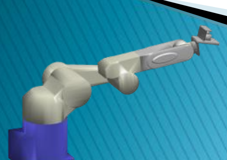
Week 5

<u>Schedule</u>

- Introduction (10 mins)
- Quiz 2:
  - Individual (30 mins) (in Quiz groups)
  - Buffer (5 mins)
  - Team (25 mins)
- Review & discuss Lab 4 (5 mins)
- Introduction to Lab 5 (5 mins)
- Assignment 1 update
  - Simulated Robots in ROS
- 3 breakout rooms (30 mins)
- Tutors available for call-backs (till end of class)



Overview of Week 5: Quiz 2

- In total worth 5%
- Individual quiz
  - worth 4%
  - will go from lab start time for 30 minutes
  - 10 questions (1 question from each category)
  - No talking
  - New question

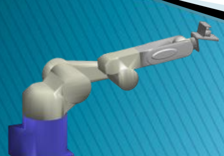
Share your screen with the tutor and they will give you the answer via a private chat.
DO NOT GUESS THIS QUESTION
A5

You will share your screen with your tutor and in the private chat the tutor will reply something like "the answer is 123".
It is a multiple choice question so you simply select that answer given to you,
If you get it correct (which everyone should), you get a bonus 10 marks.
But if you get this question wrong you will get **0%** for the whole quiz and I'll call you to discuss all your quiz answers and working,
So you **<u>must</u>** get it correct! Ask the tutor and please **<u>do not guess</u>** the answer.

- Group quiz
  - worth 1%
  - will start immediately after the individual quiz
  - will go for 20 minutes
  - 20 questions (2 question from each category)
  - Lots of talking

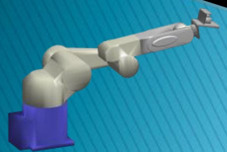
Question Categories (10)

- Baxter
  - Distance between arms in certain joint states
  - Distance between bases and/or arms in certain joint states
  - Straightforward Inverse kinematics using robotics toolbox
- Puma 560
  - Distance from end effector to surface in environment
  - Distance from end effector to surface in environment given new information about base location
  - Point in the end effector’s coordinate frame
  - Straightforward Jacobian analysis
  - Straightforward Inverse kinematics using robotics toolbox
- Self collisions for a Hyper Redundant 2D arm
- D&H Transforms and forward kinematics in Matlab



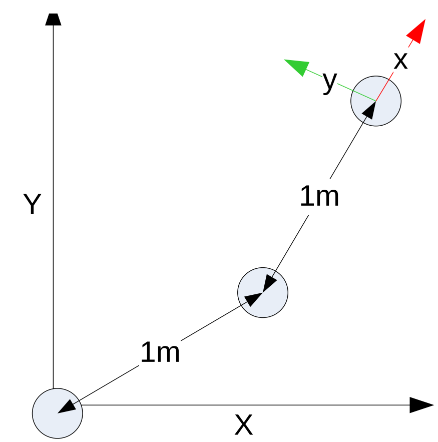
Quiz password

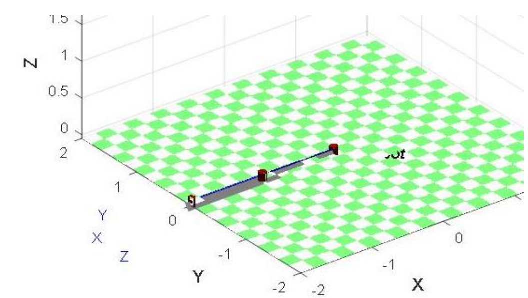
- Individual
- Team



Review of Lab 4 – Q1

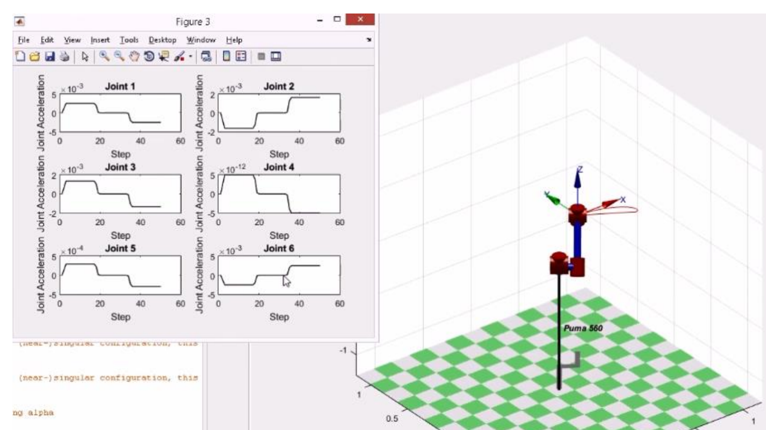
- Draw with 3DOF Planar Arm so the Z axis faces down ways
- What did you observe when using "teach" to move the robot?
- Use ikine and mask out the impossible-to-alter values
- Using ikine draw a line. What was observed about its "straightness"?
- Trace out a circle using ikine. What was observed about limits and collisions?





Review of Lab 4 – Q2 and Q3

- Q2: Inverse Kinematics to Determine Joint Angles of Puma 560.
  - Did you have any cases that didn't work?
  - What error messages?
  - Did you get a joint angle returned that didn't allow the robot to be at the requested transform
  - Does ikine obey joint limits?
  - What other inverse kinematic solver is in the toolbox? Does it obey joint limits?
- Q3 Interpolation: Get from A to B using puma560. What was learned from this exercise about:
  - Quintic Polynomial Profile
  - Trapezoidal Velocity Profile



Introduction to Lab 5

- Q1: Help the Robot Blaster save the Planet from UFOs! Develops understanding of
  - Line-plane intersections
  - Inverse kinematics
  - Gamifying Matlab
  - Note: Lab5SolutionQuestion1Skeleton.m

- **Q2: Collision checking** for 3-link planar robot
- **Q3: Collision avoidance** for 3-link planar robot

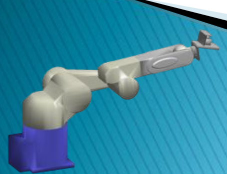
Pre-work before Week 6 lab

- Completed week 5 lab exercises
- Ensure you have achieved 80% in Quizzes 1&2
- Read textbook
  - Section 3.2 (pages 51-56): "Time Varying Coordinate Frames"
  - Appendix E (pages 517-522): "Ellipses"
- Watch videos on UTSOnline
- **<u>Complete Lab Assignment #1 (due next week)</u>**

Note/slides from the textbook

- Textbook readings:
  - Section 3.2 (pages 51–56): “Time Varying Coordinate Frames”
  - Appendix E (pages 517–522): “Ellipses”

- Although it is better to read the textbook, some notes (in slide format) have been summarised below



Time Varying Coordinate Frames

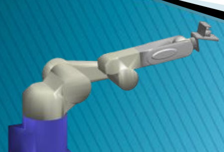
- The translational velocity is the rate of change of the position of the origin of the coordinate frame. Rotational velocity is a little more complex.

Rotating Coordinate Frame

- A body rotating in 3-dimensional space has an angular velocity which is a vector quantity

$$\omega = (\omega_x, \omega_y, \omega_z).$$

- The direction of this vector defines the instantaneous axis of rotation, that is, the axis about which the coordinate frame is rotating at a particular instant of time.
- The magnitude of the vector is the rate of rotation about the axis – in this respect it is similar to the angle-axis representation for rotation.



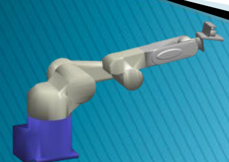
Rotating Coordinate Frame (continued…)

- From mechanics there is a well known expression for the derivative of a time-varying rotation matrix:

  $$\dot{\boldsymbol{R}} < t > = \mathrm{S}(\omega)\mathbf{R}<\mathrm{t}>$$

- Where $R < t > \in SO(2)$ or SO(3) and S(·) is a skew-symmetric matrix that, for the 3-dimensional case, has the form

$$\mathrm{S}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$



Rotating Coordinate Frame (continued…)

- we might ask what does $\dot{\boldsymbol{R}}$ mean?

$$\dot{\boldsymbol{R}} \approx \frac{R<t+\delta_t> \; - \; R<\delta_t>}{\delta_t}$$

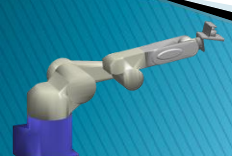
- which we rearrange as

$$\boldsymbol{R} < t + \delta_t > \approx \delta_t$$

- and substituting Eq. 3.4 we obtain

$$R<t+\delta_t> \; \approx \delta_t S(\omega)\boldsymbol{R}<t> \; \approx (\delta_t S(\omega) + I_{3x3})\boldsymbol{R} < t >$$

- which describes how the orthnormal rotation matrix changes as a function of angularvelocity.



Incremental Motion

- Consider a coordinate frame that undergoes a small rotation from $\boldsymbol{R_0}\ to\ \boldsymbol{R_1}$. We can write as:

$$\boldsymbol{R_1} = \delta_t\ S(\omega)+I_{3x3})\boldsymbol{R_0}$$

- and arrange as:

$$\delta_t S(\omega) = \boldsymbol{R_1}\boldsymbol{R_0^T} - \boldsymbol{I_{3x3}}$$

- and then apply the vex operator, the inverse of $S(\cdot)$, to both sides

$$\delta_\theta = vex(\boldsymbol{R_1}\boldsymbol{R_0^T} - \boldsymbol{I_{3x3}})$$

- where $\delta_\theta = \delta_t\omega$ a 3–vector with units of angle that represents an infinitesimal rotation about the world x–, y– and z–axes.

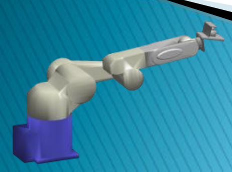
Incremental Motion (continued…)

- Given two poses ξ_0 and ξ_1 that differ infinitesimally we can represent the difference between them as a 6-vector

$$\delta = \Delta(T_0, T_1) = \begin{bmatrix} t_1 - t_0 \\ vex(\boldsymbol{R_1}\boldsymbol{R_0^T} - \boldsymbol{I_{3x3}}) \end{bmatrix}$$

- where $T_0 = (R_0, t_0)$ and $T_1 = (R_1, t_1)$
- The inverse operation is: $\xi = \Delta^{-1}(\delta)$
- and for homogenous transformation representation is:

$$T = \begin{bmatrix} S(\delta_\theta) & \delta_d \\ 0_{3x1} & 0 \end{bmatrix} + I_{3x3}$$



Inertial Navigation Systems

- An inertial navigation system is a “black box” that estimates its velocity, orientation and position with respect to the inertial reference frame (the universe).
- It has no external inputs such as radio signals from satellites and this makes it well suited to applications such as submarine, spacecraft and missile guidance.
- An inertial navigation system works by measuring accelerations and angular velocities and integrating them over time.

Inertial Navigation Systems (continued…)

- A **discrete-time** such as:

$$\mathbf{R} < \mathrm{k} + 1 > = \delta_t S(\omega)\mathbf{R}<k>+\mathbf{R}<k>$$

  is used to numerically integrate changes in pose in order to estimate the orientation of the vehicle.
- The measured acceleration $B_a$of the vehicle's body frame is rotated into the inertial frame

$$0_a = 0_{R_B} B_a$$

  and can then be integrated twice to update the estimate of the vehicle's position in the inertial frame.

Ellipses

- An ellipse belongs to the family of planar curves known as conics.
- The simplest form of an ellipse is defined implicitly: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$

Ellipses (continued…)

- We can write the ellipse in matrix quadratic form as:

$$\begin{pmatrix} x & y \end{pmatrix} \begin{bmatrix} {}^{1}/_{a^2} & 0 \\ 0 & {}^{1}/_{b^2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1$$

$$x^T \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix}^{-1} x = 1$$

$$x^T E^{-1} x = 1$$

Ellipses (continued...)

- In the most general form *E* is a **symmetric matrix** $E = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$
- and its determinant $\det(E) = AB - C^2$ defines the type of conic

$$\det(E) \begin{cases} > 0 \; ellipse \\ = 0 \; parabola \\ < 0 \; hyperbola \end{cases}$$

Ellipses (continued...)

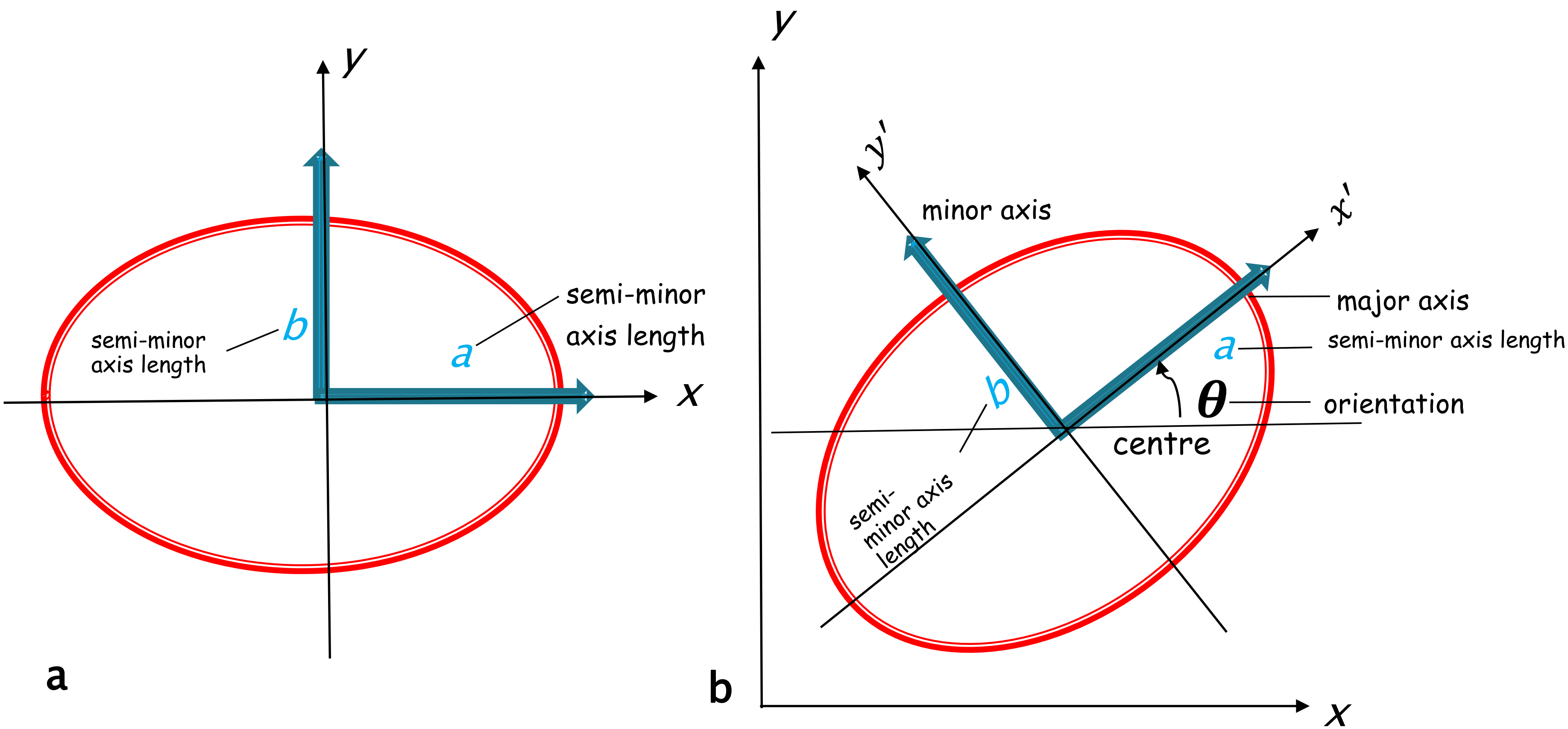


Fig. E.1. Ellipses. **a** Canonical ellipse centred at the origin and aligned with the *x*– and *y*–axes; **b** general form of ellipse

Ellipses (continued…)

- An ellipse is therefore represented by a positive definite symmetric matrix ***E***.
- Non-zero values of C change the orientation of the ellipse. The ellipse can be arbitrarily centred at x_c by writing it in the form $(x - x_c)^T E^{-1}(x - x_c) = 1$
- which leads to the general ellipse $E = X\Lambda X^T$
- where ***X*** is an orthogonal matrix comprising the eigenvectors of ***E***. The inverse is $E^{-1} = X\Lambda^{-1}X^T$
- so the quadratic form becomes

$$x^T X\Lambda^{-1}X^T x = 1$$

$$(X^T x)^T \Lambda^{-1}(X^T x) = 1$$

$$x'^T \Lambda^{-1} x' = 1$$

Ellipses (continued…)

- Alternatively the ellipse can be represented in polynomial form.

$$\left(x-(x_0,y_0)\right)^T\begin{bmatrix} a & c \\ c & b \end{bmatrix}\left(x-(x_0,y_0)\right)=1$$

  - and expand we obtain

$$e_1x^2+e_2y^2+e_3xy+e_4x+e_5y+e_6=0$$

  - where e_1=a,$e_2=b, e_3=2c, e_4=-2(ax_0+cy_0)$, $e_5=-2(by_0+cx_0)$

  $and\ e_6=ax_0^2$+by_0^2+2x_0y_0-1

- For a non-degenerate ellipse $e_1\neq\ 0$ and we rewrite the polynomial in normalized form

$$x^2+E_1y^2+E_2xy+E_3x+E_4y+E_5=0$$

Properties

- The area of an ellipse is πab and its eccentricity is

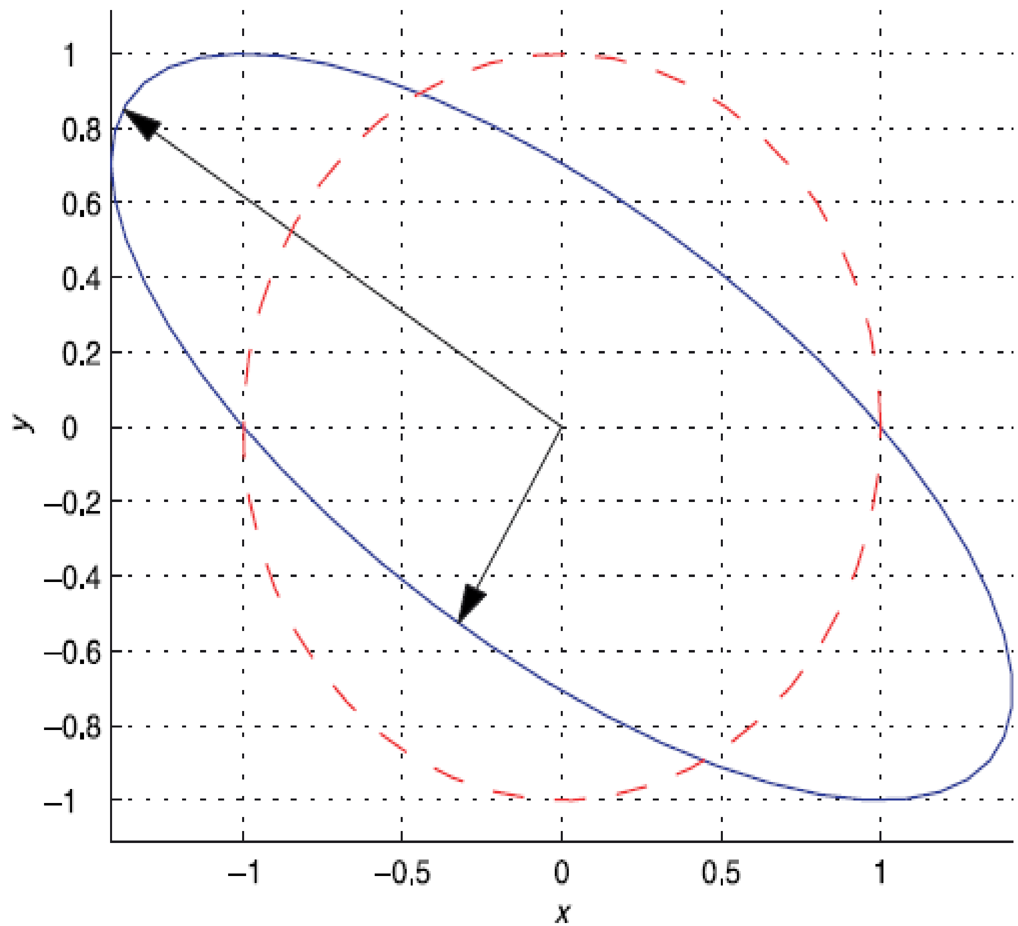
$$\varepsilon = \frac{\sqrt{a^2 - b^2}}{a}$$

- The eigenvectors of ***E*** define the principal directions of the ellipse and the square root of the eigenvalues are the corresponding radii.
- Consider the ellipse

$$x \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}^{-1} x = 1$$

Properties (continued…)

- **Fig. E.2.** Ellipse corresponding to symmetric 2×2 matrix, and the unit circle shown in red. The arrows indicate the major and minor axes of the ellipse



Drawing an Ellipse

- In order to draw an ellipse we first define a point $y = [x, y]^T$ on the unit circle $y^T y = 1$ and rewrite as $x^T E^{-\frac{1}{2}} E^{-\frac{1}{2}} x = 1$
  - where $E^{\frac{1}{2}}$ is the matrix square root
  - Equating these two equations we can write

$$x^T E^{-\frac{1}{2}} E^{-\frac{1}{2}} \mathrm{x} = y^T y$$

  - It is clear that $y = E^{-\frac{1}{2}} \mathrm{x}$ which we can arrange as: $\mathrm{x} = E^{-\frac{1}{2}} \mathrm{y}$
  - which transforms a point on the unit circle to a point on an ellipse.
  - If the ellipse is centered at x_c rather than the origin we can perform a change of coordinates:

$$(x - x_c)^T E^{-\frac{1}{2}} E^{-\frac{1}{2}} (x - x_c) = 1$$

- from which we write the transformation as: $x = E^{-\frac{1}{2}} + x_c$

Fitting an Ellipse to Data From a Set of Interior Points

- A common approach is to find the ellipse that has the same mass properties as the set of points.
- From the set of *N* points $x_i = (x_p, y_i)$ we can compute the moments

$$m_{00} = N$$

$$m_{10} = \sum_{i=1}^{N} x_i$$

$$m_{01} = \sum_{i=1}^{N} y_i$$

Fitting an Ellipse to Data
From a Set of Interior Points (continued…)

- The centre of the ellipse is taken to be the centroid of the set of points

$$(x_c, y_c) = \left[\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right]$$

  - which allows us to compute the central second moments

$$\mu_{20} = \sum_{i=1}^{N} (x_i - x_c)^2$$

$$\mu_{02} = \sum_{i=1}^{N} (y_i - y_c)^2$$

$$\mu_{20} = \sum_{i=1}^{N} (x_i - x_c)(y_i - y_c)$$

Fitting an Ellipse to Data From a Set of Interior Points (continued…)

- The inertia matrix for a general ellipse is the symmetric matrix

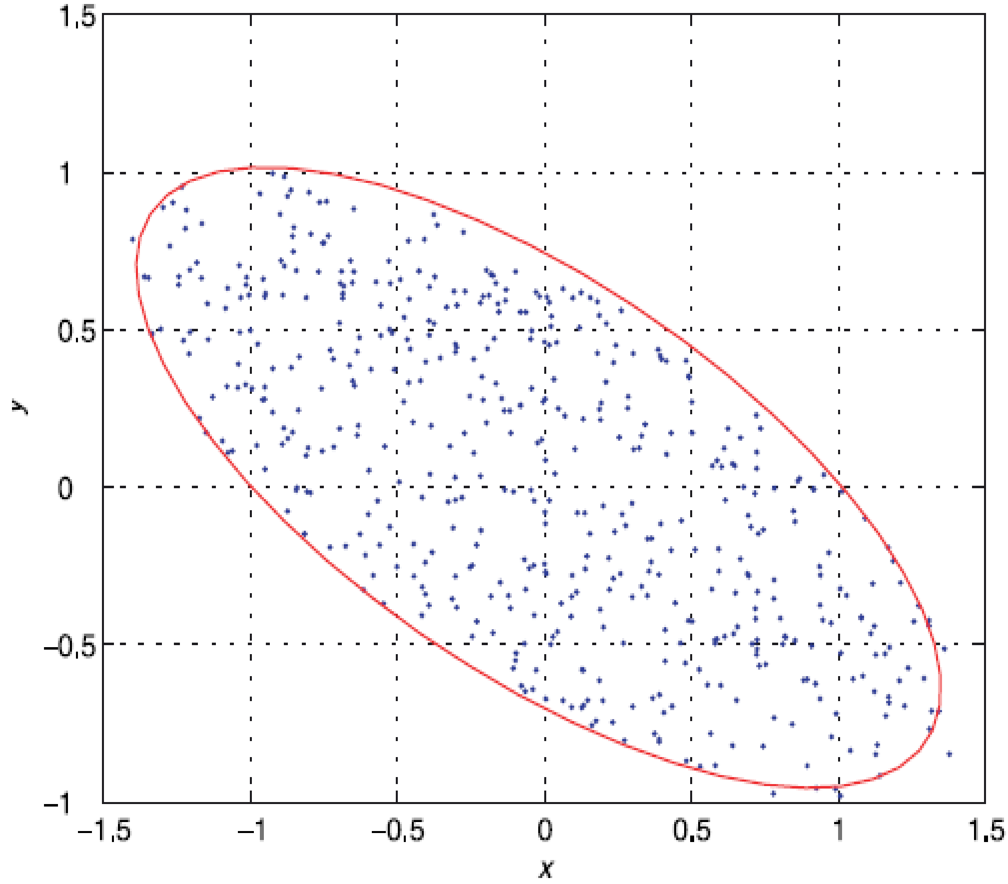
$$J = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$$

- where the diagonal terms are the moments of inertia and the off-diagonal terms are the products of inertia. Inertia can be computed more directly by

$$J = \sum_{i=1}^{N} (x - x_c)(x - x_c)^T$$

Fitting an Ellipse to Data From a Set of Interior Points (continued…)

- **Fig. E.3.** Point data



Fitting an Ellipse to Data From a Set of Interior Points (continued…)

- The relationship between the inertia matrix and the symmetric ellipse matrix is

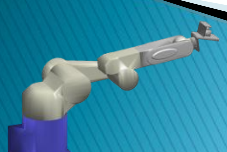
$$E = \frac{4}{m_{00}} \boldsymbol{J}$$

From a Set of Boundary Points

- Using the polynomial form of the ellipse for each point we write this in matrix form

$$\begin{bmatrix} y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ & & \vdots & & \\ y_N^2 & x_N y_N & x_n & y_N & 1 \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \end{bmatrix} = \begin{bmatrix} -x_1^2 \\ -x_2^2 \\ \vdots \\ -x_N^2 \end{bmatrix}$$

and for $N \geq 5$ we can solve for the ellipse parameter vector.



References

- [1] Robotics, Vision and Control. Peter Corke