Trajectory optimisation

What is trajectory optimisation ?

For a dynamical system

$$\underbrace{\dot{x}(t)}_{\text{state derivative}} = f(t, \underbrace{x(t)}_{\text{state}}, \underbrace{u(t)}_{\text{control}}) \qquad (1)$$

Find a trajectory $x(t)$ that

- respects the dynamics (1)
- satisfies boundary conditions ...

  (ex : $x(0) = 0$, $x(1) = 1$
  $\dot{x}(0) = 0$, $\dot{x}(1) = 0$)

- ... and other constraints :

  $g(t, x(t), u(t)) \leq 0$ path constraints

  $x^- \leq x(t) \leq x^+ \ \forall t$ path bounds on state

  $u^- \leq u(t) \leq u^+ \ \forall t$ path bounds on control

  $t^- \leq t_0 < t_F \leq t^+$ bounds on initial/final time

- is optimal in some sense ie minimises a cost function ℓ

$\Rightarrow$ this consists in finding $u(t)$ such that all of the above is true, ie the decision variables are the control and the time)

Two options to solve a T.O. problem:

- global method. Find optimal policy $u^*(x)$:
  for all possible states characterize optimal control
  - difficult to compute
  - better suited for low-dimension problems.

- Local method:
  - compute a single trajectory
  - locally optimal
  - Easier to compute   ⇐ we'll go with this
  - Open loop

- A few assumptions:
  - continuous problem + Everything is smooth and differentiable
  - we will use numerical optimisation to solve the problem

- question:
  How to write the T.O. problem as an optimisation problem?

There are many ways to do this. Today we will see one set of approaches called direct methods.

- Discretize the problem
- approximate functions
- interpolate solution

Problem transcription :

- we will discretize time into N+1 points :

$$\begin{array}{lllllll} t & \to & t_0 & \dots & t_k & \dots & t_N \\ x & \to & x_0 & \dots & x_k & \dots & x_N \\ \dot{x} & \to & \dot{x}_0 & \dots & \dot{x}_k & \dots & \dot{x}_N \\ u & \to & u_0 & \dots & u_k & \dots & u_N \end{array}$$

- we will verify system constraints and cost at these points only
- How are these points related ? we introduce collocation constraints :

ex: how is x_{k+1} defined wrt x_k ?

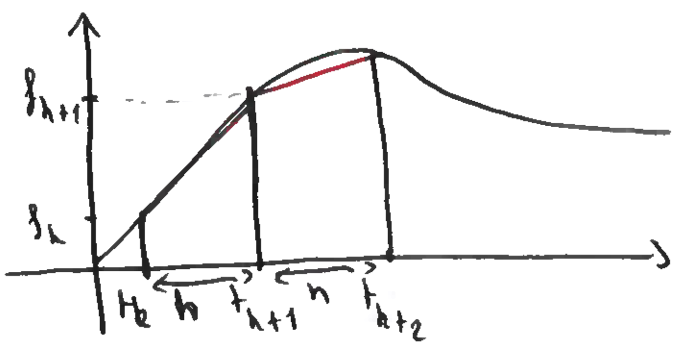
- we know that

$$\dot{x} = f$$

$$\Rightarrow \int_{t_k}^{t_{k+1}} \dot{x}\, dt = x(t_{k+1}) - x(t_k)$$

$$= x_{k+1} - x_k = \int_{t_k}^{t_{k+1}} f\, dt$$

- How to calculate $\int f$? we will use a linear approximation :



assume linear interpolation between f_k and f_{k+1}

- Integral is area under the curve :

with approximation, this is a trapezoid!

$$A = \boxed{\frac{1}{2} h (f_{k+1} + f_k) = x_{k+1} - x_k}$$

the smaller h is, the better the approximation

What about the constraints?

Only checked at discretized points (called collocation points)

e.g.:

- $x(t) < 0 \ \forall t \Rightarrow x_k < 0 \ \forall k$
- $u(t) < 0 \ \forall t \Rightarrow u_k < 0 \ \forall k$
- $g(t, x(t), u(t)) < 0 \ \forall t \Rightarrow g(t_k, x_k, u_k) < 0 \ \forall k$

What about the cost function?

Typical cost functions minimise integral of cost over time (e.g., force / acceleration)

$$\min \int_{t_0}^{t_F} \ell(t, x(t), u(t))\,dt \simeq \sum_{k=0}^{N-1} \frac{1}{2} h_k (\ell_k + \ell_{k+1})$$

trapezoidal collocation

- Our fully transcribed problem is thus written as:

minimise over:

$X = [x_0, \dots x_k \dots, x_N]$

$U = [u_0, \dots, u_N]$

t_0

t_F

$$\sum_{k=0}^{N-1} \frac{1}{2} h_k (\ell_k + \ell_{k+1})$$

s.t. $\forall k \ g(t_k, x_k, u_k) \leq 0$ ⇐ includes bound constraints

$x^- \leq x_k \leq x^+$

$u^- \leq u_k \leq u^+$

$x_{k+1} - x_k = \frac{1}{2} h_k (f_{k+1} + f_k)$

$x_0 = init\_x_0$

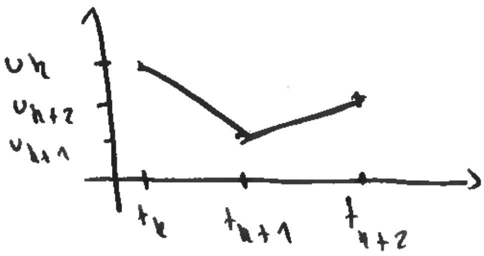
$x_N = goal\_x_N$

$t^- \leq t_0 < t_F \leq t^+$

Trajectory interpolation

- we go from the control. Once problem is solved, we know all u_k and t_k.

  $u(t)$ is a piecewise linear function (C^0)



we define $\tau = t - t_k$

$$h_k = t_{k+1} - t_k$$

$$\Rightarrow u(t) \simeq u_k + \frac{\tau}{h_k}(u_{k+1} - u_k)$$

for $t \in [t_k, t_{k+1}]$

- Dynamics is also linear (we defined this with trapezoidal)

$$f(t) = \dot{x}(t) \simeq f_k + \frac{\tau}{h_k}(f_{k+1} - f_k), \quad t \in [t_k, t_{k+1}]$$

To obtain $x(t)$ we integrate:

$$x(t) = \int \dot{x}(t)\, d\tau = c + f_k \tau + \frac{\tau^2}{2h_k}(f_{k+1} - f_k)$$

with $c = x_k$ since $x(t_k) = x(\tau = 0) = c$

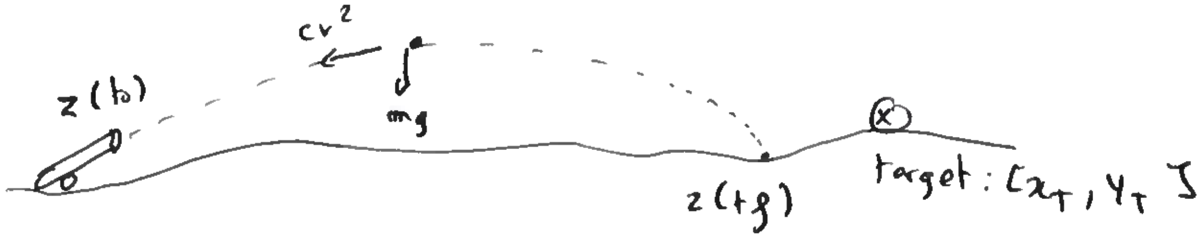
$$\Rightarrow \boxed{x(t) = x_k + f_k \tau + \frac{\tau^2}{2h_k}(f_{k+1} - f_k), \quad t \in [t_k, t_{k+1}]}$$

-

- Observations

  - There are other collocation methods than trapezoidal
  - You can use simplifications eg
    - $t_{k+1} - t_k = c \quad \forall k$ (constant)
    - sometimes solve easier problem with fixed t_0 and t_f to obtain initial guess
  - Sometimes you can use a continuous formulation.
  - In any case polynomials are often liked:
    - few parameters
    - Easy to integrate / derivate.

- Example: shooting a canon
  (from Matthew Kelly)



- find trajectory that minimises ~~target~~ amount of powder (assumed proportional to speed squared)
- canon ball is point mass
- air friction is modeled as quadratic drag

problem is thus:

find
$z(t) = [x(t), y(t), \dot{x}(t), \dot{y}(t)]$

$$\text{minimise} \quad \cancel{\#}\ \ell = \dot{x}(t_0)^2 + \dot{y}(t_0)^2 \ \} \ \text{Cost}$$

$$\text{s.t.} \quad \left.\begin{aligned} x(t_0) &= 0 \\ y(t_0) &= 0 \\ \dot{x}(t_f) &= x_T \\ y(t_f) &= y_T \end{aligned}\right\} \text{Boundary conditions}$$

$$\left.\begin{aligned} v &= \sqrt{\dot{x}^2 + \dot{y}^2} \\ \ddot{x} &= -c(\dot{x} v) \\ \ddot{y} &= -c(\dot{y} v) - g \end{aligned}\right\} \text{dynamics}$$