Preface – Zooming out

[*Optimization*](https://towardsdatascience.com/optimal-control-a-preamble-6c0bc2b12ed6)

What does a dog, an Argentinian soccer player, an economist all have in common? They try to optimize.

* Dog catching a frisbee on the beach (optimizes distance travel in water vs. land)
* Messi running from half-field to the goal (optimizes his path to avoid defenders)
* Economist calculating marginal profit (optimize risk-reward)

Optimization programs are defined by **constraints**

Optimization - Mathematics

Functions pair given inputs with a given outputs

* + If I run down field at 45 degree angle, then I will run into defender
  + If take main street to school; then I will get there at 9:10 and be late

Functions are like a big conditional

Optimization asks, “What is the output for a given range of inputs?”

Necessary Conditions:

* + (Gradient vector must be 0)
  + (Hessian matrix must be semi-definite positive)
  + Recall: Hessian = Square matrix of Second-order partial derivatives
    - First-order – rate of change of f with respect to (x,y)
    - Second-order – rate change of f with respect to (x,y) with respect to (x,y)
      * Visualize gradient vector field and break apart into components

*Constraints*

* Equality / Inequality constraints
* Equality 🡪 finding minimum of lower dimension function (more complex)
  + Solved using Lagrange Multipliers
* Inequality
  + Solved using KKT equations

*Path optimization*

* Google maps (find path from A to B that minimizes time)
* Finding an input that minimizes function 🡪 finding a function that minimizes a cost (functional space)
* **Brachistochrone** – What curve would get make a bead rolling from point A to point B get there the fastest? 🡪 Founding problem of **calculus of variations** 
  + Break apart each path into small arc lengths
  + Approximate time it takes to travel down each of the arc lengths and sum them up
  + Use calculus to integrate
  + How small variations in path affect cost of taking that path

*Control Theory*

Let’s say we have some projectile (cannonball, satellite), and we want it to follow specific trajectory

* Add **unit** to projectile system that allows us to correct its motion
  + Input = Commands to system
  + Output = Changes in the motion (state) of the object
* Controller = Creates and communicates adequate *input commands* to the unit so that the system reaches its desired states
  + Ex. Remote control car
    - System = Plastic Car
    - Unit = Battery / Circuitry that generates torque in wheels
    - Controller = remote control
* **Open-loop** control:
  + x(t) 🡪 controller 🡪 system 🡪 y(t)
  + x(t) and y(t) are the desired state and output state of system respectively
* **Closed-loop** control
  + x(t) 🡪 [ e(t) 🡪 controller 🡪 system 🡪 y(t) 🡪 e(t)]
  + e(t) = error between desired state (the reference) and current state (estimated by sensors)

*Optimal Control*

Ex. We have a car moving in straight line from point A to point B.

We want to find a sequence of inputs to give to the system so that car reaches point B in minimum time.

Defining a input function u(t) that gives input at time step t

* x(t) = traces out the path of the car over time
* u(t) = gives input to the car’s control system over time

How to find function u(t) yielding path x(t) that minimizes time to travel from point A to B:

*Find min of sum of time(t) it take to travel along each segment of path x(t), and find minimum x for that*

Min(u,x) = integral of (t) with respect to x

*Add in constraint that change in state x’(t) is dependent on current state x(t) and controls u(t)*

X’(t) = f(u(t), x(t))

*To impose another cost other than time, we can define a cost function L (for Lagrangian) that we are trying to minimize. If that cost is dependent on controls (rate of change in robotic arm) and or position (pose of joints carrying weights), we can write*:

Min(u, x) = integral of L(t, x(t), u(t)) dx

Constraint: X’(t) = f(u(t), x(t))

*This general formulation of the optimal control problem can be solved in time-efficient manner using dynamic programming.*

Text, letter

Description automatically generated

Breaking this apart:

**Law (Rule) of Evolution** for dynamical systems **–** Change in the state is dependent previous state