Maths Skills (MTCS) G5071

Lecture 7

Kingsley Sage
Room 5C16, Pevensey III
khs20@sussex.ac.uk

Lecture 7

- Vector applications

Vector functions

- The common functions e.g. sin and log take a number as an argument and produce (or "map onto") another number – nothing to stop us extending the idea of a function to include vector functions.
- The wind velocity vector at a point has two components (one might represent it using northerly and easterly components for example).
- If the wind velocity was represented by a vector ***v***, and position on the ground by a vector ***r***, then we can write a 2-D function ***v***=*f(r).*
- Often the shorthand notation ***v(r)*** is used to indicate that ***v*** depends on ***r***, without giving the function a separate name

Vector functions

- If the vertical component of the wind velocity was represented by a well, and the position included height above ground, then we have a function from 3-D position vectors to 3-D velocity vectors.
- Manipulation of large arrays representing approximations to such functions is one of the main tasks of the UK Meteorological Office (the weather office).

Optical velocity fields

- The *optical velocity field* is useful for studies of perception in the control of robot and animal locomotion.
- Suppose a camera attached to a robot is moving through the world – the image formed by the camera will be changing.
- We cam imagine drawing arrows on the image representing speed and direction of motion of features at a given moment.
- In practice, an approximation to the image velocity field is easy to obtain – get two images from different positions of a camera (not too far apart) superimpose them and join the corresponding features.

Optical velocity fields

- In general, this velocity field will be a complex function like the wind field over Sussex – there is no simple function for it.
- If the following apply:
  - the camera is viewing a smooth surface;
  - the field of view is reasonably small; and
  - the camera is panned and tilted so that it tracks a feature at the centre of the image, which is taken as the origin for position;
- then the image velocity field is approximately a linear function of image position.

Matrix multiplication as a field operation

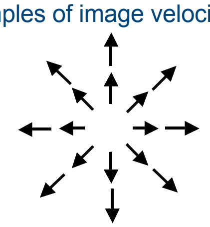
- If position in the image is represented by ***r*** and the image velocity by ***v***, the relationship is just a matrix multiplication: ***v=Mr …***
- where ***M*** is a 2×2 matrix – 4 components of ***M*** depend on direction the camera is moving and slant and tilt of the surface …

Matrix multiplication as a field operation

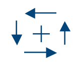
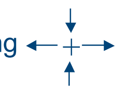
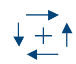
$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \text{ mapping is a dilation (see below) } \begin{array}{ccc} (1,0) & \rightarrow & (0.5,0) \\ (1,2) & \rightarrow & (0.5,1) \\ (0,-1) & \rightarrow & (0,-0.5) \end{array}$$

etc.

Plot out some examples of image velocity field as an arrow diagram.

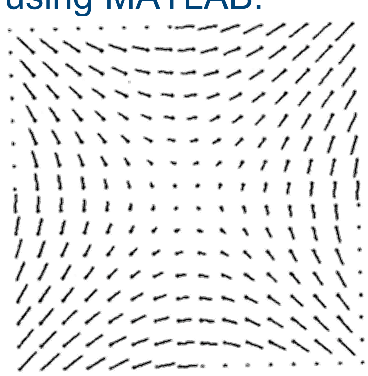


Flow patterns

- The effects of other forms of ***M*** are easy to calculate:
  - $\begin{bmatrix} 0 & -0.5 \\ 0.5 & 0 \end{bmatrix}$ – produces rotation mapping 
  - $\begin{bmatrix} 0.5 & 0 \\ 0 & -0.5 \end{bmatrix}$ – produces shear component 1 mapping 
  - $\begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$ – produces shear component 2 mapping 

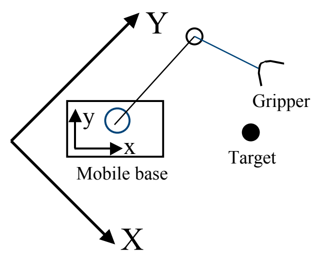
Flow patterns

- We can generate representations of flow patterns using MATLAB:



Co-ordinate transformations for control

- A second application is the control of robot arms – outlined here to give the general idea.
- Suppose a robot's gripper is operated from some base, and designed so that it can be moved on command to a given position above a plane, expressed in *(x,y)* co-ordinates.



- The controller for the mobile base can turn it to face in any direction and can move it around the lab – its position in the lab is also expressed in Euclidean co-ordinates, relative to some axes fixed to the floor, expressed in *(X,Y)* co-ordinates.
- The orientation of the base is indicated by the angle θ, which is the angle anti-clockwise from the X-axis on the floor to the x-axis on the base.

Simple robot arm kinematics

- If the position of an object in the room is specified in terms of *(X,Y)* co-ordinates, how do we move the gripper to it?
- Since the gripper control works in terms of *(x,y)*, we need to work out where the target is relative to these co-ordinates.
- Suppose robot base (origin of its *(x,y)* co-ordinates) is at position ***B*** relative to the floor co-ordinates systems, and the target is at position ***T***. What we need is the target position in the base co-ordinates, *t* - we can then move the gripper directly to it, provided that it is in range.
- The formula to get from ***T*** and ***B*** to *t* is simply:

$$\mathbf{t} = \mathbf{R}(-\theta)(\mathbf{T} - \mathbf{B})$$

- Here ***R(-θ)*** means 2×2 matrix that rotates a vector anti-clockwise through an angle $-\theta$ I.e. matrix-vector multiplication using:

$$\mathbf{R}(-\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Why this formulation?

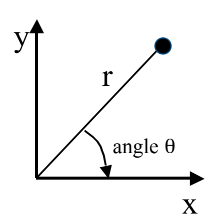
- We can see why the formula is correct; ***T-B*** is the vector from the base to the target in floor co-ordinate frame, and we need to express it in base co-ordinate frame, which has been rotated anti-clockwise by θ.
- It is quick and easy to write down and implement – since we know what a rotation matrix looks like.
- It is easy to manipulate – for example, we might want to go in the other direction with a robot that has a sensor on its base that detects the target and produces the components of *t*.
- We might then want to know ***T*** in order to plot the position of the object in a map that is being built of the lab – equation reversed:

$$\mathbf{T} = \mathbf{R}(\theta)t + \mathbf{R}$$

- The method generalises in a variety of ways – mist obvious is to 3-D equations stay the same, though rotation matrix becomes 3×3.
- Also easy to incorporate additional links into the chain – for each of these there is a co-ordinate transformation.

Alternative Co-ordinate systems

- So far, vectors represented using Euclidean *(x,y)* co-ordinates – but often useful to represent then in polar co-ordinates.
- In 2-D polar co-ordinates, a position is represented by its distance from the origin, *r*, and the angle a line joining it to the origin makes with the x-axis θ e.g. eye position in a visual system.



- The relationships between polar and Euclidean co-ordinates are:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \operatorname{atan}(y/x) \text{ if x is positive}$$

$$\theta = \operatorname{atan}(-y/x) \text{ if x is negative}$$

Next time …

- Introduction to probability and statistics …