Arm with 14 degrees of freedom preliminary model

Arm with 14 degrees of freedom is an early effort to model an entire arm with its many joints. Even though the program is somewhat unpolished and temperamental, you can clearly see the exquisite performance of a handful of simple control systems in action. The point is that the neurological structure required to control a flailing arm, with multiple finger and thumb joints, moving freely in space, can be remarkably simple. When you begin to play with this model, it is a good idea to start out with slow, methodical mouse movements. Sideways only. Up and down has no effect.

Annotated screen shot illustration

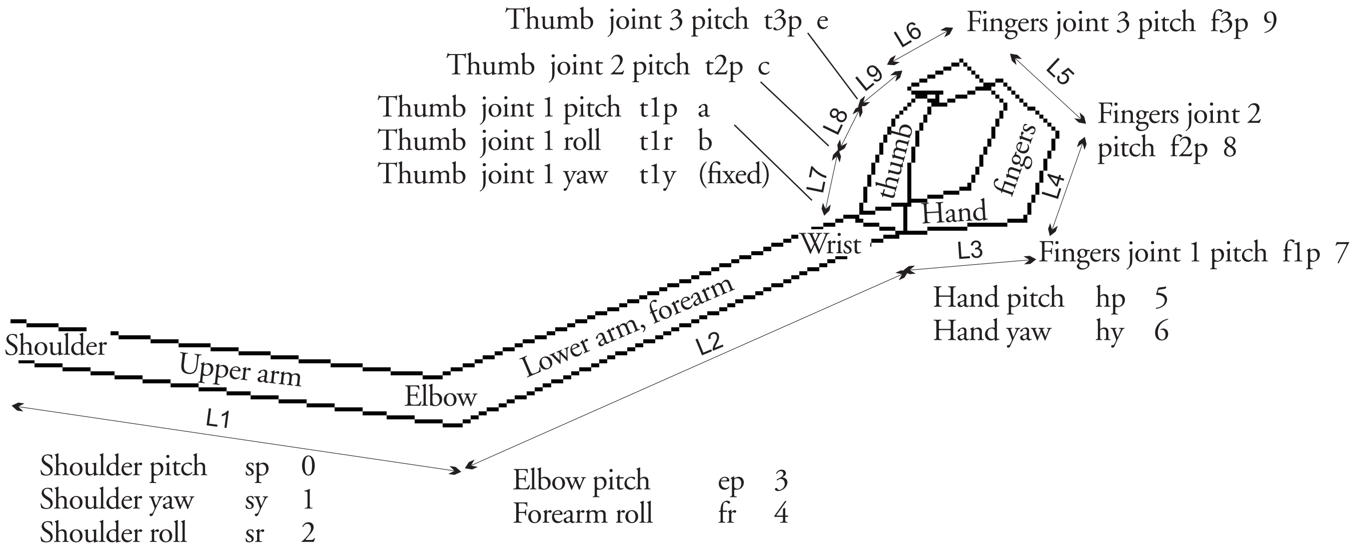
After pressing F2 (or when the program first starts)

Menu on screen	*Equivalent detail in writeup on next page*	
Radius 0	Distance from shoulder to wrist	0
Elevation...1	Elevation of shoulder-wrist line	1
Azimuth.....2	Azimuth of shoulder-wrist line	2
Arm Roll....3	Roll about shoulder-wrist line	3
Hand Yaw....4	Yaw about vertical axis through wrist	4
Hand Pitch..5	Pitch about horizontal axis through wrist	5
Hand Roll...6	Roll about axis through center of closed hand	6
Grasp.......7	Closure of fingers and thumb as a unit	7

Quit: ESC

Viewing angle:	
Up	u
Down	d
Right	r
Left	l
Reset	z

After pressing F1 — See instructions on next page



Here is the writeup in the source code for the program 14deg.c

Labeling of variables for parts of model. Details below

L1 ... L9 are lengths. p, y, and y means pitch, yaw, and roll. All the fingers move together en bloc; the thumb moves separately. s = shoulder, e = elbow, f = forearm, h = hand, f = fingers, t = thumb, numbers = which finger or thumb joint.

Operating instructions: using x direction of mouse movement to affect model appearance. Y direction has no effect. Note that the word "LIMIT" appears on the screen when ANY joint is at its physiological limit. Some experimenting may be required to figure out which joint it is.

After pressing F1, typing a number or letter gives the user control of a single degree of freedom using the x dimension of mouse movement (or + and - keys), as listed below

Shoulder pitch	sp	0	Fingers joint 2 pitch	f2p	8
Shoulder yaw	sy	1	Fingers joint 3 pitch	f3p	9
Shoulder roll	sr	2	Thumb joint 1 pitch	t1p	a
Elbow pitch	ep	3	Thumb joint 1 roll	t1r	b
Forearm roll	fr	4	Thumb joint 1 yaw	t1y	(fixed)
Hand pitch	hp	5	Thumb joint 2 pitch	t2p	c
Hand yaw	hy	6	Thumb joint 3 pitch	t3p	e
Fingers joint 1 pitch	f1p	7	(notice that letter d has been skipped)		

After pressing F2 (or when the program first starts), typing the numbers 0 to 7 gives the mouse x-axis (or the + and - keys) control over seven artificial degrees of freedom created by seven control systems that affect combinations of the joint angles. The reference signals for unselected systems remain at the last setting. Thus if the pitch about a horizontal axis through the wrist (5) is set to make the hand level, the hand will remain level while distance from shoulder to wrist (0) is varied. If the hand is rolled (6), the pitch angle about the wrist (5) still pitches the hand, even though this must now be accomplished in part or totally by an actual yaw at the wrist joint. Commands for switching control as follows:

Distance from shoulder to wrist	0	Yaw about vertical axis through wrist	4
Elevation of shoulder-wrist line	1	Pitch about horizontal axis through wrist	5
Azimuth of shoulder-wrist line	2	Roll about axis through center of closed hand	6
Roll about shoulder-wrist line	3	Closure of fingers and thumb as a unit	7

Note 1: there is an offset in the hand pitch so that the roll (number 6) takes place about a line that goes from the wrist through the center of grasp when the hand is closed (using 7). This offset could also be an adjustable variable. Without the offset, the roll would be about a line lying in the back of the hand.

Note 2: number 3 is not correctly integrated into the model; changing the upper arm roll has a number of strange effects. It works best when the hand is rolled to a palm-vertical or palm-down position, (6) above.

The keys 'u', 'd', 'r', and 'l' move the user's viewpoint of the arm (up, down, right, and left). The 'z' key zeros the viewpoint angles.

An arm image is created by loading points into an array, then applying rotations about the x, y, or z axes beginning with the most peripheral points (the hand) and progressing toward the shoulder. The result is an array describing an arm in a new configuration, which is then projected into two dimensions and drawn in perspective using the drawpoly function of Turbo C 2.0.

Limits are placed on arm movements similar to those in a real (right) arm. When any joint reaches a limit (whether the one being adjusted or not) the indication "L I M I T" appears on the screen and that joint angle stops changing.

Graphics initialization is self-contained. The Borland .bgi files must be present in the same directory as the program.