Arm One *C-language*
Little Man One *DOS*
Description

THE CONTROL SYSTEMS IN ARM VERSION 1, ORIGINAL DOS VERSION

This program simulates a human arm reaching out to touch a target the user can move in three dimensions. The arm has three degrees of freedom (two at the shoulder and one at the elbow). The position of the "fingertip" is ray-traced to form two retinal images in which both the target and fingertip positions appear. These images are used to derive x, y, and distance signals, which are controlled by a visual system that varies the reference signals entering the three kinesthetic higher-order control systems.

Note: Make sure the caps lock and num lock indicators are off while the program is running

INTRODUCTION

This program demonstrates a model of pointing. You can move a target around in three dimensions while a model person reaches out to touch it, following it as it moves. There are five lower-order control systems and three higher-order control systems that run the model; you can alter the basic parameters of all systems to see the effects. The model person uses binocular vision in three dimensions to detect depth information; all visual information is computed from reasonably accurate optical computations that calculate the finger and target angles that each eye will see. The gaze angle tracks the target (the head moves) while the "finger" at the end of the arm does the same. You can alter the direction in space from which you view the model, which is shown in three dimensions and perspective. Graphical tracings of the three higher-order perceptual signals are shown.

The program allows operation using a mouse or the keyboard. A joystick connected to game port channels 1 and 2 can also be used.

When you uncompress the program, be sure the THREE auxiliary files are included and in the same directory: SINCOS, INVTAN, and DIAGRAM. If these are missing, the program will create all but DIAGRAM—that one's ESSENTIAL. After the first time, the files will simply be loaded when the program starts.

Details of operation

ARM1.exe is the executable file. The first time you will be asked for configuration information; a file is created. From then on the program will simply start without any questions. To change configurations, delete the file called "ADCONFIG" and start the program again. New configuration information will then be requested.

When the program is running, always use lowercase letters. Make sure the Caps Lock and Num Lock indicators are off (press those keys if the letter commands or arrow keys don't work).

With the game port (g) A-to-D (a), or mouse you have two control actions to move the target in three dimensions. By typing "x" you can make one axis of the controller affect either the x or the y direction of target movement, toggling back and forth. The other controller motion always controls the depth direction of the target movement.

With keyboard (k) operation, the arrow keys move the target. The up and down arrows move it up and down, the left and right keys move it left and right (from the little man's point of view). The PgUp and PgDn keys move it in depth, away from and toward the model's eyes. The "+" key increases the size of the movement steps for all three axes; the "-" key decreases it. You can use the +/- keys near the number pad. Screen instructions remind you of the effects of all these keys.

Program operation

The program comes up running. Type 'q' to leave it. Instructions are at the bottom of the screen. Typing u, d, r, l, moves the viewpoint in the indicated direction (up, down, right, left), moving YOU around the display. To pause the display, type "p." Any key restarts the action. The pause is useful for printing screens if you have loaded a graphics print- screen program.

To set the parameters, type "s". A new screen appears. With the up/down/right/left/enter keys you can move the highlight to the various items. Just start typing to replace the existing item. See PARAMETERS below for details of how to set the parameters. The End key exits to the graphics display.

To show an animated block diagram, type "b". The meanings of the variables are given in ARMCALC.DOC. Another "b" turns it off.

Toggle volitional action on/off with the !.

This model is much simpler than a real human system. As a result, it can get into trouble that a real person doesn't experience. When the gain, slowing, and integration factors are set for slow and loose operation, rapid target movements can drive the calculations beyond the limits of the fast integer arithmetic used in the program, particularly with the fingertip close to the eyes. When the arm gets totally outside the range of the computations, it may lock up. When it does, move the target farther from the eyes and press the "z" key to reset the model.

Occasionally the program will halt on a "Divide by zero" error (error 200), and will apologize to you. Checking for this condition would have cut the operation speed by a factor of two to three. The cure, which is to use floating-point arithmetic, would have cut it far more than that. If you keep target movements slow when the fingertip is near the eyes, you will seldom see this effect. When you do see it, just restart the program.

When you enter numbers to set parameters, don't use any decimal points—WHOLE NUMBERS ONLY. Decimal points and non-numerical characters are simply ignored. The arrow and enter keys move you to a new item. When you're done, use the End key to get back to the action. If you make a mistake use the backspace key to start over.

Theory

The little man has one arm consisting of an upper arm and a forearm. The upper arm can swing left and right about a vertical axis through the shoulder, and up and down around a horizontal axis through the shoulder. The horizontal axis swings as the arm swings laterally. The forearm can swing up and down around the elbow joint, always in a vertical plane. The man's head can swing left/right and up/down, both movements pivoting on the neck. There are two eyes on the man's head, set forward of the plane of the head by a small amount. Figure 1 shows the variables involved in all movements.

This model has been constructed so that as far as possible everything it does is based on what it, not its designer, knows. It knows nothing of the physics or geometry of the environment, or of the linkages in its arm or the geometry of its head. The only information it uses consists of the "felt" angles at shoulder and elbow, and the "seen" positions of the fingertip and the target as projected onto on its retinas. Of course the designer had to provide the calculations of where the fingertip would be for the given joint angles, and how the visual images of fingertip and target would be affected by head angle and optical projections. Those environmental calculations show how perception will depend on action, and are completely independent of the organization of the behaving system. In the real system these "calculations" are done by the physics of the environment.

Lower-level control systems

The position of the (right) arm is controlled at the lower level of the model by three control systems that use "kinesthetic" joint angle information. The elevation and azimuth angles at the shoulder are sensed, and the elbow angle is sensed. It's assumed that the actual joint angles will come to the requested values through the action of still lower-order control systems that are not included in this version of the model. It isn't necessary that the angles be precisely or linearly controlled.

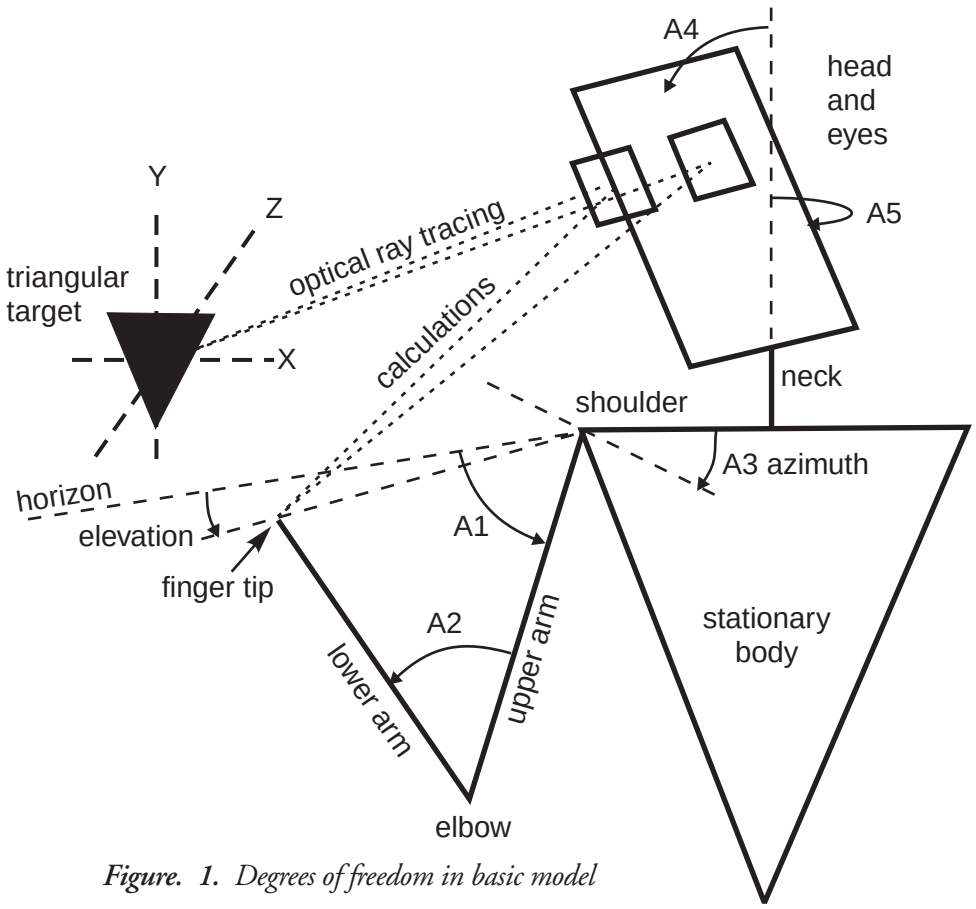

Figure. 1. *Degrees of freedom in basic model*

Angles:

A1: Vertical angle between upper arm and horizon. (Positive above horizon. Negative as shown).
A2: Inside elbow joint angle in vertical plane. Always positive. (Shoulder to fingertip angle in vertical plane. Not shown. Computed = A1 + 1\2(180-A2). (180-A2) = outside elbow angle).*
A3: Horizontal angle between arm and body. (Zero when arm is straight ahead. Positive to the right).
A4: Vertical rotation angle between head and body. (Zero when head upright, negative when head tilts down).
A5: Horizontal angle between head and body. (Zero when head straight ahead, positive when turned to the right).

Kinesthetic vertical angle control:

(angle A1 in Figure 1, negative as shown).

The first of the control systems perceives the angle from the shoulder to the fingertip. This angle needs not be defined by itself. Since the upper and lower arms are of equal length, the angle can be computed from the upper-arm angle and the elbow angle. Fingertip angle = A1 - 1/2 A2 + 90. This sensed angle is compared with a reference signal from a higher-order system. The error in this kinesthetic system is amplified and slowed; the gain and slowing factor are adjustable.

This system makes the perceived shoulder-to-finger angle match the given reference signal, no matter what the angle at the elbow.

Kinesthetic azimuth (horizontal) angle control:

(angle A3 in Figure 1, positive as shown)

The next system senses the azimuth angle at the shoulder and compares that perception with the reference signal received from the fingertip azimuth control system of higher order. The error signal is amplified by a gain factor and slowed by a slowing factor, both adjustable. The output determines the azimuth angle at the shoulder.

Kinesthetic finger distance control:

(angle A2 in Figure 1, positive as shown.)

The third arm control system at this level senses the elbow joint angle, compares it with a reference signal from the finger distance control system of higher order, and uses an amplified and slowed error to drive the elbow angle. The variable actually controlled has a nonlinear relationship to the true shoulder-to- finger distance. The true distance varies as the cosine of half of the quantity (180 degrees minus the external elbow angle), whereas the model controls only the angle.

Interlude—environmental calculations

The remaining control systems at both levels use ONLY information available to the eyes. The environmental calculations are by far the most complex of the model. The program first computes the position of the fingertip in X,Y,Z coordinates, as determined by the lengths of the arm segments and the angles A1,A2, and A3 generated by the kinesthetic control systems (see Figure 1). The target position is given by the analogue or keyboard inputs that you enter. The program then computes the direction from each eye to the target and to the finger as a horizontal and a vertical angle. These angles correspond to positions on the retinas. It is assumed that the model's brain can identify which object in the visual field is the target and which is the fingertip. It's assumed to sense relative angles as separations on the retina—between the lines of sight to the objects, and also between the center of vision in each eye and the image of the target. Note that the environmental calculations deal with the physics of the external situation and the optics of the eye—they are not part of the model of the behaving system, but describe its environment in terms of geometrical and optical relationships.

The environmental and control-system calculations are repeated each time a new display is generated. On a 10 MHz AT-type computer with monochrome graphics, the updates occur 25 times per second. On the same computer with CGA, they occur 30 times per second. On a vanilla PC there will be only 2 or 3 updates per second.

Visual-motor Head Azimuth (left-right) control:

(angle A5 in Figure 1, positive as shown)

The fourth lower-level control system uses optical information to position the head so the right eye's gaze angle is in a vertical plane that passes througbh the target. The environmental calculations yield the horizontal angle between the direction of gaze and the direction of the target. This perceived difference in angle is compared with a fixed reference signal of 0, implying that the system wants to look AT the target instead of to one side of it. The error is integrated to produce head angle in the horizontal direction. This is a pure integrating control system, for simplicity. The effect is to bring the head to a horizontal angle where the horizontal gaze direction is the same as the visual direction of the target. The integration factor is adjustable (the number adjusted is 1 divided by the integration factor, so a larger number means a slower integration).

Visual-motor Head Elevation (up-down) control:

(angle A4 in Figure 1, negative as shown)

The fifth lower-level control system is identical to the Azimuth head-angle control system, except that the elevation angle is perceived and the output alters the head's elevation. The integration factor is adjustable.

Higher-level control systems

There are three higher-level control systems. One perceives and controls the longitudinal distance of the fingertip from the target; the second controls the distance of the fingertip left and right of the direction of the target; the third controls the distance of the fingertip above and below the direction of the target. In the latter two systems, the angular deviations of finger and target are perceived after multiplication by the perceived distance of the objects, which is calculated as the reciprocal of the retinal disparity in each case. The model therefore perceives lateral movements through a given angle as representing larger movements when the object is farther away. This effect is analogous to visual "size constancy."

Visual distance control

The perceived distance of an object (as opposed to actual distance) is computed as the eye separation divided by the angle between the lines of sight from the two eyes to the object (the retinal disparity). The depth control system's reference signal is zero. Its perceptual signal is the difference between the perceived distances of the fingertip and the target. In this model the eyes gaze in parallel, but even if their gaze angles were independent or convergent the

same calculations would work. When the perceived fingertip distance equals the perceived target distance, whatever the nonlinearities, the finger is at the same distance as the target. The depth-difference is then perceived as zero. The absolute spatial meaning of these signals and their linearity in representing real spatial relationships are therefore irrelevant.

This distance-control system is a pure integrating system; the time-integral of the error signal becomes the output signal, which in turn becomes the reference signal for the elbow-angle control system of lower order. The combination of systems is very nonlinear, but still works.

Visual azimuth control

The second higher-level control system receives a signal representing the angular distance between the target and the fingertip as seen by the right eye. The angular deviations of finger and target are first multiplied by the perceived distance to each object, the left-right or azimuth distance then being the difference between the products. This multiplication creates an approximate "size constancy" perceptual effect. An important consequence of this effect is to keep the loop gain of this control system nearly constant for all extensions of the arm.

The reference signal for this system is zero, so the error signal is a measure of the lateral separation of fingertip and target. This system is a pure error-integrator. Its output signal becomes the reference signal for the kinesthetic azimuth control system of lower order. The center about which the arm moves is offset to the side and below the eyes; the resulting nonlinearities have little effect except at extreme positions.

Visual elevation control

The final control system receives a signal representing the vertical angle between the lines of sight from the right eye to finger and target. The individual angles are multiplied by the distance signal from the distance-control system; the difference between the products yields a perceptual signal that depends primarily on vertical distance (normal to the line of sight) from the finger to the target. The reference signal is zero. The error signal is integrated to produce an output signal. The output signal becomes the reference signal for the lower-order system that controls the vertical angle from shoulder to fingertip. Again the center about which actions take place is offset from the visual point of view.

Some details of interest

This model does not use any complicated output computations. It therefore differs from other models in which the brain is assumed to compute output signals that will have the required effects in the physical world. In this model, the perceptual computations determine what will be controlled, while the output computations consist of amplifying and slowing the error signals, and of routing an output signal to the appropriate action (or lower- order reference signal) with the correct sign.

The kinesthetic vertical angle control system interacts with the kinesthetic fingertip distance (actually elbow angle) control system. If the elbow control system moves the fingertip farther from the shoulder, it does so by straightening the arm at the elbow. This action will raise or lower the line from shoulder to fingertip. But this effect is immediately canceled by an automatic adjustment of the vertical-angle control system, which lowers or raises the upper arm. As a result, when the elbow angle control system straightens the elbow to move the fingertip farther away, the vertical angle control system moves the upper arm just enough, up or down, so that the fingertip moves outward in a straight line instead of following an arc as it would do if the upper arm remained still. The elevation angle remains essentially undisturbed. This apparent coordination of actions by two systems does not actually involve any superordinate coordinating system. The two systems act completely independently of each other. The vertical-angle system is disturbed by the distance-system, but keeps its own perceptual variable constant despite the disturbance.

The higher-level systems also interact. When the fingertip is brought closer to the eyes, its lateral and vertical visual offset is increased (this effect is only partly compensated by the perceptual size-constancy calculation). When the fingertip is far to one side and is swung toward the centerline, its distance from the eyes is also affected because of the shoulder's offset from the eyes. The vertical angle is also affected. When you slow the operation of the higher systems (larger numbers in the right- hand column of the parameter screen) or move the target very rapidly, you will see these interactions plainly. In this 1500- line program, only about 50 lines are devoted to the control systems themselves. Clearly this is a parsimonious model.

It was found that making the head move to face the target at all times made the system work considerably better than it did with a stationary head—primar-

ily because of widening the region of good control. Depth perception improved because of eliminating the narrowing of the visual angle as an object moves off to one side. In human beings the sideward range of binocular vision is limited by the nose cutting off the fields of vision; perhaps there is a good reason for having our noses where they are!

When the head is set to react slowly, you will see that large left or right target deviations will fool the model. It will think that the fingertip has moved farther way, because the angle between the lines of sight from the two eyes to the target becomes smaller. This effect disappears as the error is corrected.

Parameters

To adjust the parameters, give the "s" command to show the parameter screen. Set the parameters by moving the highlight with the arrow and enter keysand typing them in. Parameters are saved in a disk file, so the model comes up with the parameters entered at the last session each time it is re-started. To quit entering parameters, use the End key. You will hear disk activity as the new parameters are stored.

The Gain factor for the lower-level systems (first three in the left column of the parameter screen) works best at a setting of 5 to 50. There is an upper limit imposed by the discrete nature of the computations. If the gain is set too high, the correction made in each computing cycle will be larger than the system can accomplish, and there will be artificial high-frequency oscillations that actually mean nothing. To some extent this effect is compensated by large slowing factors. The gain setting determines how much output change there will be for a given amount of error.

The slowing factor for these three systems determines how fast a given output change is allowed to occur. With a large slowing factor, only a small part of the change that is calculated is allowed to take place on each cycle (the slowing factor is a divisor). In the long run the same final change will occur, but with a large slowing factor it will take longer. The slowing factor that gives optimal response is numerically equal to the gain factor for the same system, plus 1. If you make the slowing factor larger or smaller than this value, you will see differences in the way errors are corrected—watch the plots.

Three of the lower-order systems allow you to adjust both Gain and Slowing factor. This type of control is called slowed proportional control. For those who understand, the transfer function is 1/(s + a). For a pure integrator it is 1/s. The remaining systems (the two head systems in the left column of the parameter screen, and all three arm control systems in the right column) are pure error-integrating systems. For a constant error signal, the output will increase at a constant rate, in the appropriate direction. The larger the integration factor, the more slowly the output will change, because this factor enters as a divisor. Integrating control systems will eventually bring their errors exactly to zero. You will find that if you reduce the integration factor in all three higher-order systems to 1, meaning that they react as fast as possible (using integer arithmetic), the arm will be unstable in some or most positions.

An integration factor of 30 will make the corresponding integral control system extremely sluggish, but it will eventually correct its errors. It's useful to watch the action slowed down in this way—you can think of it as a slow-motion movie of normal behavior. This is a good way to see how the model would work if the computer could run faster.

If you combine a moderate integration factor in a higher-order system with a large slowing factor in the corresponding lower- order system, you will begin to see overshoot—the "mass-on-a- spring" kind of movement will appear after a sudden change of target position. There are, of course, neither masses nor springs in this system. It is assumed that the joint-angle control systems work via still lower-order systems that cancel the effects of inertia, at least on the time-scales involved here. The real mass and spring effects are hidden by normal lower-order control processes.

The plots

The left half of the screen shows three plots. The upper one shows the perception of elevation of finger relative to target, the middle one shows the perception of azimuth in the same way, and the bottom one shows the perceived difference in depth between finger and target. These plots progress gradually to the center of the screen, then that part of the screen blanks and they start over.