MRC

MIDTERM

```
&Sigma;
&Lambda;
$ V^t $
$\det(A) \neq 0$
$P(x_1|y_1) = \frac {P(y_1|x_1) * P(x_1)}{P(y_1)}$
$E[X] = \sum_{x}{xP(x)}$
```

▼ Frames

Orientation of a body is defined by attaching a coordinate system to the body and understanding relation between body and reference frame using a rotation matrix

▼ **Homogeneous Transform** - 4x4 matrix combining rotation and translation into a single compact form, representing transformation between coordinate frames

Gimbal lock- having diff rotation angles but same rotated vector, 3D -losing a degree of freedom, when 2 or more axis align in a parallel config, then the rotations are avail in 2D

Inverse transforms: Inverse of rotation matrix is transpose, translation is negation

- **Transformation matrix T** → If T is a square matrix, the inverse T^(-1) can be obtained using the formula: $T^{(} - 1) = (1/det(T)) * adj(T)$ *where det(T) is the determinant of T and adj(T) is the adjoint (adjugate) of T (cofactors matrix)*

Fixed angles- all rotations take about an axis in fixed reference frame

Euler angles- set of angles describing orientation of an object around the coordinate axes **zyx -yawpitchroll**

Eigenvector of a matrix A is a vector that doesn’t change its direction when multiplied by A, the eigen vectors of covariance matrix represent the direction of max variance in the data i.e. obeys

```
Ax= &lambda; x where &lambda; is constant called eigenvalue
- Eigenvalues of cov matrix, represent the amount of variance
explained by each principal component
Eigenvalues - if matrix has non-zero determinant,
all eigenvalues are non zero
```

Orthornormal basis of a vector space is a basis where all are unit vectors are orthogonal to each other **— atan2 belongs to -180 to 180**

▼ Matrices

Condition number ||A^{-1}|| ||A|| → whether matrix is susceptible to slight changes

ill-condition can be fixed with epsilon

Isometry - Transformation of vector space such that it preserves distances between every pair of points - if T is linear and also preserves orientations, its called **rotation or mirror matrix**

The characteristic polynomial of a matrix is obtained by subtracting the eigenvalue from the diagonal entries and taking the determinant. The determinant is the product of the eigenvalues.

Matrix decomposition-

- P(LU) facotizes matrix into product of Permutation, Lower triang and upper triang
- Polar decomposition decomposes a matrix into a product of a rotation matrix and a positive semi-definite Hermitian matrix. It is useful in robotics for separating the rotational and scaling components of a transformation.

▼ Matrices type

Rotation matrices: rotates a matrix through rotation R and is same as rotation that describes a frame rotated by R relative to reference frame—Orthogonal, inverse is equal to transpose, and det is +1(anti-clock) or -1, ensuring matrix is non-singular

Non-singular matrix - Det = non-zero,inverse exists, all eigenvalues non-zero

Singular matrix - Det = 0, inv doesn’t exist - some or all eigenvalues are 0

Matrix Rank- max num of linearly independent rows and cols

▼ Matrix norms

Norm is a function that assigns a positive length to all vectors in a vector space

Spectral norm of A == max eigenvalue of A^t A == singular value

Matrix norm measures the maximum stretch of a vector for a given vector norm, max can be determined by taking max over a unit circle

Rotations are norm-preserving →**(spec norm of rotation matrix =1)**

Markov chain→ is a mathematical model used to describe a sequence of events where the probability of each event depends only on the current state and independent of past

▼ Errors

Relative errors won't tell about the relative errors in each component x is not equal to 0, and its approx is y

Absolute error

$$\underline{e} = \|\mathbf{x} - \mathbf{y}\| \qquad !$$

Relative error

$$\rho = \frac{\|\mathbf{x} - \mathbf{y}\| \checkmark}{\|\mathbf{x}\| \checkmark} \qquad !$$

Eigenspace - *it is the direction of eigenvector x which is not changed by transformation;*

```
- det(A-&lambda;I) =0 - this is characteristic polynomial, we can find eigenvalues
- if A is square matrix, with det(A)=0, it is singular, Ax=0 for $A \neq 0$
```

▼ Fitting

▼ Perpendicular Line fitting

```
a1=np.column_stack((np.ones(len(px)),np.zeros(len(px)),px))
a2=np.column_stack((np.zeros(len(qx)),np.ones(len(qx)),qy))
b1=py[np.newaxis].T
b2=-qx[np.newaxis].T

y1=slope*p[:,0] + c1
y2=(-1/slope)*q[:,0] + c2
```

▼ Circle

```
A1=np.column_stack((px,py,np.ones(len(px)),np.zeros(len(px))))
A2=np.column_stack((qx,qy,np.zeros(len(qx)),np.ones(len(qx))))
B1=(px**2 + py**2)[np.newaxis].T
B2=(qx**2 + qy**2)[np.newaxis].T
def plot_circle(cx, cy, r):
    theta = np.linspace(0., 2 * np.pi, 100)
    x = cx + r*np.cos(theta)
    y = cy + r*np.sin(theta)
    # plot circle
    plt.plot(x, y, color='y', linewidth=2)
    # plot center
    plt.plot(cx, cy, 'o', color='g')
a = X[0] / 2.
b = X[1] / 2.
r1 = np.sqrt(X[2] + a**2 + b**2)
r2 = np.sqrt(X[3] + a**2 + b**2)
plot_circle(a, b, r1)
plot_circle(a, b, r2)
```

▼ PCA

is a dimensionality reduction technique, high dimensional data can be converted to lower dim data retaining the original variation as much as possible.

- it finds a set of variables, known as principal components, essentially which are linear combinations of the original variables, accounting for highest variance in the data
- Then first principal component is the direction having highest variance, second pc is the one having 2nd highest variance etc
- it is performed using eigenvalues and eigenvectors of covariance matrix of the data

- Allows data transformation into a new coordinate systems based on these eigenvectors, enabling dimensionality reduction, data visualization

Transformation to same dim → less variance

Benefits of transform to less dim →

- Improved EDA, Computational efficiency increases
- *Helpful when working with high dim data, where no of features > no of samples*

Choosing proper dim →

- Check most variance explained by each principal comp, like elbow plot/scree plot
- Observe a point of inflection/elbow, indicating where variance starts to settle down
- Ideal amt of components to keep → no of principal comp before the elbow
- One can plot a cumulative explained variance plot.. like line graph

▼ SVD

SVD is matrix factorization technique which breaks down a matrix A into 3 matrices U, Σ, and V^t.   U and V are both orthogonal matrices.The columns of V are the eigenvectors of A^T A; the columns of U are the eigenvectors of AA^T .

`A = U Σ V^T`

U - columns are left singular vectors

V - columns are right singular vectors

Σ - diagonal matrix containing singular values of original matrix

▼ Singular values

- *equal to **square root of eigenvalues of cov matrix***
- *non-negative, indicate importance of corresponding singular vec*
- *Sum of squares of sing val is total variance*
- **rank estimation**, *no of non-zero sing values corresponds to the rank of matrix, to find redundancy*

- **Dimensionality reduction**- *in SVD we select a subset of the largest singular values and their corresponding sing vec, this allows to represent data in lower dimensional space retaining the imp info.*

▼ Covariance matrix

- *Square matrix that summarizes the pairwise covariances between different variables or features in a dataset*
- *Provides info about linear dependencies and relationships between variables.*
- *Covariance btw 2 variables indicate how they vary together and whether they tend to increase or decrease together or if they are independent*

The Principal components of data is obtained by performing SVD on centered data matrix

If X is a data matrix, with n rows (samples) and p columns (variables)

- with each column as centered (mean=0), first K principal components would be first K columns of U in the SVD of X.
- In case of singular matrix, one of the singular values must be zero
- SVD allows to solve linear least square problems

```
1. Finding covariance matrix
2. We then look for the eigenvalues and eigenvectors of $\Sigma$
3. The eigenvalues are sorted in decreasing order and the $k$ largest ones are taken
4. The eigenvectors corresponding to the $k$ largest eigenvalues are put in a matrix $W$
5. We finally project the original data onto the $k$ dimensional subspace: $Y = W^TX$
```

▼ RANSAC

Estimating parameters using inliers and ignoring outliers, used as an iterative estimation algorithm to estimate model parameters from noisy data by selecting random subsets of data points.

- *Select random points, fit plane using SVD, obtain approx model params*
- *Compute cost/distance of these points from plane and determine inliers within threshold*

- *no of inliers > threshold, re-estimate the parameters using inliers*
- *loop until convergence and select plane with largest inliers as best*

▼ Jacobians

Jacobians are matrices of partial derivatives that *relate changes in one set of variables to changes in another set of variables.* In solving non-linear equations, they help **linearize** the problem, **represents the sensitivity or responsiveness of the system to changes in the input variables.** Jacobian-

- **Intrinsic Jacobians f**or manipulators calculate the linear and angular velocities of the end-effector based on the joint velocities of a robot.They are used in forward and inverse kinematics problems.
- Amplifiers of errors in a system, Small errors in the input variables(joint angles) can be amplified in the output variables(end-effector position or vel).
- accelerates iterative convergence of Newton zero finding algo, itapproximates the solution by iteratively updating the variables.

Three applications of Jacobians.
Solving fixed point problems, mapping joint velocities to end-effector vel in Cartesian space, differential error analysis
c) The **Banach Fixed Point Theorem** guarantees the existence and uniqueness of solutions in a fixed-point iteration process.
f) Overdetermined sets of non-linear equations can be solved by combining Newton's method with Gaussian error minimization
g) The **Gauss-Newton method** is an optimization algorithm that iteratively minimizes the sum of squared errors between predicted and observed values, used for non-linear least squares problems, utilizes a linear approximation of the model function by computing the Jacobian matrix

▼ ODE

- **General solution**→set of all possible solutions to the differential equation.
  A **particular solution** is a solution to a differential equation that satisfies specific initial or boundary conditions, obtained by assigning specific values to the parameters in the general solution

- **Initial Value Problem (IVP)** →where **initial values of the dependent variable** and its derivatives are specified at a particular point in the domain. The goal is to find a particular solution that satisfies the differential equation and the given initial conditions. They differentiate a particular solution from general solution
- **Boundary Value Problem (BVP)** →values of the dependent variable are specified at the **boundaries of the domain**. The goal is to find a particular solution that satisfies the differential equation and the given boundary conditions.

▼ ODE - defined by linearity, coeff and boundary condition

▼ types

F(x,y,y') — y dependent and x independent

Non-linear - where dependent variables and derivatives are non-linear, so the relationship between dependent variables and derivatives aren't proportional or additive

Linear ODE - where dependent variables and derivatives are linear, so the relationship between dependent variables and derivatives are proportional or additive, can be written as linear combination of dependent variables, derivatives and any independent variables with constant coeff

Explicit ODE- dependent variable and its derivatives are expressed explicitly in terms of independent variable and dependent variable, like derivatives are on one side, and rest variables on other side

Implicit ODE —not explicit—-, usually involve terms where derivatives appear in non-isolated manner, such as square root or inside a trigo func

1st order ODE- involves first derivative of dependent variable wrt independent variable

2nd order ODE- —second derivative—-

Constant Coefficients Systems - where coeff multiplying the derivatives are constants,

Coupled ODEs - system of equations which are interdependent, behaviour of one variable is influenced by behaviour of one or more other variables

Initial value problems- in which solution is sought with specific initial conditions. You are given the values of variables or functions and their derivatives at a specific point at t0.

L =y' + g(t)y - $linear$**, first order**

- L(y) is linear, contains only first order derivatives
- non-homogeneous if L(y) = h(t) , for homogeneous L(y) =0

▼ Solving ODEs

Separation of variables

Analytically by Ansatz

Variation of params- incase of non-homogeneous

By numerical calculation

Using sympy or MATLAB

Laplace

Integrate real-valued function of 1 real variable

- **Antiderivative** - reverse of differentiation. F'(x) = f(x), F(x) is anti-derivative whose derivative is f(x)
- **Indefinite integral** - family of anti-derivatives, when evaluating an indefinite integral, we find a function whose derivative is equal to integrand

```
Integration by parts:
∫ u dv = uv - ∫ v du
Integration by substitution:
u = g(x)
```

▼ Probability

```
Random variable— is defined as a function that assigns a
numerical value to each outcome in the sample space of a random experiment
```

CDF → The Cumulative Distribution Function, denoted as F(x), of a random variable X is a function that gives the probability that X takes on a value less than or equal to x.

F(x) = P(X ≤ x) ———— $ E[X] = \sum_{x}{p_n} $ eg- line increasing linearly from 0 to 1

PDF → Probability Density Function (PDF) of a continuous random variable X is a function that describes the probability of the random variable taking on a specific value within a certain range. ex- gaussian distribution (normal)

Expected value of random variable X →*average value we would observe after many iterations of the experiment, provides a measure of central tendency for the random variable.*

Empirical Mean → defined as the diff to all elements, is also minimal for the sum of squared diff

▼ Bayes theorem

- update probab based on new evidence, combining prior knowledge and observ to obtain posterior probab - `$posterior = \frac {likelihood*prior}{evidence}$`
- Accumulating knowledge refers to *updating probab* as new evidence is acquired…. **Priori knowledge** →initial beliefs or probab before evidence
- Reversal of cause and effect refers to inferring the cause given the effect, Bayes allows us to reverse conditional probab
- Independence(no 2 events are related) and conditional independence(2 events independent given knowledge of third)

  ```
   Independent Events: $P(A \cap B) = P(A) \cdot P(B)$
  Conditional Independence:$P(A \cap B | C) = P(A | C) \cdot P(B | C)$
  ```

- **Bayesian decision theory** involves making decisions based on probabilities and expected utilities

▼ Bayes Filtering

Bayes filtering is motivated by the need to estimate the state of a dynamic system based on noisy measurements. It provides a probabilistic framework for combining `prior knowledge, sensor models, and probabilistic inference to improve the state estimation.`

Probability measure defines the likelihood of events occurring in a sample space. Probability algebra involves operations such as union, intersection, and complement of events

Join probability describes the probab of events occurring together, **conditional probability** represents probab of event given another has occurred

▼ Eigenfaces

For group of imgM, construct an N no of eigenfaces, such as N < M, images can be repr as linear comb of set of eigenfaces

- *we calculate cov of A^t which is MxM,*
- *Recognizing- calculating a vector of weights, w=U^t, then we calculate dist btw projection of face and projection of faces in the db*
- *error= w-wk, error is min the face is related to that of corresp k face*