STEP BY STEP TO ROBOT CONTROL

The task of controlling a robot is usually two-fold, and can be summarised as

```
repeat
     go There;
     do That
until finished;
```

`There` and `That` are arbitrary, which is what you'd expect for a general-purpose machine. The aribitrariness of `That` is great, as it depends on the end-effector; there is little of general relevance that can usefully be said about it.

This sheet is about `There`, which is the problem of coping with space, and is much more generally applicable. Coping comes in two parts : the fundamental problem is to be able to deal with the geometry involved in robots, which is mainly a matter of transforming between sets of spatial coordinates which we find easy to manage – typically cartesian or polar coordinates – and the set of joint angles and offsets, which we can call the joint coordinates. In practice, we have to do a lot of these transformations, so it's important to find efficient ways of doing them.

(It can be even worse : the joint coordinates are not always the same as the actuator coordinates. If a joint angle is controlled by a hydraulic ram working between the two links which meet at the joint, then we have to do yet more trigonometry to find out what to do with the actuator itself.)

After the geometry comes the physics. The spatial problems aren't merely abstract geometry; we want to know the coordinates so that we can move largish chunks of (usually) metal around in space, and to do that we have to worry about dynamics.

It's illuminating to think of the development in six stages.

FORWARD KINEMATICS.

This is an easy problem. (It is the only easy problem.) The requirement is to work out the robot's state given the details of what each joint is doing. It's solved in principle in the sheet *ROBOT GEOMETRY*; using the information collected there, it is straightforward to combine the parameters of all the robot links and joints to work out its overall configuration.

INVERSE KINEMATICS.

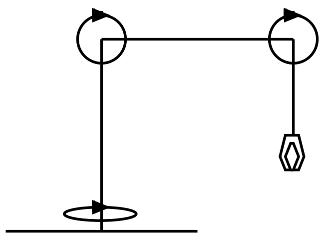
It is unfortunate that we rarely, if ever, want to solve the forward kinematics problem. The inverse kinematics problem is much more useful, and usually far more difficult : given the required coordinates of the robot's end effector, what joint coordinates must we set to get it there ?

The question is easy to answer if the joint coordinates are simply related to our preferred coordinates. The joint coordinates of gantry robots are in effect the cartesian coordinates, and the problems vanish as by magic. Sadly, gantry robots are not very well suited to most work which people want robots to do, and the robots which are suited are nothing like so nice mathematically. SCARA robots are not too bad, because one coordinate (height) is separated from the others, but the most common arm robots are horrendous. It's a pity that they work so well ......

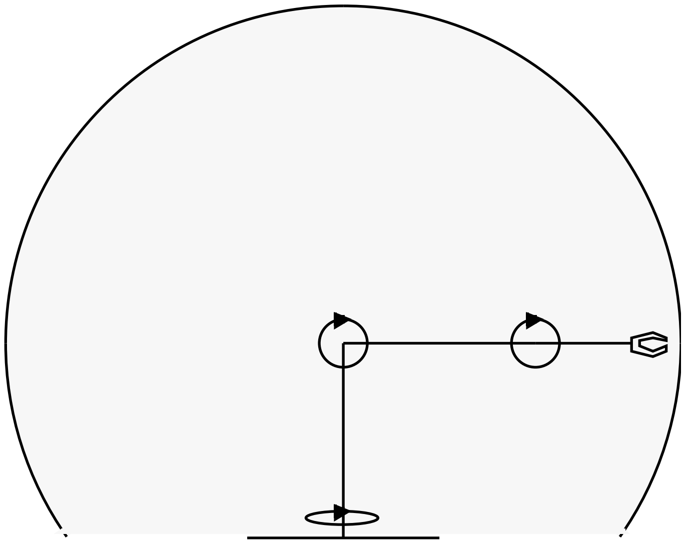
Generally, the problem can be expressed as the solution of a set of simultaneous equations, most often fiercely non-linear. I shall not go into details; you'll find them in books on robotics. All we're concerned with at the moment is the principles.

One of the principles is whether or not there's a solution to the equations. Generally, there might be zero, one, or more solutions.

- If there is **no solution**, then the robot can't reach the point specified. The point might simply be too far away, or the robot might not be able to approach it from the required direction. This leads to the idea of *workspaces*, otherwise called *envelopes*. A robot has a reachable workspace, which covers all the points it can touch, but is often not very relevant except as an absolute outer bound to where it can get and as a guide to where to put the safety barriers. There are also *dexterous workspaces*, which are regions where the robot can reach from any direction. Generally, if you impose some constraint on how you want the robot to behave, you can define a corresponding dexterous workspace. Here's a picture of a robot and its reachable workspace :

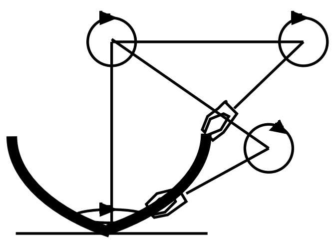


Robot



Reachable workspace

Here is (a rough impression of) the dexterous workspace, defined as the space within which the gripper can be pointed directly towards the base joint, assuming that the minimum angle between the final two links is 45˚ :

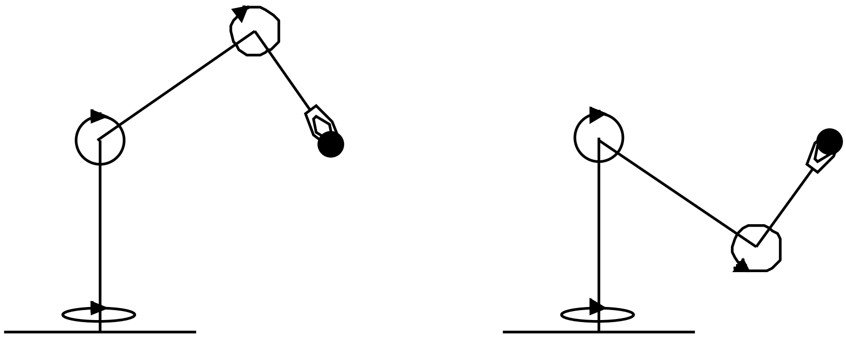


Dexterous workspace

The workspace is the thick line. This is a trick to make the point; this robot is exceedingly limited, and just doesn't have the flexibility to show up well in this sort of comparison, but in practice dexterous workspaces can be surprisingly small even for quite capable robots.

- There is **one solution**. Easy. Do it. But that probably means that the robot can only just reach the point, and there might be difficulties in any sort of manoeuvring you need.

- There are **several solutions**. These are usually alternative ways of arranging the links round the sides of polygons, as in the diagram :



(There are also two other solutions with the trunk rotated through 180˚ and the arms rotated backwards over to the other side, but they look just the same as these.) In this case, the problem is to select the most useful of the available solutions, which is usually that which is closest to the robot's previous position.

TRAJECTORIES.

It is rarely useful only to know where the robot must finish up; if you just set all the coordinates as final set points for the motor controllers and leave it to its own devices, the behaviour is not very predictable. It is certainly not likely to take a straight-line path, or anything else easily recognisable; the actuators will move at uncoordinated speeds, and the resultant trajectory of the end effector is left to chance.

To achieve a desired trajectory, much more careful analysis is required. At the very least, it is necessary to evaluate the required joint coordinates as "enough" points along the path, and to take care to make sure that the various actuators keep "sufficiently" in step. The precise meanings of the two words in quotation marks depends on how precisely you want the path to be followed.

DYNAMICS.

Once you know where, and how fast, you want to move the various joints, you can start working on the physics. If you know the robot coordinates and the physical properties – masses, moments of inertia, etc. – of its parts, you can work out how hard you have to push to cause the movements you want, and thence – given the properties of the actuators – how much current you want to put through the motors as a function of time.

POSITION CONTROL.

If you can do all that, your robot should behave reasonably well, but it's behaviour is unlikely to be perfect. For very precise manipulations, it is useful to have some form of sensory feedback which directly measures the position of the end effector. You can then use this for fine adjustment of the configuration in the final approaches.

FORCE CONTROL.

Getting to the right place is not necessarily enough. If the robot is performing some task such as drilling or grinding with its end effector, it must apply a force to the workpiece in some appropriate direction. Further calculation is necessary to work out how the several actuators should be driven to exert the required force in the required direction.

Alan Creak,
April, 1997.