A SIMPLE AND ROBUST HIERARCHICAL CONTROL SYSTEM FOR A WALKING ROBOT

J R Kennaway

University of East Anglia, U.K.

INTRODUCTION

The present work applies a control architecture proposed by W.T. Powers [3, 4], to several problems in robotics, and suggests that it may have wide practical applicability. The architecture is called (Hierarchical) Perceptual Control Theory, or HPCT, and was proposed by Powers as a possible organisation for living control systems.

EXAMPLE: THE INVERTED PENDULUM

We shall introduce this approach by way of a simple example, designed by Powers. Consider the inverted pendulum shown in Figure 1. We assume that the cart travels on

Figure 1: An inverted pendulum

a frictionless track, the rigid pendulum rod swivels freely at its base, and that there is an actuator which applies any specified horizontal force to the cart. To move the pendulum bob to a specified horizontal position by means of this actuator is a complicated task. Nevertheless, it can be achieved by breaking the matter down into simpler tasks, as follows.

If we had an actuator that could set the bob immediately to any desired position, no control system would be necessary. We don't have such an actuator; but if we had one which could set the pendulum's horizontal velocity, we could use this to control the position: set the velocity equal to $k_0(r_b - b)$ for some constant k_0, where r_b is the demanded position and b is the current position. We don't have such a velocity actuator, but if we had an actuator that set the bob's acceleration, we could control the velocity $\dot{b}$ to approach a reference value $r_{\dot{b}}$ by applying an acceleration $k_1(r_{\dot{b}} - \dot{b})$. The acceleration is proportional to the pendulum angle, which is proportional to $o = b - c$, where c is the position of the cart. So we can set the acceleration by setting o. We cannot set o directly, but we could control o if we could set the cart's velocity, by setting $\dot{c} = k_2(r_c - c)$, where r_c is the reference cart position. We cannot set $\dot{c}$ directly, but we could control it if we could set the cart's acceleration: $\ddot{c} = k_3(r_{\dot{c}} - \dot{c})$, where $r_{\dot{c}}$ is the demanded cart velocity. Finally, we can set the cart's acceleration by applying a force to the cart, which by hypothesis we are able to do.

The resulting arrangement of four proportional controllers is shown in Figure 2. For suitably chosen values of the gain parameters, it is found to work very stably and robustly (although it is not able to swing the pendulum up from the straight down position). Although the construction has been described above on the assumption of linearity, which fails when the pendulum angle departs too far from the vertical, the non-linearities are controlled against in the same way as external disturbances. The physical simulation (from which Figure 1 is a screen shot) uses the true differential equations, valid for all pendulum angles.

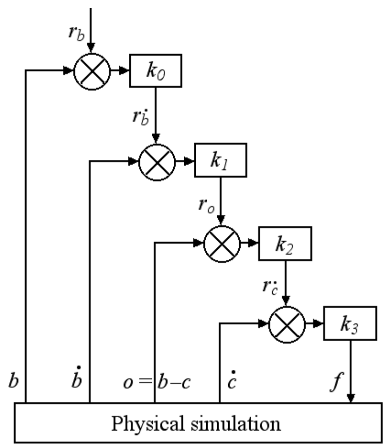

Figure 2: Control hierarchy for inverted pendulum

ANALYSIS OF A 2-LEVEL CONTROLLER

As an example of the mathematical analysis of such a system, we illustrate a two-level controller, consisting of the bottom two levels of the pendulum controller. The position x is controlled by setting a reference for $\dot{x}$, which is controlled by setting a force, which determines $\ddot{x}$ by Newton's second law. Letting r_x be the reference position and $r_{\dot{x}}$ the reference velocity, the equations are:

$$r_{\dot{x}} = k(r_x - x) \qquad m\ddot{x} = k'(r_{\dot{x}} - \dot{x})$$

and therefore

$$m\ddot{x} + k'\dot{x} + kk'x = kk'r_x$$

This is identical to the equation of damped harmonic motion, although there are no physical springs involved. If we write $\rho = k'/mk$ (the ratio of the time constant $1/k$ of the upper controller to m/k', that of the lower controller), then the roots of the characteristic equation are $\frac{1}{2}k(-\rho \pm \sqrt{(\rho-2)^2 - 4})$. For large ρ, these tend to $-k$ and $-k(\rho - 1)$. For $\rho = 4$, the roots coincide at $-2k$, and as ρ approaches zero, they describe circular arcs in the complex plane as in Figure 3. The fastest response is

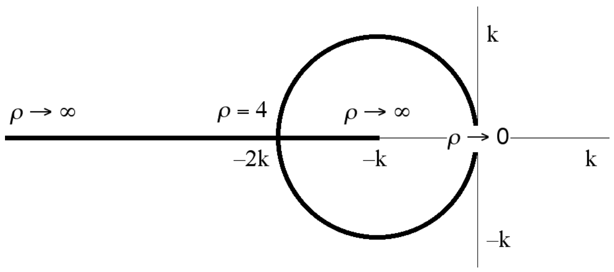

Figure 3: Root locus of two-level control system

obtained for ρ equal to 4 or a little more (depending on the precise definition of response time).

Cascade control

The above control scheme closely resembles a standard configuration in process control known as cascade control, although the motivation is somewhat different. In cascade control, where a single controller produces unacceptable performance, due to the chosen actuator having a slow effect on the controlled variable, a second controller is introduced which controls some variable (called the secondary variable) which has a more rapid effect on the primary controlled variable. The output of the primary controller connects to the secondary reference (setpoint) input, and the secondary output connects to the actuator. Cascades of three or more controllers are possible, but typical practice employs just two controllers.

We find it more instructive to consider each controller as providing a *virtual actuator* to the next controller up, as suggested by our description of the inverted pendulum controller. In addition, the hierarchical arrangement is capable of much wider application, which we will demonstrate with the walking robot which forms the main example of this paper.

The value of 4 that we found above for the ratio of upper to lower level time constant agrees with a standard rule of thumb for cascade design, that the secondary controller should have a response 4 or 5 times as fast as the primary.

A FOUR-LEGGED WALKING ROBOT

We have constructed a physical simulation of a walking robot (see Figure 4) with four or more legs in which there are two levels of the hierarchy, six controllers at the upper level, and 12 controllers at the lower level. All of these controllers are of the PID type. At the upper level there is one controller for each degree of freedom of the robot's body. We assume that the robot can perceive the height of its body above the ground; its other five degrees of freedom are assumed to be defined relative to the positions of its feet on the ground (calculated relative to the body from the joint angles by forward kinematic calculations). At the lower level, there is a controller for each of the three degrees of freedom of each leg: one at the knee and two (pitch and yaw) at the shoulder. Each of the lower-level controllers controls the rate of change of joint angle; its output is the torque applied to the joint.

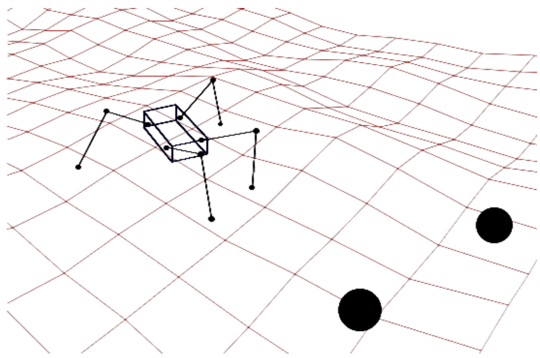

Figure 4: Four-legged robot. The blobs are virtual food particles.

Each top level controller's output is directed to some subset of the lower level controllers' reference inputs. Thus the reference of each lower level controller is a weighted sum of the outputs of the upper level controllers. The weightings are given by a 6 by 12 linkage matrix, in which all of the elements are 1, 0, or -1. (The behaviour of the robot is experimentally found to be insensitive to the exact values.) For this example, it is sufficient to choose the weights by straightforward physical intuition. For the robot to lift its body higher, it must decrease the pitch angle at each shoulder. To swing its body to the left, it must swing each shoulder joint to the right. To sway the body towards the right, it must decrease the knee angles on the right, and increase those on the left. And similarly for the other three degrees of freedom: pitch, roll, and forward sway.

With this control system, the robot is able to stand up and balance on uneven terrain, and resist random external forces. The architecture works equally well for six, eight, or more legs (the linkage matrix having a general definition that is uniform in the number of pairs of legs), and the robot can continue to stand and resist disturbing forces even when a leg is removed. The values of the linkage matrix are not critical. One can even replace a few of them by random values and obtain a system that controls almost as well. Provided that there are sufficient degrees of freedom at the lower level, and that different body controllers do not both try to use the same set of signals to the lower level in order to control different perceptions, it is possible for all of the top level controllers to simultaneously achieve good control, despite their interactions.

It is important to the functioning of the robot that the body controllers do not try to set the joint angles, but only their velocities. Computing the angles required to produce a given posture of the body requires complicated inverse kinematic calculations and sensing of the terrain, and errors in the data on which these calculations depend would result in errors of comparable size in the body parameters. The control scheme described avoids this problem by having the body controllers demand certain rates of change of the joint angles. The negative feedback action of the controllers ensures that the joint angles will arrive at whatever values are required, with residual errors depending only on the tuning of the controllers and the accuracy with which the controled variables are measured.

As an indication of the simplicity of the control scheme, in a physical construction (which would dispense with the physics simulation code), the most complicated calculations would be the forward kinematic computation of the body position relative to the feet. There is no motion planning, inverse kinematics, learning, adaptation, or modelling by the robot of its environment. The robot is described in more details in [2].

Analysis of a simplified robot

To mathematically illustrate, in a simpler setting, the operation of a hierarchy such as we have described, we will consider a greatly simplified robot with two legs and one degree of freedom in each leg (Figure 5). Each leg is a

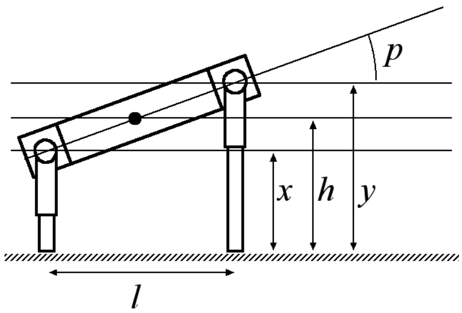

Figure 5: Two-legged two d.o.f. robot

linear actuator which has a length x or y, and exerts a vertical force f_x or f_y. The body has two degrees of freedom of movement: height (h) and pitch (p). We assume the centre of the body is constrained to a vertical line, and that the pitch remains small, so that we can approximate the kinematics and dynamics by linear equations:

$$h = (x+y)/2 \qquad p = (y-x)/l$$
$$\ddot{h} = (f_x+f_y)/m \qquad \ddot{p} = (f_y-f_x)l/I$$

To simplify things, we choose units so that the mass m of the body, its length l, and its moment of inertia I are all 1. There are four proportional controllers, arranged according to the network of Figure 6. The lower level controllers control $\dot{x}$ and $\dot{y}$, and the upper level controls h and p. For simplicity we will take the reference inputs r_h and r_p to be zero (measuring h, x, and y relative to some convenient point above the ground). The resulting

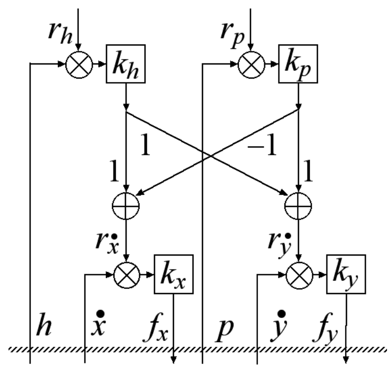

Figure 6: Control hierarchy

equations are:

$$r_{\dot{x}} = -k_h h + k_p p \qquad r_{\dot{y}} = -k_h h - k_p p$$
$$f_x = k_x(r_{\dot{x}} - \dot{x}) \qquad f_y = k_y(r_{\dot{y}} - \dot{y})$$

By symmetry it is reasonable to choose $k_h = k_p$ and $k_x = k_y$. By rescaling of variables we can choose $k_h = k_p = 1$. Writing k for k_x and k_y, the resulting equations for h and p are:

$$\ddot{h} + 2k\dot{h} + 2kh = 0 \qquad \ddot{p} + 2k\dot{p} + 2kp = 0$$

which again is the equation for damped harmonic motion for each variable. The optimal value for k is 2; this is equivalent to the value of 4 found for the two-controller hierarchy, taking into account that the linkage matrix maps each top-level output to both the bottom-level references.

It is instructive to consider what happens if we change the linkage matrix. If we replace the equation for $r_{\dot{x}}$ by

$$r_{\dot{x}} = -k_h \alpha h + k_p p$$

(that is, replacing 1 by α in Figure 6 on the line from the h controller to the $\dot{x}$ controller) then the equations for h and p become:

$$\ddot{h} + 2k\dot{h} + k(\alpha+1)h = 0$$
$$\ddot{p} + 2k\dot{p} + 2kp = k(\alpha-1)h$$

When $\alpha = 1$, this is the original system. For $\alpha = 0$, the height and pitch control interact, but both still reach their reference value. As α approaches -1, the response time of the height controller becomes longer and longer. When $\alpha = -1$, the linkage matrix is singular, which means that the height and pitch controllers are attempting to control different variables by means of identical actions. The result is that disturbances to the height of the robot are not controlled. For $\alpha < -1$, the system is unstable.

Walking and navigation

To make the robot walk, all that is required is for it to repeatedly lift up some subset of its legs, swing them forwards, and put them down again. The controller for forwards position relative to the footprint will then pull the body forwards. On uneven terrain, the body pitch and roll controllers will keep the body aligned with the plane of the footprint. Similarly, to turn anticlockwise, it repeatedly lifts a subset of legs, swings them anticlockwise, and puts them down, letting the body heading controller bring the body into alignment with the new footprint.

By these means, the robot is able to walk and turn on uneven terrain, and go up and down (shallow) stairs. Adding rudimentary senses to detect the direction of a landmark enables it to navigate towards it by varying the magnitude of the walk and turn actions. The robot has been implemented in a simulation that can be run from http://www.cmp.uea.ac.uk/∼jrk/Archy/Archy.html.

The simulation includes an implementation of the dynamics of a rigid body (the robot body excluding the legs), acted on by forces exerted by the legs between the body and the ground. The dynamics of legs lifted from the ground has not been modelled; nevertheless, the control problem, although simplified from reality, is a complex control problem in its own right, which the control architecture we have described is empirically able to solve.

TWO MORE CASE STUDIES

A backhoe excavator

A backhoe excavator, such as that of Figure 7, has three joints, each actuated by a hydraulic cylinder. If one wishes to drive the bucket in a straight horizontal line, keeping it in a constant orientation, one must operate all three actuators in a rather complex way. A control system whose controlled variables are the reach, lift, and inclination of the bucket can provide the operator with the ability to directly drive the bucket straight forwards, backwards, up, and down. It is straightforward to apply the architecture described above to the task, and we have done so using a simulation based on the Vortex physical simulation library[1]. Some screen-captured movies of the resulting simulations are available at http://www.cmp.uea.ac.uk/∼jrk/Robotics/digger.

The control architecture is essentially the same as that of the 2-degree of freedom robot, but with three controllers on each level. On the top level there is one to control each degree of freedom of the bucket, and on the bottom level there is one to control the velocity of each joint. The actuators are assumed to supply a specified torque to the joint. (We have not modelled actual hydraulic actuators.)

The linkage matrix connecting the top-level outputs to the bottom-level references is more complicated than for the robot, as it depends on the current state of the robot. When the bucket is far, increasing the dipper angle will raise it, but when it is close, increasing the dipper angle lowers it (Figure 8). When the dipper is vertical, it has no effect on bucket height. The routing of the output from the height controller to the reference input of the dipper controller must therefore depend on the current configuration. For each top-level variable x, and each joint y, we have chosen the corresponding element of the linkage matrix to be $\partial x/\partial y$. That is, the more effect an actuator has on a top-level controlled variable, the more that actuator will be used to control it. As with the 4-legged robot, the precise values are not critical. In the resulting simulation, the bucket closely follows the reference point and reference inclination as the references are moved by the operator.

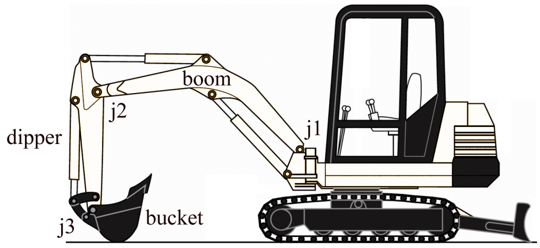

Figure 7: Backhoe excavator (JCB model 803)

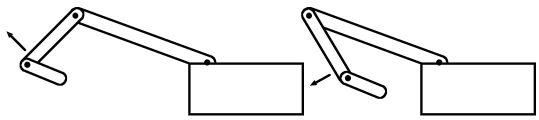

Figure 8: Dependence of linkage matrix on configuration

An obvious generalisation of this design is to a backhoe with more joints in its arm. With four joints, as in Figure 9, there is an extra degree of freedom which is not constrained by the position and orientation of the bucket. A simple way of fixing the extra degree of freedom is to add another top-level controller whose controlled variable is $a_3 - a_2$ (the difference between the exterior angles at the second and third joints), or more generally, $a_3 - ka_2$ for some constant k. This forces joints 2 and 3 to have similar exterior angles, avoiding either of the extreme configurations shown in the Figure. While a four-jointed backhoe may not necessarily be a more practical excavation machine than the standard three-jointed configuration, there are applications to robotic arms requiring extra joints to reach into confined spaces. Arms with five or more joints can be controlled similarly, using extra equalisation controllers to fix all the extra degrees of freedom. Such arms may be beyond the capacity of a human controller to operate effectively by directly driving the joints.

[1]Vortex is produced by CM Labs, http://www.cm-labs.com.

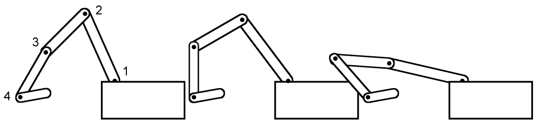

Figure 9: Four-joint backhoe

A two-dimensional biped robot.

The physical simulation of our four-legged robot deliberately simplifies reality in order to alleviate the programming task. Using the Vortex library, we have begun to develop more faithful simulations with the goal of simulating the robot with sufficient physical fidelity to justify actual construction. Initially, we have constructed a two-legged robot constrained to a vertical plane, capable of standing and controlling the three degrees of freedom of its body: height, pitch, and sway (Figure 10). As with the backhoe, legs with more joints than necessary can be used, with the extra degrees of freedom taken up by controllers keeping consecutive joint angles equal. There is one further degree of freedom unaccounted for: this is the force between the two feet tending to splay them apart or draw them together. An extra controller can be added to maintain the splay force near zero, by adding suitable amounts to the rate of change of the torques at the leg joints.

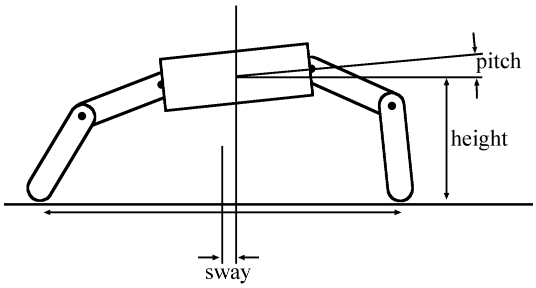

Figure 10: Two-legged two-dimensional robot

FUTURE WORK ON WALKING

The four-legged robot simulation described above omitted both the physics and the control problem of controlling a leg whose foot is off the ground. A controller for foot position can be designed along similar lines to the backhoe control system (assuming the other legs are meanwhile providing sufficient support for the body). This will require switching each leg between being used either to control foot position or to control body position. A third-level walking controller would accomplish this by altering the linkage matrix connecting the foot and body controllers to the joint angle velocity controllers.

RELATED APPROACHES

Besides cascade control, described above, there is another approach to the design of robotic systems which bears a superficial resemblance to the present proposal, called *subsumption*. This is an architecture originally devised by Brooks [1], in which the control problem is, as for HPCT, broken down into a hierarchical arrangement of simpler agents. There are two fundamental differences with HPCT. Firstly, in a subsumption architecture, the agents are not necessarily conceived of as controllers, that is, agents which attempt to produce a certain input by means of their outputs. Secondly, the main principle of the subsumption architecture, for which it is named, is that all of the controllers at all levels act directly on the actuators, controllers at higher levels suspending the actions of controllers at lower levels as necessary, the lower level resuming its operation when the higher level has completed its task. Thus some of the actions taken to balance a legged robot in a standing posture are suspended when a higher-level agent for walking needs to lift some legs off the ground; an agent for walking in a straight line will be suspended by an agent for collision avoidance, and so on. In HPCT, only the bottom-level controllers send signals to the actuators. Higher level controllers send their outputs only to the reference inputs of controllers at the next level down. In subsumption, higher-level agents operate *instead of* lower level agents; in HPCT, higher-level controllers operate *by means of* lower level controllers.

In principle, an HPCT controller could act not only by altering the references of lower level controllers, but also, for example, by altering parameters of their output functions, or the linkage matrix connecting them to their descendants. However, there is never any skipping of levels. As mentioned above, we intend to employ the latter scheme of modifying the linkage matrix to implement walking, to switch each leg between the roles of supporting the body and moving the foot to a new position.

References

[1] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[2] J. R. Kennaway. Control of a multi-legged robot based on hierarchical perceptual control theory. *J. Perceptual Control Theory*, 1(1), 1999.

[3] W. T. Powers. *Behavior: The Control Of Perception*. Benchmark Publications, 1998. Originally published by Wildwood Press, 1974.

[4] W. T. Powers. *Making Sense of Behavior: The Meaning of Control*. Benchmark Publications, 1999.