Animation & Robotics Summary

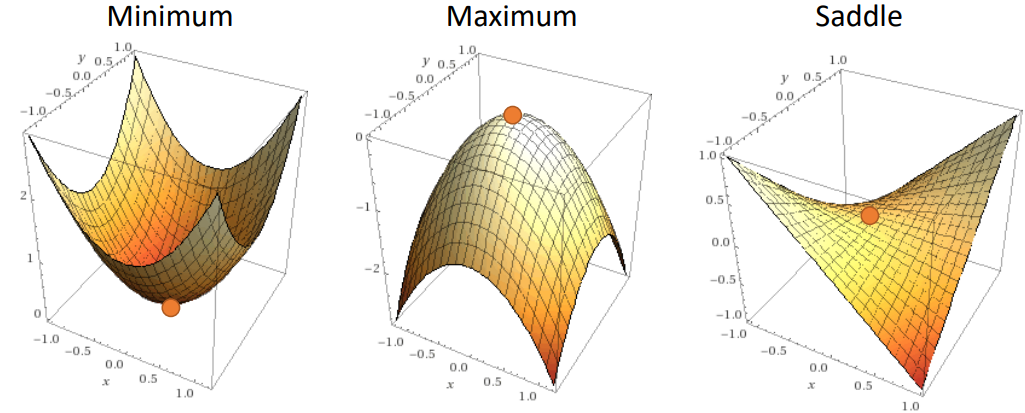
Lecture 01 – Introduction To Optimization

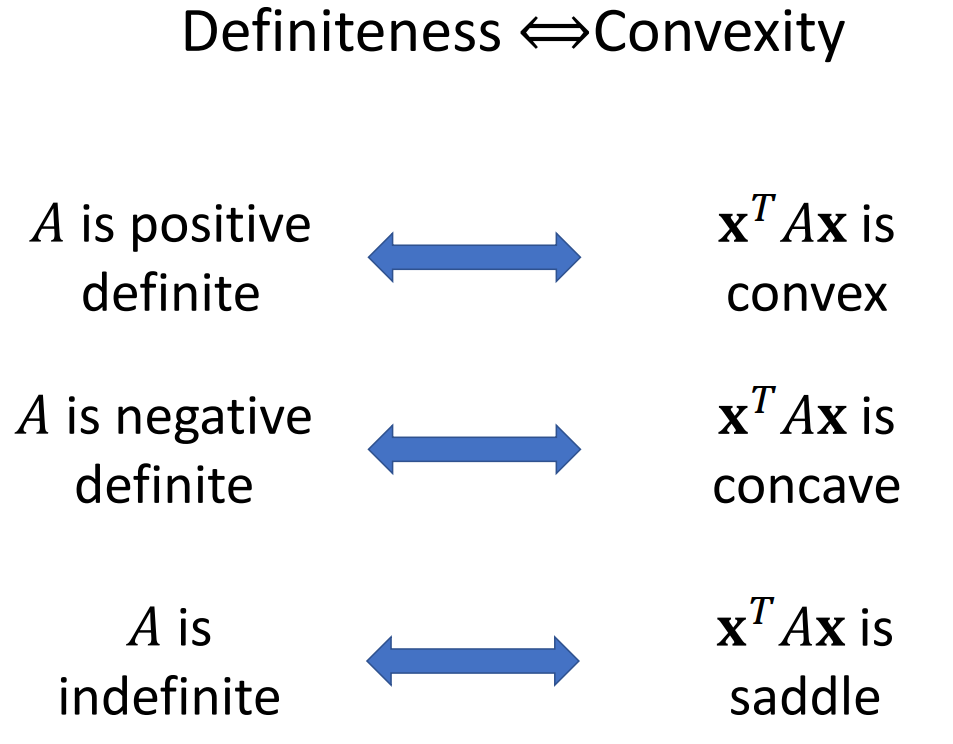
Optimization Problems:

Let’s seek on problem. Finding .

In 1D, making the points that fulfill this to be נקודות קיצון where .

In nD, .





A quadratric form is a function of the form :



NOTE :

1. Gradient Descent

Objective : Gradient is direction of steepest ascent. Function value gets largest if we move in direction of gradient, it doesn’t change if we move orthogonally but it decreases fastest if we move exactly in opposite direction

A graph of a function

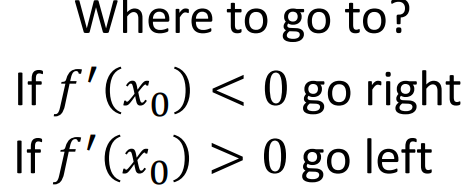
Description automatically generated

A diagram of a mathematical equation

Description automatically generated

1. Line Search

Objective : Given and a step direction , determine step size such that :



A white rectangular sign with black text

Description automatically generated

1. Newton Step

Objective : Start with Taylor expansion, then We want the minimum, so we take the gradient and look for 0.

NOTES:

1. In general, Newton’s method requires many less iterations to converge.
2. Newton’s method only work if the Hessian is positivedefinite.
3. In general, Newton’s method is preferable.

Lecture 02 – Soft Body Simulation

***Definition*** *: Elasticity The ability of a material to resist a deforming force and to return to its original size and shape when that force is removed.*

* Hooke’s Law :
* Spring Energy :

Equilibrium : A physical system will always reach a state which no further change is possible.  
Newton Law on Equilibrium : and the energy reaches a minimum state : whereas present a spring between vertices .

Spring System :

Gradient/Hessian Matrix Calculation :

*Definition: Compressed Column Storage (CCS) ,a sparse matrix or sparse array is a matrix in which most of the elements are zero.*

Lecture 03 – Deformations

There are three common Metaphors : Point, Skeleton and Cage.

Map 2D representation : ; for example, .  
The problem with this representation is that setting maps is big too handle. Therefore, representing maps as linear combination of basic functions.

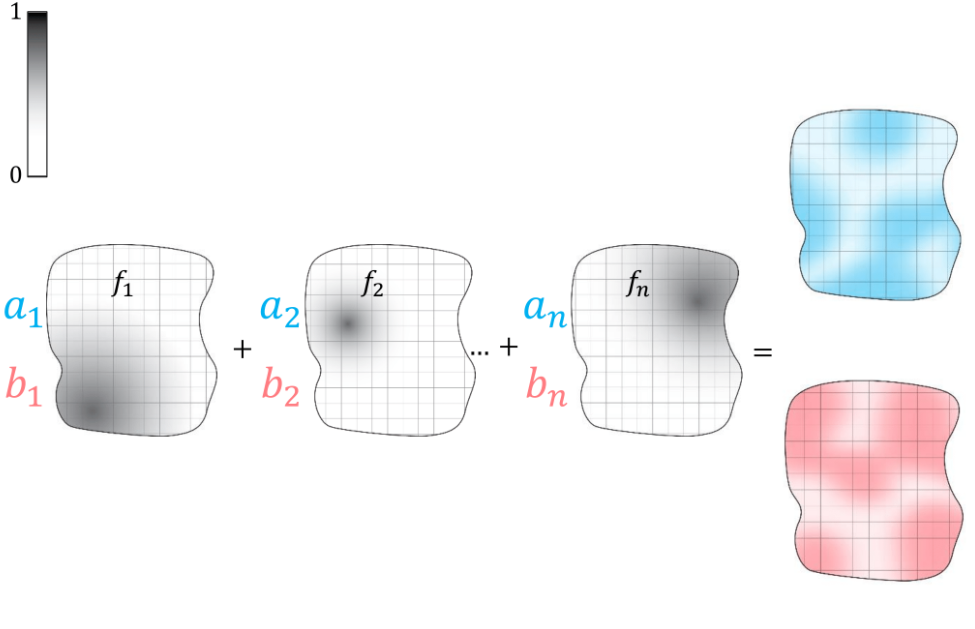


Figure : linear combination of basic functions

It can also be presented as .

Locally Bijective versus Globally Bijective;

***Definition*** *: A function is said to be bijective or bijection, if a function satisfies both the injective (one-to-one function) and surjective function (onto function) properties.*

**Question:** if a present a linear combination of 2d mapping considered to be locally bijective (Each interior function is bijective) would the function consider to be globally bijective? Vice versa?

Answer : No, if you are interested “Global Inversion Theorems” even though there are some cases that fulfill that. But the other direction is yes(!).

A close-up of a dart board

Description automatically generated

Figure : This Mapping cosidered to be not globally bijective but locally bijective.

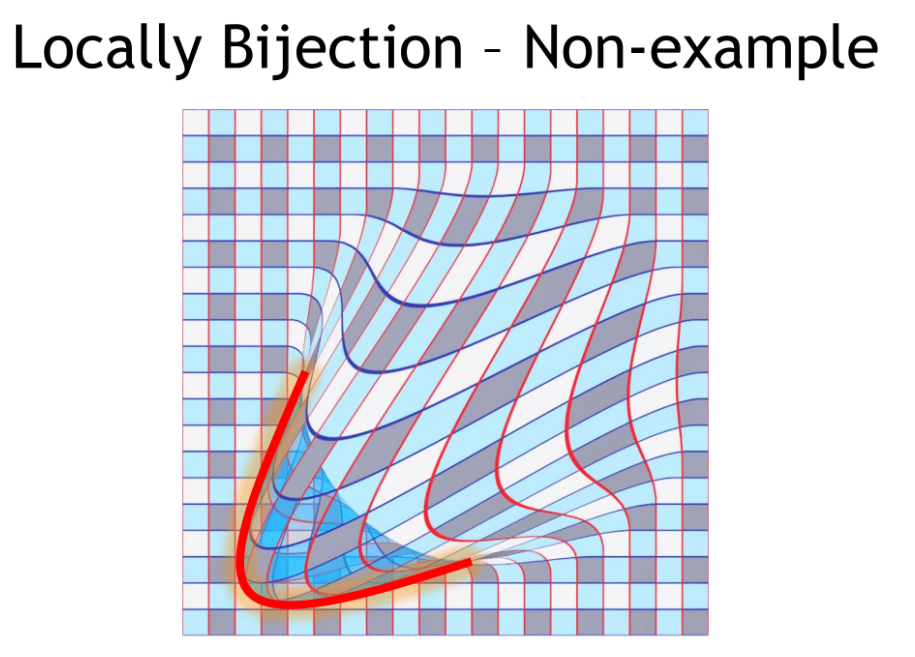


Figure : Not Locally Bijective.

**Condition for Locally Bijective;** suppose the function of a deformation is . The Jacobian Matrix would be . The function is locally bijective if

Good Mapping : an example,



Figure : in the second map is not bijective due to data loss

***Definition*** *: Distortion is a function of the Jacobian at a point. There two types of distortion; Conformal distortion and Isometric distortion.*

Visually : The lengths in Conformal distortion are changed and the angles of the square/rectangle remained 90o.

Visually : The lengths in Isometric distortion are kept and the angles of the square/rectangle has shifted to parallelogram.



Figure : Conformal distortion (on left) versus Isometric distortion (on right)

NOTE : represents a deformation map.

* LSCM (Least Squares Conformal Map)

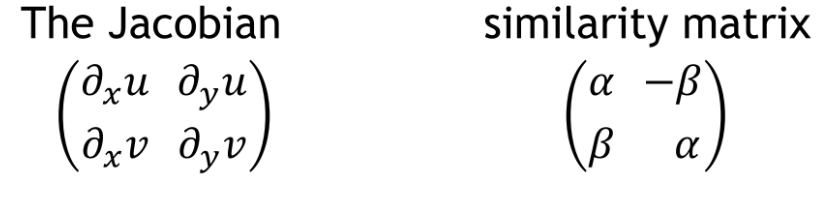


Figure : Jacobian Vs Similarity Matrices

Cauchy Riemann Equations

* ASAP (As Similar As Possible) :
* ARAP (As Rigid As Possible) :

Distortion On Triangle Meshes:

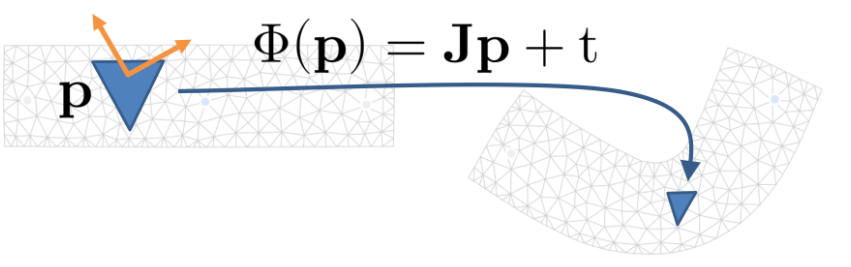


Figure : Distortion On Triangle Meshes

* Conformal (LCSM) :
* Isometric (ARAP) :
* Authalic :

Lecture 04 – Transformation:

1. Scaling : Let a vertex, Scaled vertex would be like s.t . A case when called uniform/isotropic scaling(preserve 1:1 ratio).
2. Rotation : Let a vertex, Rotated vertex would be like s.t whereas matrix representation be like
3. Translation : Let a vertex, Translated vertex would be like s.t whereas matrix representation requires shifting to Homogeneous Coordination by added a z-value for . Therefore, Matrix transformation would shape like:
4. Shear : (

NOTE : Matrices Order is IMPORTANT!

Singular Value Decomposition (SVD): Every Matrix M has factorization of the form .

Lecture 05 – Rigid Motion:

1. Translation : upon the mesh located at point such that . At mesh is transferred to point which located at .

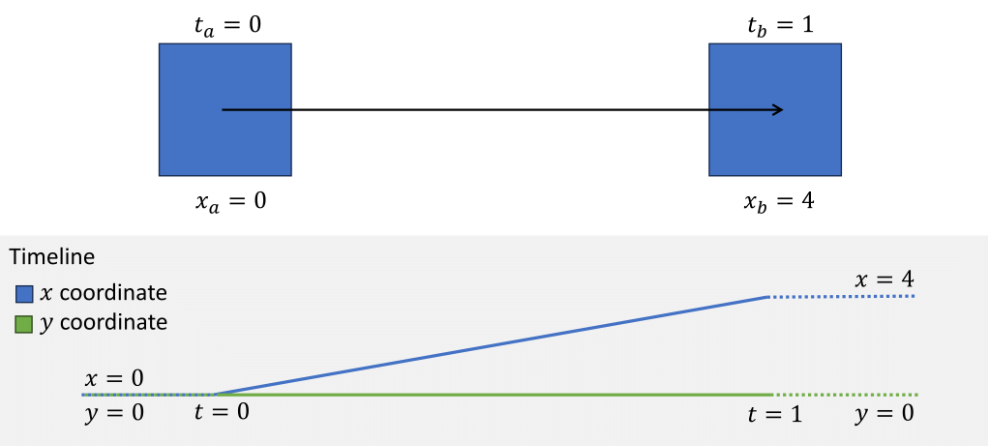


Figure : Translation from A To B in timeline

* At Translate Matrix representation :
* At : Matrix would reach ,
* At Matrix representation :

1. Rotation : upon the mesh located at point had . Then the mesh is rotated to with .

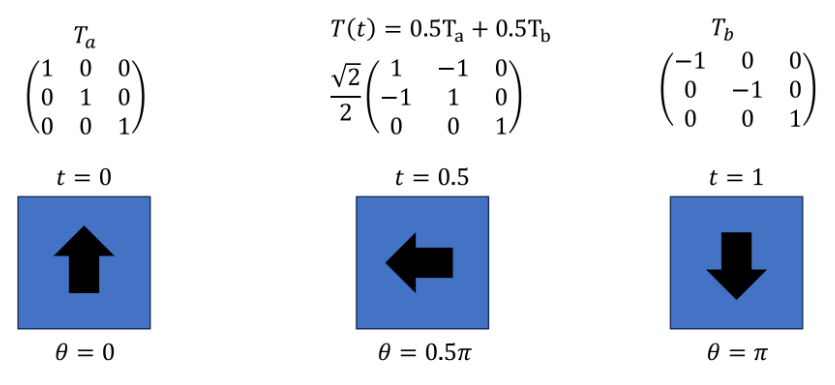


Figure : Rotation timeline

As we notice in Translation, representation as vector and interpolation as lines.

What about Orientation? Representation as angles and interpolation as ..??

Leaving 6 equations, 9 unknowns and 3 numbers. Therefore, 3 angles.

1. .
2. .
3. .

Note that the order is IMPORTANT but ARBITRARY.

*Definition : Extrinsic or intrinsic with respect to world coordinates or local coordinates. Extrinsic rotations about the axes xyz of the original coordinate system, which is assumed to remain motionless. Intrinsic rotations about the axes of the rotating coordinate system XYZ, solidary with the moving body, which changes its orientation with respect to the extrinsic frame after each elemental rotation.*

* Gimbal Lock :

Upon , resulting in loss of one degree of freedom .

**Theorem:** (Euler): Any orientation can be obtained from a fixed reference orientation by a single

Unique rotation around an appropriate axis in space.

Any orientation may be described by four parameters: angle and unit axis .

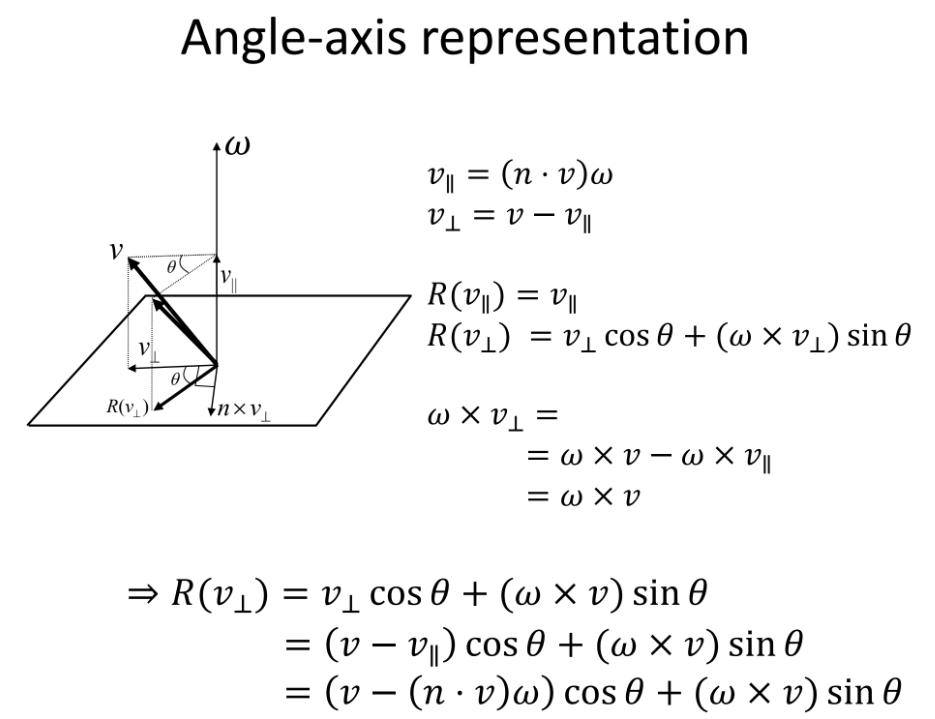


Figure : Angle-Axis representation

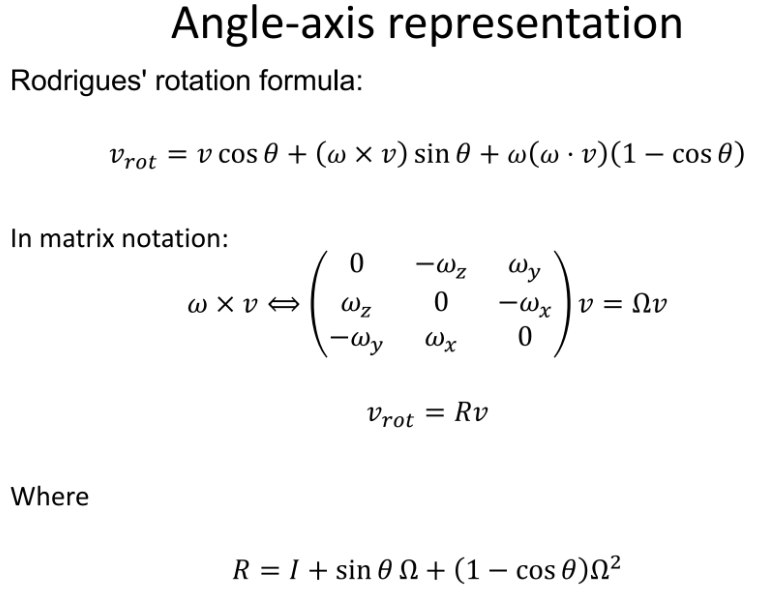
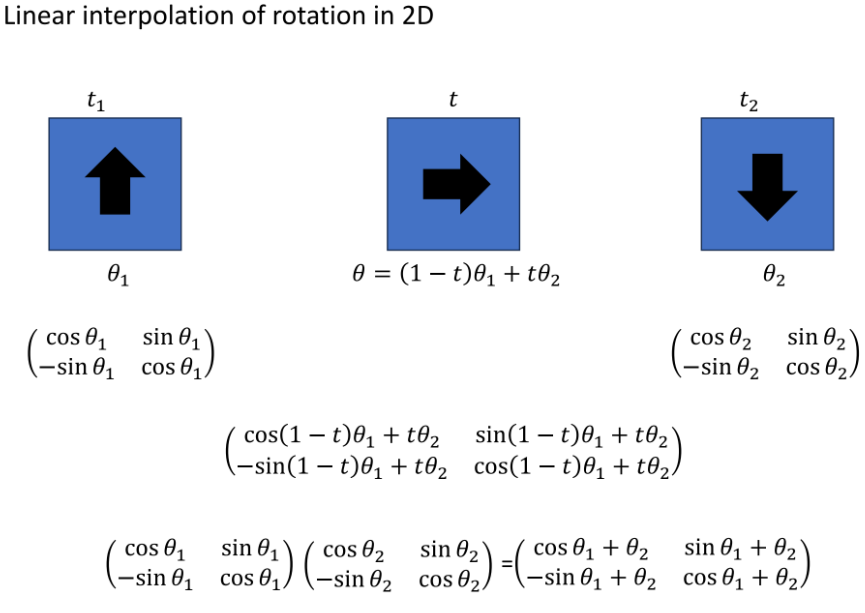


Figure : Angle-Axis representation via Rodrigues formula

So, Back to Rotation Interpolation :



Complex Numbers:

Multiplication :

Therefore,

Linear Interpolation of and :

Quaternions – ATTEMPT OF EXTENDED 3D COMPLEX NUMBER

Definition : .

* Addition Operation
* Multiplication Operation
* Conjugate Operation
* Size

Rotation : Rotation by around unit direction may be represented by the unit quaternion.

Example : Therefore, The 3D Vector is rotated by to .

The shortest rotation Given rotation and what is the “shortest” path between them?

For any two quaternions and the SLERP between them is :

Which is equivalent to :

Lecture 06 – Kinematics:

A black and blue symbol

Description automatically generated with medium confidence

* One Link : Local frame aligns with parent joint axis.
* Two Link :

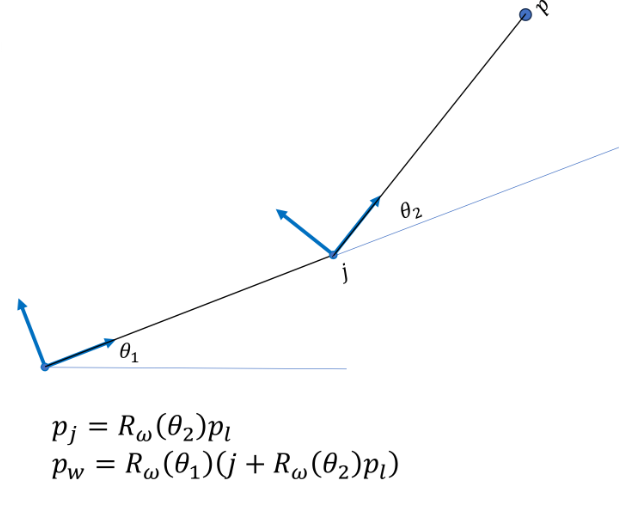


Figure : Two Links

* Three Links :
* Four Links :

Velocity Jacobian: How does a small change in joint angles change the position of ?

* One Link :
* Two Link :
* More than two:

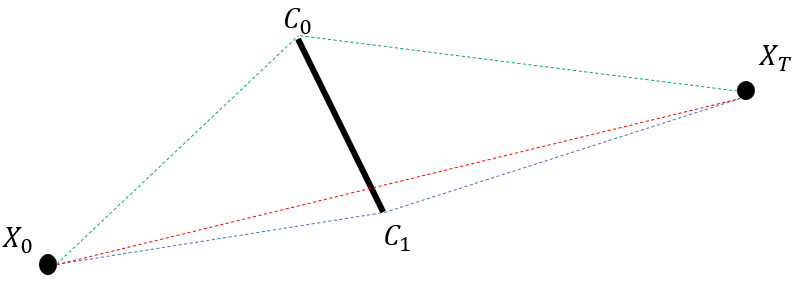
Forward kinematics on three links:

A black and white image of a long line

Description automatically generated

**Inverse Kinematics : I will have to watch the lecture again to write it down.**

Lecture 07 – Path Planning:



* Analytical method:

F(x) =

* Numerical method :

1. **Search-based algorithms (Grid-based algorithms) :**
   * + BFS(Time-Complexity ) /Dijkstra(Time-Complexity )
     + matrix where we define as the following:

same as Dijkstra. The real cost to reach a node n.

approximate cost from node n to goal node (heuristic function).

Dijkstra is a special case for (when the heuristi0cs is zero).

1. **Sampling-based algorithms (graph-based algorithms)**
   * + Random Tree(RT) : Doesn’t Explore well.
     + Rapidly Random Tree (RRT) : Rapid Exploring but it is not OPTIMAL
     + Rapidly Random Tree \* (RRT\*) : is the same as RRT, but two key additions to the algorithm result in significantly different results :
2. Cost Optimization: It records the travel cost from each vertex to its parent and selects the cheapest neighboring node within a fixed radius for connection, eliminating the cubic structure.
3. Tree Rewiring: It rewires neighbors to a newly added vertex if this reduces their travel cost, resulting in a smoother path.

Despite that it gives result closer to optimal solution, it takes High time complexity execution.

* + - Probabilistic Roadmaps (PRM) : Builds the graph only once and find multiple paths, Not optimal and Clusters.

Lecture 08 – Trajectory Optimization: