and more than one set of stiffnesses can achieve the same equilibrium position.
In the single joint case, a small perturbation of $\delta\theta$ produces a restoring torque $\delta\tau$ that causes the link to regain its posture. The stiffness is defined as $\delta\tau/\delta\theta$. Similarly in the case of the two link manipulator a perturbation of $\delta\mathbf{r}$ at the tip causes a perturbation of $\delta\theta$ at the joints, causing a restoring torque $\delta\tau$ at the joints. If we look at the restoring force at the endpoint $\mathbf{F}$ then we have an endpoint stiffness $\mathbf{K}$ such that $\mathbf{F} = \mathbf{K}\delta\mathbf{r}$ where $\mathbf{K}$ is of the form:

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix} \tag{8}$$

This is the end point stiffness matrix. In order for the force $\mathbf{F}$ to be like a spring restoring force, it must satisfy two conditions. Firstly it must vary with position, i.e $\mathbf{F} = \mathbf{F}(\mathbf{r})$, and secondly it must be conservative, i.e $\nabla \times \mathbf{F} = 0$, which implies that the off diagonal terms of the stiffness matrix must be 0. Stiffness matrices measured in different subjects had this property. Clearly this endpoint stiffness matrix is related to the stiffnesses of the springs at the joints. In fact the endpoint stiffness is a function of the joint stiffness *and* the position of the arm, This dependence on position implies that we can change the endpoint stiffness by simply changing the position of the arm without changing the stiffness of the springs. This implies that if there is kinematic redundancy then the controller can exploit it by being able to choose among the different stiffnesses associated with the multiple arm configurations for the same end effector position, this is very useful in load compensation. Movement with multiple joints:

Like the single link case, movement is produced by a shift of posture, i.e a virtual trajectory $\theta_{1e}(t)\theta_{2e}(t)$ is specified. How is this virtual trajectory profile chosen? Equilibrium point control has to deal with the same problem as conventional approaches in the choice of a trajectory profile. Interestingly the choice of a straight line endpoint virtual trajectory with a bell shaped velocity profile, produces slightly curved actual paths (in simulations by Flash) that have a very close match with experimentally observed paths in humans. The arm tries to follow this virtual trajectory but the perturbations caused by the Coriolis, Gravitational, and Reaction forces make the actual trajectory deviate from the virtual trajectory. As we saw in the simple case of the single link, the ability to change the mechanical behavior of the system by modulating the stiffness can be used to come close to the desired trajectory without having to explicitly compute the interaction forces. Slow movements can be handled by this method, but not fast movements because the centripetal and Coriolis forces increase as the square of the velocity (because of the $\dot{\theta}_1^2$, $\dot{\theta}_2^2$, and $\dot{\theta}_1\dot{\theta}_2$ terms), thus to scale the velocity by c, the stiffnesses would have to be scaled by c^2 to maintain stability. Given some limit on the stiffness values, this is not a viable strategy for fast movements. Data from human movements show an initial *decrease* in stiffness and then an increase in stiffness towards the end of the movement. Another way to handle fast movements may be to change the virtual trajectory so that large deviations between virtual and actual trajectory cause fast movements. The problem with this is that the virtual trajectory is no longer the same as the actual trajectory, the trajectory planner becomes more complicated, because it must take the dynamics of the system into account. A forthcoming Ph.D thesis by Terence Sanger shows how the required virtual trajectories can be learned by making perturbations to the simplest virtual trajectory (= the desired trajectory) and observing the results. This experience is used to choose appropriate perturbation in the future that will produce the desired velocity (we encountered the idea of using learning instead of explicit modeling before while considering an alternative to doing inverse dynamics explicitly).
A third way to produce fast movements has been proposed by McIntyre [8] in his Ph.d thesis. It involves a simple combination of the alpha and lambda models, as shown in figure 6. The combination is augmented with a desired velocity signal which is used in the velocity feedback loop of the reflex. The desired velocity signal is used by the reflex feedback to set the desired position signal of the muscle. The problem with this seems to be that because the velocity control is being initiated by an active feedback loop, it will be limited by the delay. As the movement becomes faster the delay in reporting the actual velocity becomes more significant. It is known however that humans are not limited by this delay in performing fast movements. It seems the only way that this can be possible, is by learning. A combination of McIntyre's and Sanger's ideas might be able to accomplish this. In this combined model the velocity feedback signal would not only go to the reflex

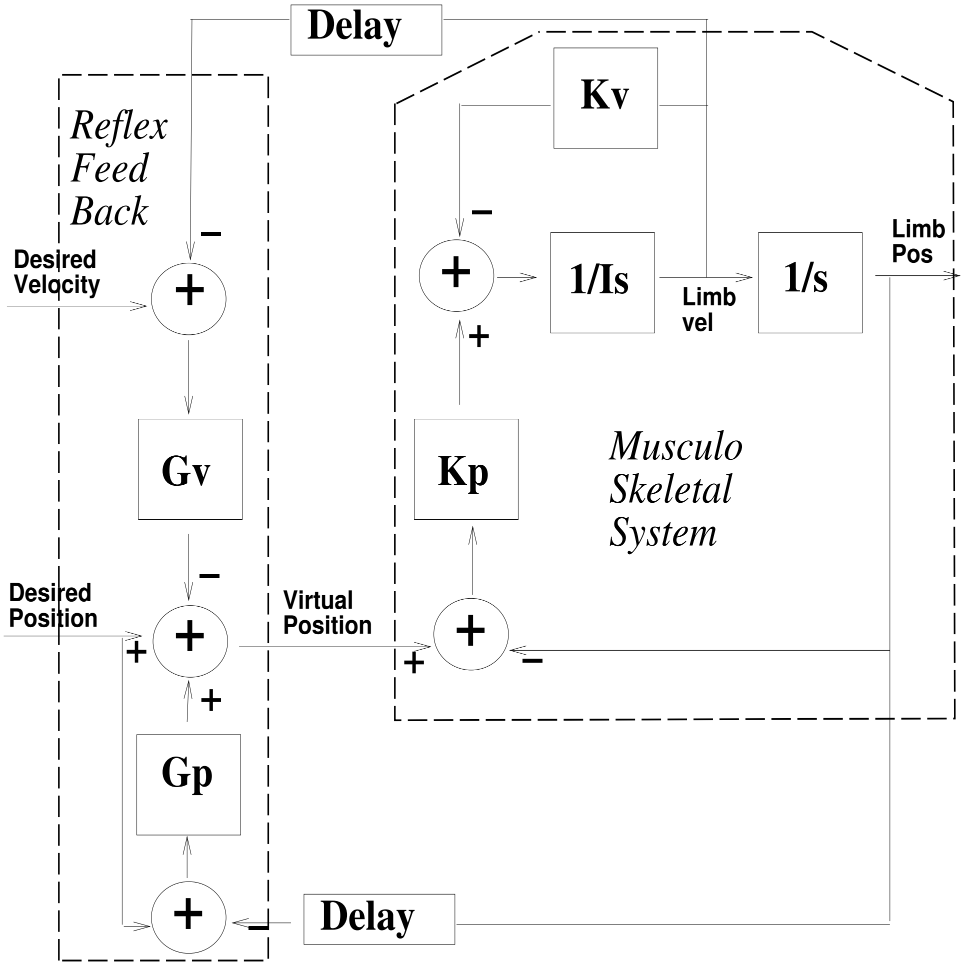


Figure 6: The augmented alpha-lambda model for fast movements, from McIntyre's Thesis.

feedback mechanism but also to higher centers that would remember the correlation between the virtual trajectory just specified and the velocity signal.

Experimental evidence for the Equilibrium point hypothesis:
Three experiments are cited in the Bizzi paper as support for the Equilibrium point model, all of them deal with the single joint case. As my goal is to contrast the inverse dynamics approach with the equilibrium point approach, I looked for results that pointed to one or the other. Here are the summaries of the experiments and their findings. In my view the third experiment is the best of the three, it's results strongly favor the equilibrium hypothesis and are damaging to the biological validity of inverse dynamics.

Experiment 1:
In the first experiment a monkey was trained to move its forearm to different positions on cue. After cueing the monkey to move to some new position the arm of the monkey was briefly restrained and the acceleration was measured at the time of release. The experiment was performed on normal and deafferented animals. It was found that as the holding time increased initial acceleration increased and then reached a constant value. This result is interpreted in the following manner: The controller specified some virtual trajectory from

initial to final position. As the difference between the desired arm position (on the trajectory) and the actual arm position (being restrained at the initial position) grows the restoring forces increases because it is proportional to the difference between the initial and desired values. Finally the virtual equilibrium point reaches the final desired position after which the torque plateaus because the difference between actual and virtual position does not change any more. What kind of accelerations should we expect to see if the controller were using inverse dynamics to compute a torque profile? We would see a rising acceleration as shown by the data, but we would expect it to decrease and go to zero at the destination. However, the authors claim that the acceleration reached a steady state after some time, if this is true then it is the key piece of evidence is this experiment for the Equilibrium hypothesis and against the inverse dynamics model.

Experiment 2:
In the second experiment the monkey again had to move its arm to some location. This time however it was assisted by an external torque to a position beyond the desired position, and then released. It was observed that the arm first returned to an intermediate position between initial and desired positions and then went back to the final position. This experiment was performed on intact and deafferented monkeys. The explanation of the results in terms of the Equilibrium Point hypothesis is as follows: The assistance to the arm causes it to go ahead of the equilibrium point on the virtual trajectory, consequently when the arm is released it feels a restoring torque back towards the equilibrium point which is currently somewhere inbetween the initial and desired positions. Which is where it goes before resuming its virtual trajectory to the desired position. While this explanation agrees with the results, it is does not preclude the possibility of the inverse dynamics control with overshoot.

Experiment 3:
Again the monkeys were trained to move their arm to different target locations. The monkey was deafferented and a cover prevented the animal from seeing its arm. The monkey's arm was passively moved from some initial position "A" to some new location "B" - without its knowledge. Because it has no sensory feedback, the monkey still thinks that its arm is at "A". Now the monkey was asked to move its hand to target location "B" (even though it happens to be there already). The arm was held at "B" for some time before being released. It was observed (as in the previous experiment) that the monkey's arm first moved to some intermediate point between "A" and "B" before returning to "B". Furthermore, the longer the arm was restrained at "B" the closer the intermediate point was to "B". Of the three experiments I believe that this one is the most favorable to the Equilibrium Point hypothesis and most damaging to the inverse dynamics approach for the following reason: Suppose the monkey was using inverse dynamics, It would find some trajectory profile from "A" to "B" and then transform into a torque profile via inverse dynamics. If this torque profile were applied when the monkeys arm was already at "B" then it would take it in a direction *opposite* to the direction in which "A" lies, however this is not what is observed - the arm moves toward "A" and then back to "B". This has a clear explanation in terms of the equilibrium point approach - the monkey had constructed a virtual equilibrium trajectory, from "A" to "B". When the hand was released it moved towards the current position of the virtual equilibrium point which happened to be between "A" and "B". So even though the hand was already at "B" it moved towards "A".

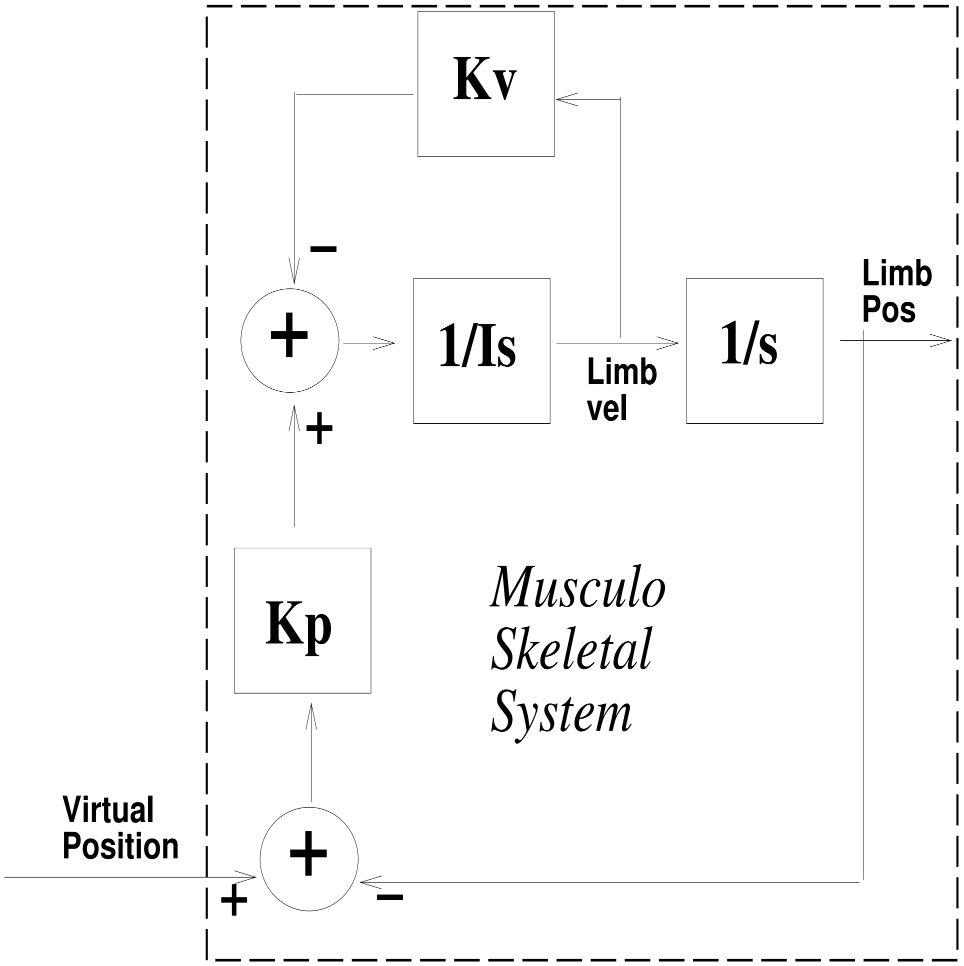


Figure 7: The Alpha model, from McIntyre's Thesis.

The virtual trajectory brings it back to "B". As the holding time of the hand was increased , the virtual equilibrium point moves closer and closer to "B". The equilibrium point hypothesis both qualitatively and quantitatively agrees with the results. The inverse dynamics predictions are qualitatively wrong here.

Role of Active Feedback:
There are two variants of equilibrium point control: One called the Lambda model originally proposed by Feldman in 1966, and another called the Alpha Model, is Feldman's characterization of the work by Bizzi and colleagues.
The Alpha model, the one described so far is shown in figure 7.
The Musculo-Skeletal system in figure 7 simply refers to the limb and the muscle *for a fixed virtual position specification.* The virtual position signal is a signal to the muscle from the controller. The signal modifies the spring constant K_p *and* the rest length of the muscle. In combination with the corresponding signal to the antagonist muscle, the signal effectively determines the virtual or desired equilibrium position. Notice that there is no delay in the generation of the restoring torque.
The lambda model has evolved from the time of its initial proposal. The simplest version of the Lambda model is identical to Merton's servo loop hypothesis, where the equilibrium

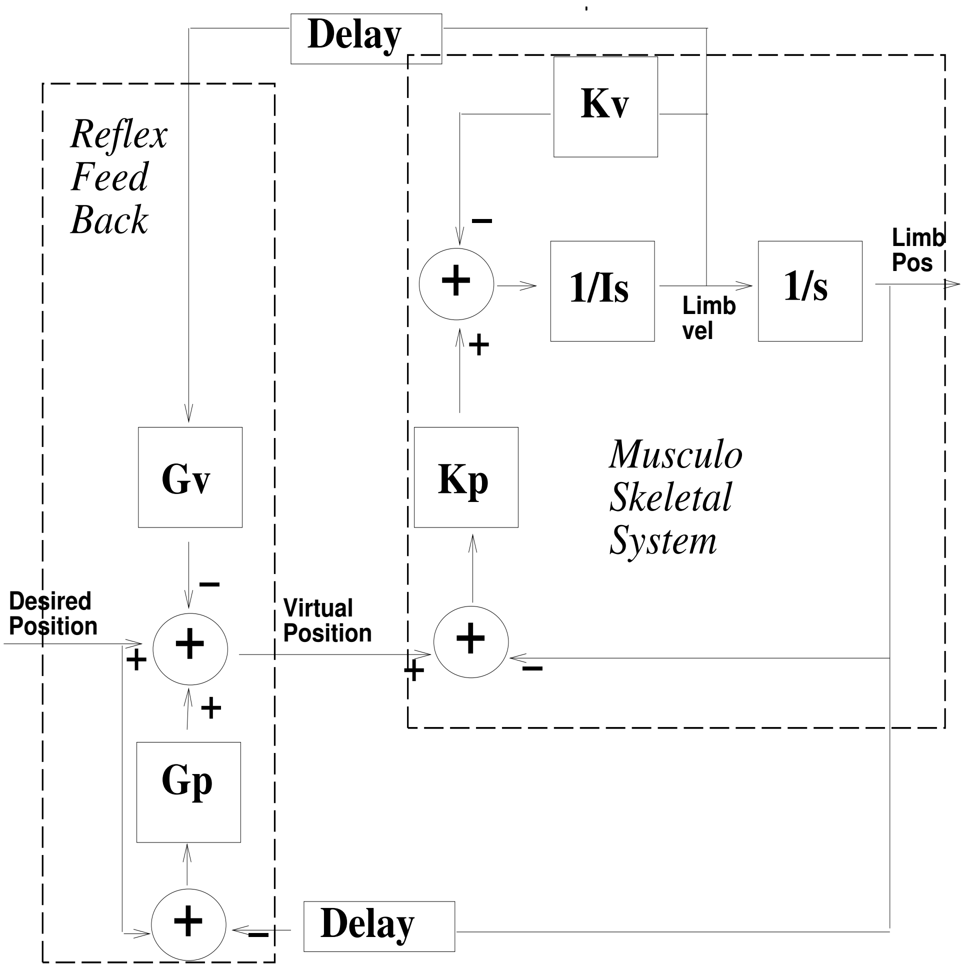


Figure 8: The Lambda model, from McIntyre's Thesis.

point is set by the active reflex feedback loop. The more recent version of the Lambda model is shown in figure 8. Notice that it subsumes the alpha model. The alpha model is augmented with a reflex feedback loop that sets the virtual position of the Alpha module. Both of these models agree on the basic equilibrium point control hypothesis; namely, that the dynamics of the actuators result in equilibrium points, and that movements can be generated by a moving virtual equilibrium point. Neither of them denies the existence of the musculo-skeletal feedback loop or the reflex feedback loop, they simply disagree on the relative importance of the musculo-skeletal system and the reflex feedback loop in achieving an equilibrium position. They are more similar than dissimilar.

Contact Tasks

By gradually moving the virtual position *into* an object, the equilibrium control hypothesis elegantly makes a transition from position to force control. The force exerted by the hand on the object can be varied by controlling the depth of the equilibrium point in the object. What role do the dynamics of the actuator play in this interaction? Hogan [10] shows that if an actuator is under active control and is coupled to an external passive object of mass m_e. Then there exists an entire range of values of m_e for which the coupled system is unstable. However, if the actuator behaves like a passive system (this will be true when the virtual position input is held fixed), then the coupled system is guaranteed to be stable.

4 Final comparison

The recurring theme in this paper has been that the dynamics of the springlike actuators are important, and can be exploited to simplify the computations in the more conventional approaches to motor control. In particular, movement tasks (slow and fast), Load compensation tasks, and Contact tasks were considered.

Movement Tasks:
It is clear that the Equilibrium point hypothesis avoids the inverse dynamic computations by exploiting the dynamics of the actuators, and the natural tendency of the passive system to settle into an equilibrium position. The question then is whether it is any easier to calculate the spring stiffnesses that achieve some equilibrium position compared to the explicit calculation of the torques through inverse dynamics. It appears that for slow movements the answer is yes. The reason is that by choosing large stiffnesses one can reduce the effect of gravitational forces and get simple expressions relating the stiffnesses to the equilibrium position. Figure 4 shows the relation for the simplest case, but it is also true for the two-link manipulator. What about fast movements? We looked at three models for fast movements. Two of them - increasing the stiffness, and computing a complicated virtual trajectory were not regarded as viable alternatives. Instead a third alternative - a combination of McIntyre's fast movement model, and Terence Sanger's trajectory perturbation learning method seems hopeful for generating fast movements. What if the torques in the inverse dynamics approach are not explicitly calculated but are learned using the memory based approach, is it any easier to learn spring stiffnesses? The answer is again yes, because for every position we need to learn only one set of spring constants *no matter where the arm is* whereas different torques have to be learned depending on the initial state of the arm.
The role of feedback: The equation that describes the restoring force produced by the springlike actuators looks identical to Positional Derivative control. So it seems like a conventional system (without springlike actuators) with a Position Derivative feedback law should be equivalent to the Equilibrium point approach. However, this is not true. The critical feature of Equilibrium point control is that the Position Derivative control is *implicit* in the dynamics of the actuator, it is not *active feedback* as it would be in the conventional approach. Because it is implicit in the dynamics, it has zero delay, consequently the stiffness of the muscle can be increased *without fear of oscillation due to delay.* Conventional feedback is unstable when the gain and delay are large. Crude estimates of neural transmission delays (30ms for spinal cord feedback loops) show that the active feedback becomes limited when the frequency of the disturbance is above 2Hz, however the almost instantaneous response of the muscle dynamics enables us to responds to events of upto 15Hz.
What role does active feedback play in Equilibrium point control? Though the arm is guaranteed to eventually reach its equilibrium position in spite of the inertial, centripetal, gravitational, and Coriolis forces, these forces may cause large perturbations during movement. Thus the actual trajectory cannot be expected to match the virtual trajectory for this reason (in the conventional system we had an inaccurate dynamic model as a cause of this error). Another reason for the error (also with the conventional approach) is an

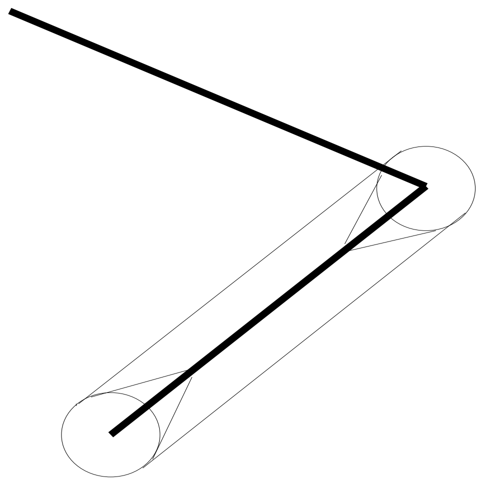


Figure 9: The two joint arm with six springs. The redundant cross coupling springs provide load compensation.

unpredictable environment. For these reasons Equilibrium point control has an active feedback mechanism that complements the ability of the feedforward mechanism to independently achieve the desired position. However as we just observed its loop delay limits its usefulness above some frequency. In the light of this problem, It has been proposed that the primary purpose of feedback is to use the state of the arm and the error to correct for *future* errors, not the one happening now. The role of active feedback in equilibrium point control is as unknown as it is in conventional approaches.

Load Compensation
We saw in the case of the two-link four spring arm that the joint stiffnesses caused an endpoint stiffness. The force field corresponding to this endpoint stiffness was springlike. This implies that perturbations of the endpoint position by a load would result in a restoring force that would try to counteract the load. Thus the dynamic properties of the actuators are providing a first line of defense, while higher level mechanisms respond to the perturbation. In fact the restoring force due to the spring dynamics can be further increased by a poly-articular muscular system - i.e a coupling spring between joints as in the six spring system in figure 9.
Putting a load on the tip would cause the cross-coupling springs to provide a restoring force in the vertical direction, countering the load.
Contact tasks:
How does Equilibrium point control compare with the conventional approach on tasks that require handling of objects? for example picking up and drinking from the cup. The following result due to Colgate and Hogan [2] is the key reason why Equilibrium point control is better to suited to contact tasks than conventional approaches. "The necessary and sufficient condition for a system such as a robot or a human limb to avoid instability when coupled to any stable, passive object is simple: Its driving point impedance must appear to be that of a passive system." The conventional robot actuator can supply energy indefinitely, whereas the closed loop transfer function of the springlike actuator is that of a passive system. Therefore equilibrium point control does not lead to contact instability - a pathological problem with conventional robot arms.

The equilibrium point control hypothesis as described by its proponents still does not explain how in fact the spring constants and rest lengths are modulated with time given a particular trajectory, nor is there a very satisfactory explanation for producing fast movements. However, What it does do is suggest a way of exploiting actuator dynamics to provide a crude “peripheral” solution before more complicated control mechanisms take over.

References

[1] Bizzi and Mussa-Ivalsi. Muscle properties and the control of movement. *Visual Cognition and Action*, 2(pp213-239), 1990.

[2] J.E. Colgate and N. Hogan. Robust control of manipulator interactive behavior. *ASME: Modelling and Control of Robotic Manipulators*, (pp149-159), 1987.

[3] Bizzi Hogan Mussa-Ivaldi Giszter. Does the nervous system use equilibirum-point control to guide single and multiple joint movements? *Brain and Behavioral sciences*, 15(pp603-613), 1992.

[4] Neville Hogan. Planning and execution of multijoint movements. *Canadian J. Physio. Pharmacol*, 66(pp508-517), 1988.

[5] John M. Hollerbach. *Visual Cognition and Action*, volume 2, chapter Fundamentals of Motor Behavior, pages pp153–182. MIT Press, 1990.

[6] John M. Hollerbach. *Visual Cognition and Action*, volume 2, chapter Planning of arm Movements, pages pp183–211. MIT Press, 1990.

[7] Richard Lathrop. Paralleism in manipulator dynamics. A.I Memo 754, M.I.T, 1983.

[8] Joseph McIntyre. *Utilizing Elastic System Properties for the Control of Posture and Movement*. PhD thesis, M.I.T, 1990. Department of Brain and Cognitive Sciences.

[9] Suzuki Uno, Kawato. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cyberetics*, 61(pp89-101), 1989.

[10] J.M Winters S.L-Y. Woo, editor. *Mechanical Impedance of Single and Multi-Articular Systems*, chapter 9. Springer-Verlag, first edition, 1990.

[11] J.M Winters S.L-Y. Woo. *Multiple Muscle Systems*. Springer-Verlag, first edition, 1990.