A fast approach to Robust Optimal Control

Dhruv Shah
Dept of Electrical Engineering, IIT Bombay

Motivation

Compute robust solutions to nonlinear optimal control problems quickly that are "good enough"

Possible Applications:

- Robotic Link Manipulators
- Control of Construction Cranes
- Physiology (walking is modelled as an inverted pendulum)

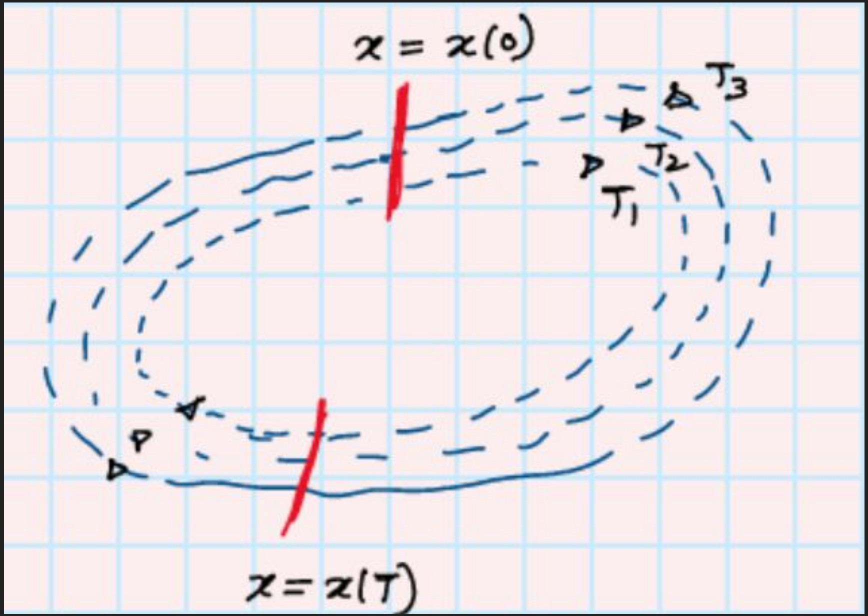
The Optimal Control Problem

$$\left.\begin{aligned} &\min_{u} \int_0^T \frac{1}{2} u^2 \, dt \\ \text{s.t.} \quad &\dot{x} = f(x,u), \quad x(0) = x_0, \quad x(T) = x^*, \\ &T \text{ fixed}, \quad u \in U \end{aligned}\right\} \rightarrow B$$

Two Point Boundary Value Problem

$$
\left.
\begin{aligned}
\dot{x} &= \frac{\partial H}{\partial \lambda}(x,\lambda) \\
\dot{\lambda} &= -\frac{\partial H}{\partial x}(x,\lambda) \\[1em]
x(0) &= x_0 \,, \quad x(T) = x^* \qquad \text{Given} \\[1em]
\lambda(0) &= \,?
\end{aligned}
\right\} \longrightarrow B^{\lambda}
$$

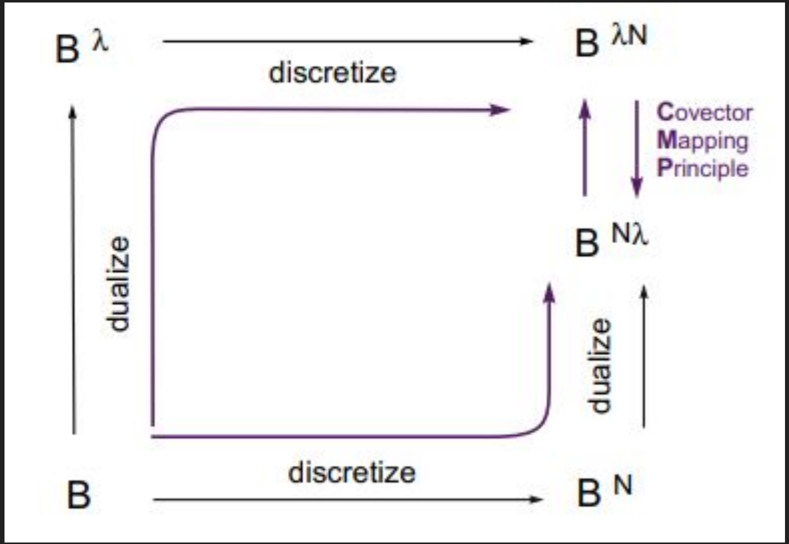
A deeper view



Conservative Hamiltonian Viewpoint

Various Approaches to solve the OCP

- Direct Methods (Solve B^N using nonlinear programming)



- Indirect methods (Solve B^λ)
  - Single Shooting (Curse of Sensitivity)
  - Multiple Shooting

Multiple Shooting Method

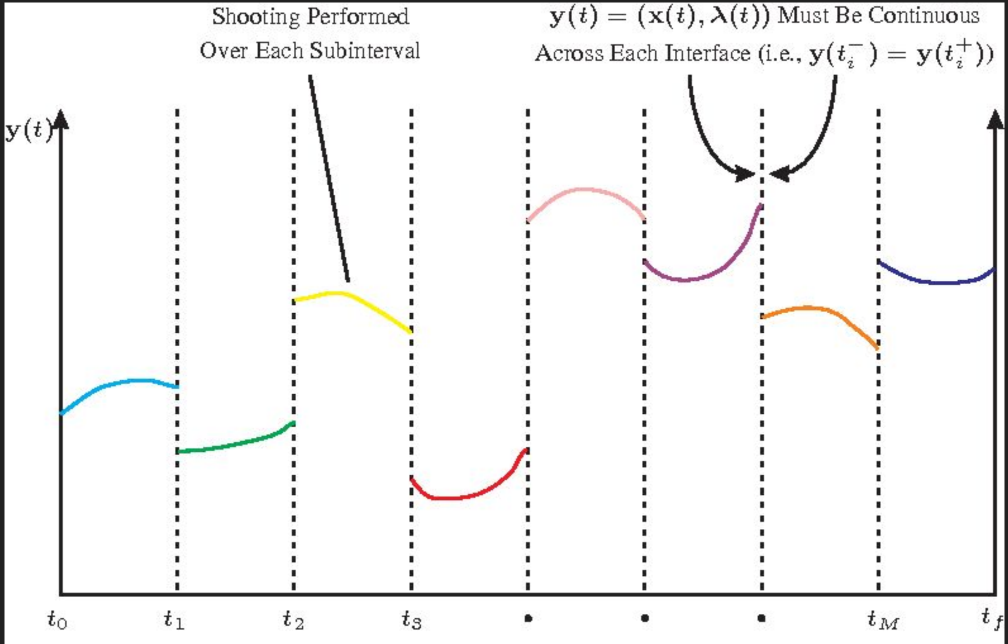


Figure 4: Schematic of the Indirect Multiple-Shooting Method.

Our Approach (Multiple Shooting)

→ Our Approach (Multiple Shooting ?)

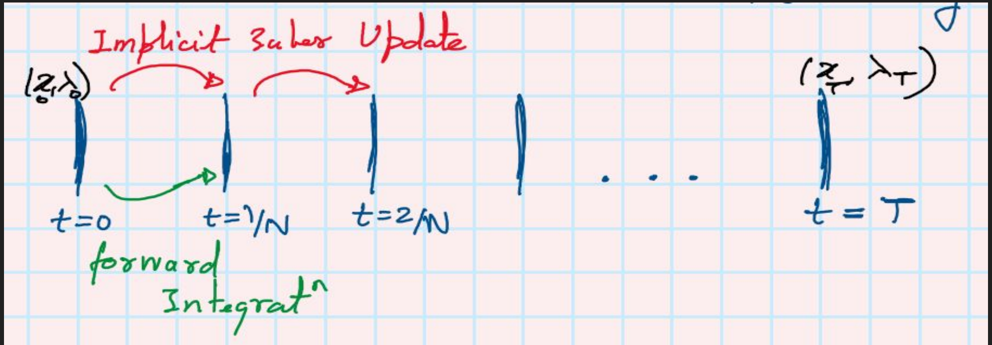
Implicit Euler Discretizatⁿ

$$\begin{pmatrix}\dot z\\ \dot\lambda\end{pmatrix} = F(z,\lambda) \longrightarrow \left[\begin{pmatrix} z\\ \lambda\end{pmatrix}_t = \begin{pmatrix} z\\ \lambda\end{pmatrix}_{t-1} + \frac{1}{2}\cdot \Delta t \cdot \left\{ F\begin{pmatrix} z\\ \lambda\end{pmatrix}_t + F\begin{pmatrix} z\\ \lambda\end{pmatrix}_{t-1} \right\}\right]$$

$$t = \underbrace{\left\{0, \tfrac{1}{N}, \dots, T\right\}}_{N \text{ steps}}$$

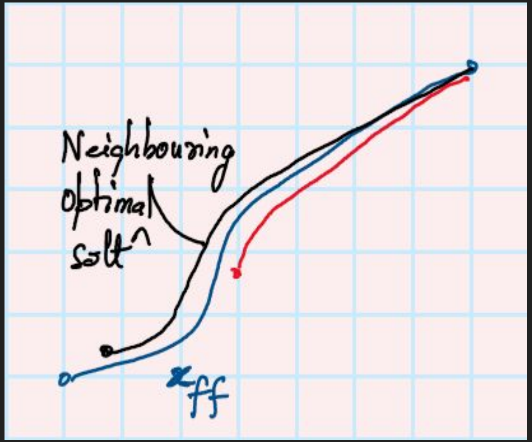
z_0 & z_T fixed

Solve using Non linear Programming



Adding Robustness using feedback

Q: Given an optimal solution (xff,uff) , can we compute all the neighbouring optimal solutions in a cheap way?



Thus, the question boils down to answering

Given, $\dot{x} = f(x,u)$ & $u_{ff} = \arg\min J$ for $x(0) = x_0$

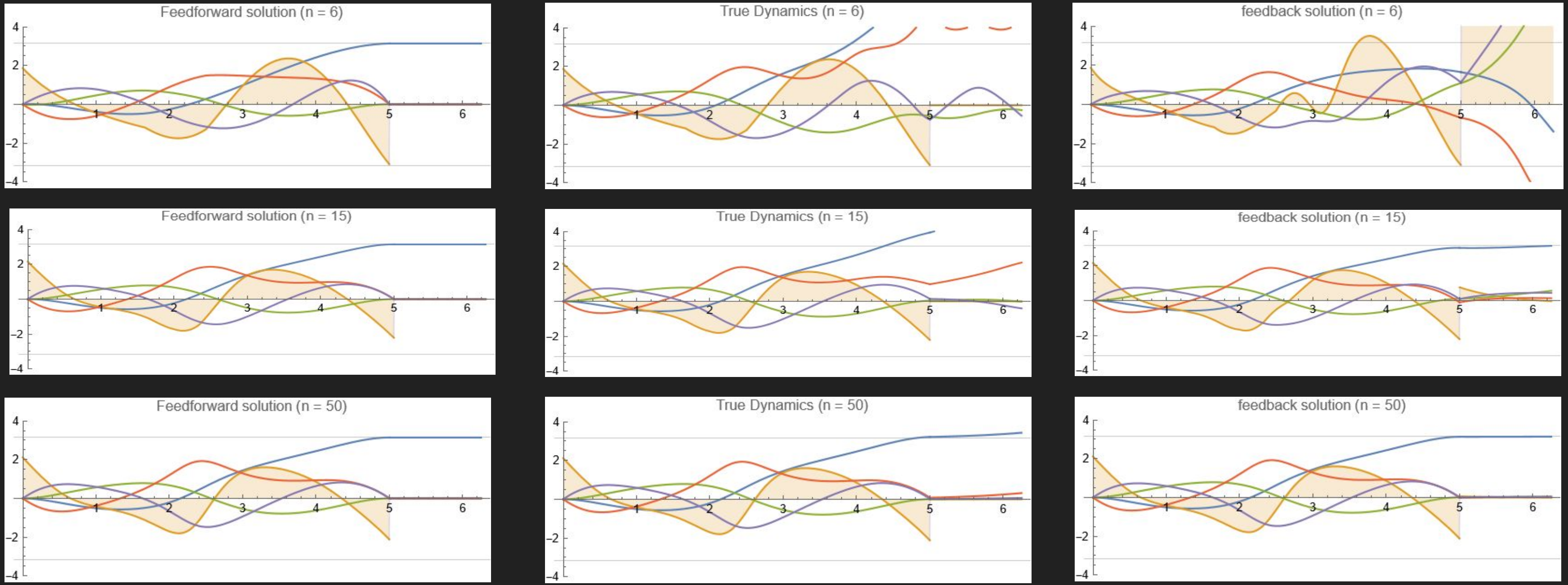
Now, if x_0 is perturbed by some δx_0, what is the signal $u_{ff} \longrightarrow u_{ff} + \delta u_{ff}$

$$\delta u_{ff} = K(t) \cdot \left(x(t) - x_{ff}(t) \right)$$

The Key Idea

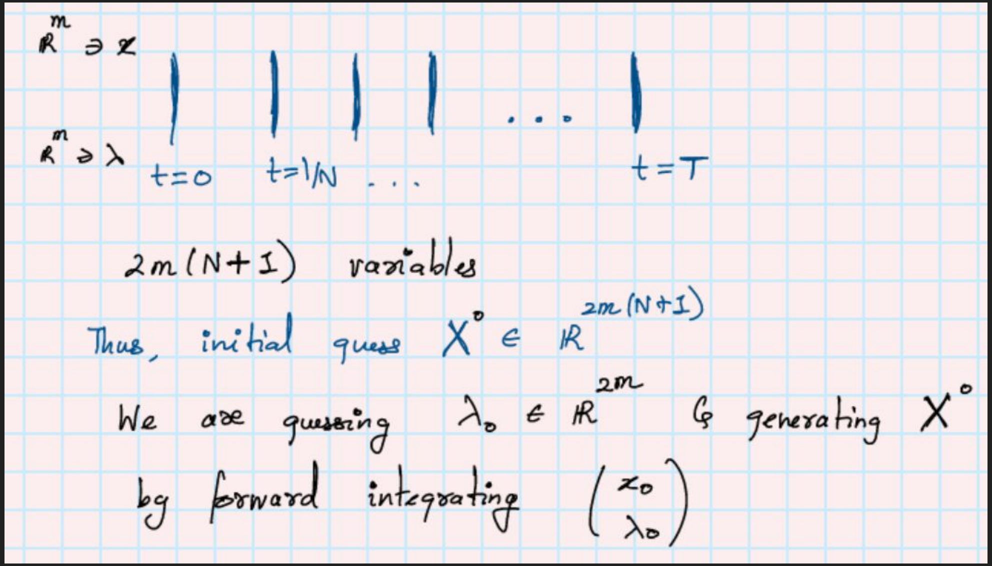
Choose N small enough so that the solution after adding feedback is “good enough”

Trade off : Optimality vs Computation Speed (0.26s,0.45s,1.26s)

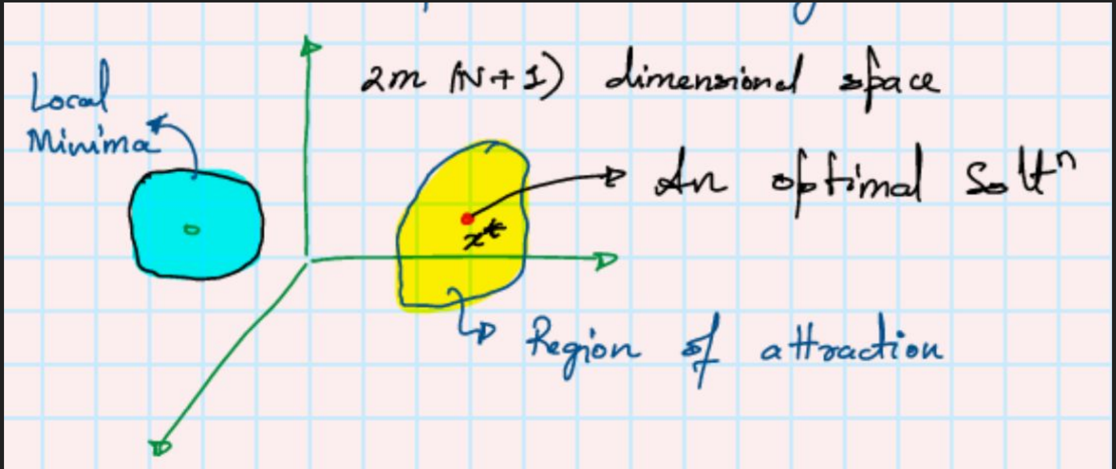


Initial Guessing Mechanism

As the nonlinear programming routines usually find local solutions, they require an initial guess



A Deeper View

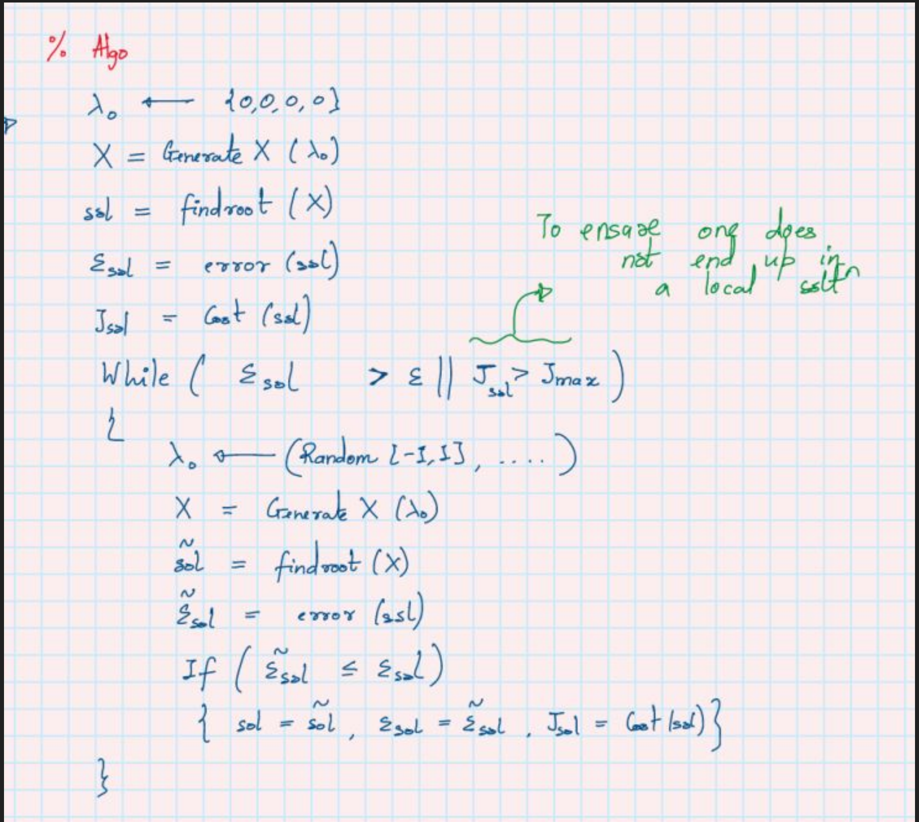


The entire landscape changes as N changes.

We expect, the region of attraction to widen as N increases

The Feedforward algorithm

We also add a count to limit the number of executions of the loop.



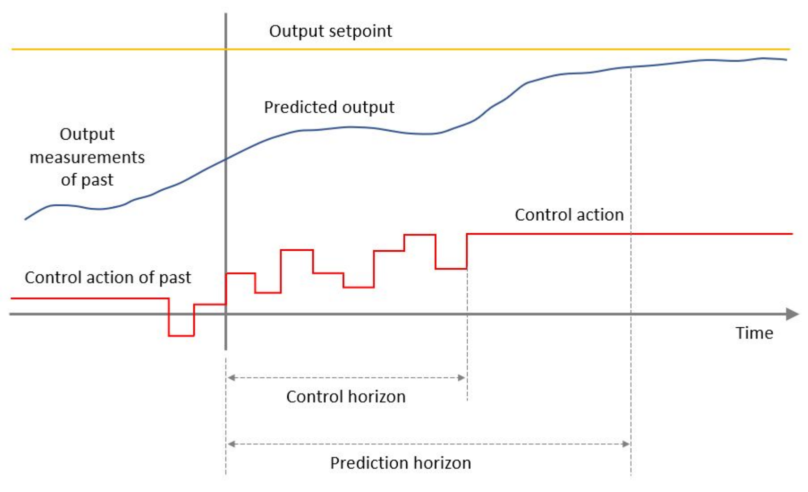
Choice of Time Horizon (T)

For the cart pendulum system we expect that control effort required reduces as T increases. However as T increases the problem becomes computationally harder.

- Advantage of quick computations is lost when T is kept variable
- If one does not care about the time of the maneuver/control task (as long as it is not too large), then the following MPC style algorithm is useful.

(MPC - Model Predictive Control)

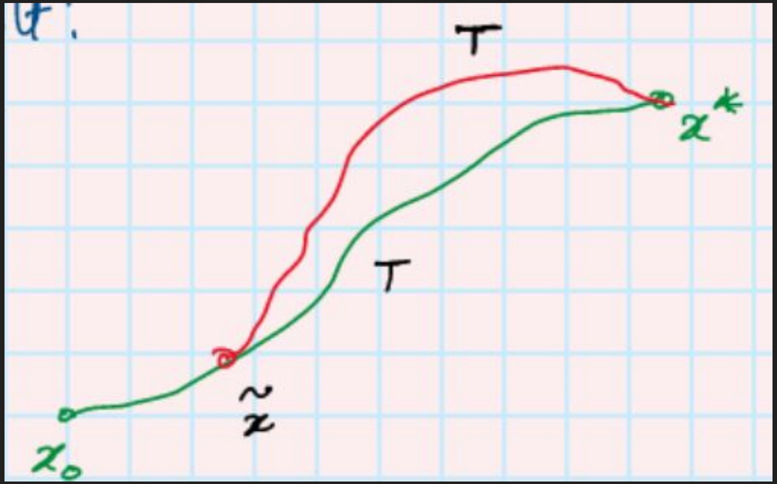
PMP based MPC



- What is MPC?
- Usually people use direct methods to solve the OCPs in MPC
- Here at every time step we use our algorithm to compute the feedforward solution and the corresponding feedback law

A few remarks

- One obtains robustness due to periodic recomputations
- Choosing a receding horizon (T decreasing at every recomputation) makes the problem harder to solve computationally
- If T is kept fixed, then it is not obvious that the solution converges



A Claim to justify the convergence of the PMP based MPC

Claim: Consider the OCP:

$$\min_u J = \int_0^T \frac{1}{2} u^2 \, dt \quad \text{——— } (*)$$

s.t. $\dot{x} = f(x, u)$; $x(0) = x_0$, $x(T) = x^*$, T fixed

where x^* is a equilibrium pt i.e. $f(x^*, 0) = 0$

Let (x_{ff}, u_{ff}) be an optimal sol^n to $(*)$ & Let J_1 be the cost associated

Now for some $0 < \tau < T$, consider the OCP:

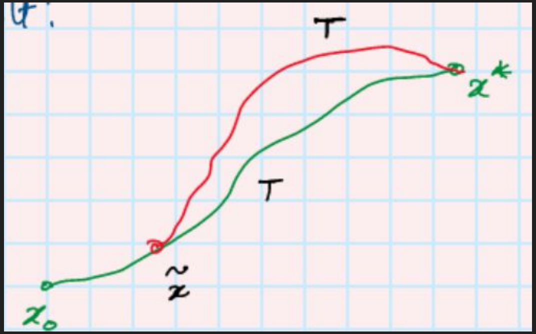
$$\min_u J = \int_0^T \frac{1}{2} u^2 \, dt \quad \text{——— } (**)$$

s.t. $\dot{x} = f(x, u)$; $x(0) = x_{ff}(\tau)$, $x(T) = x^*$, T fixed

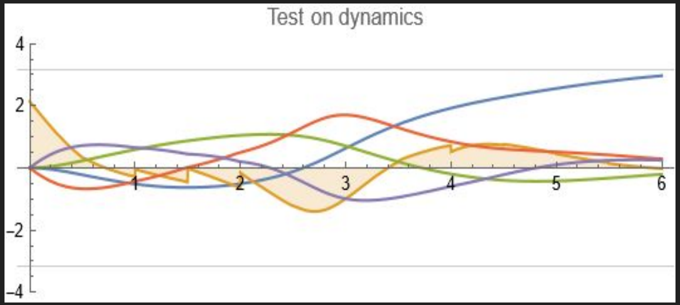
& let J_2 be the cost associated to an optimal sol^n to $(**)$

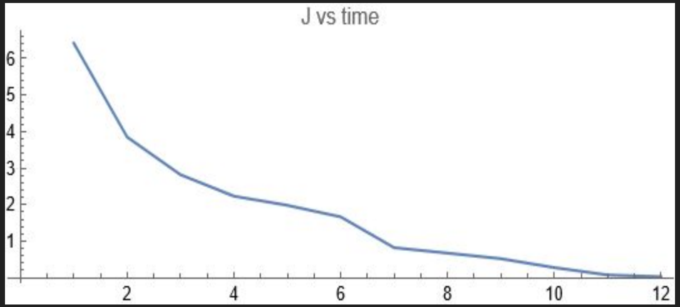
Then, $\boxed{J_2 < J_1}$

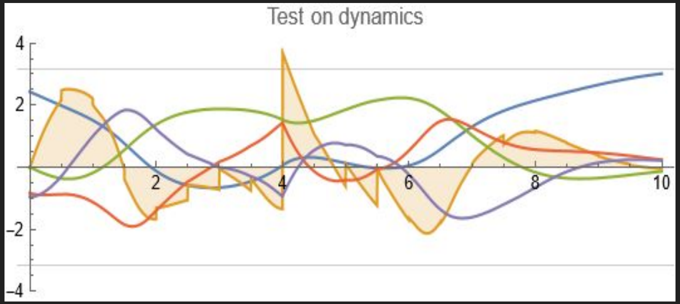
↳ strictly less

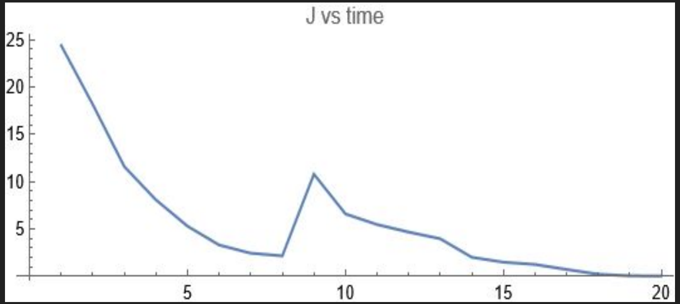


MPC without parameter mismatch

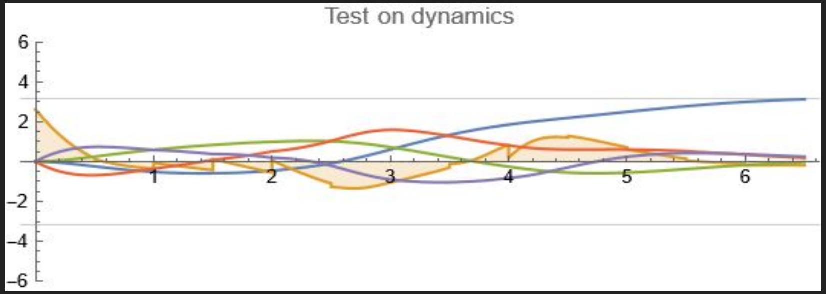


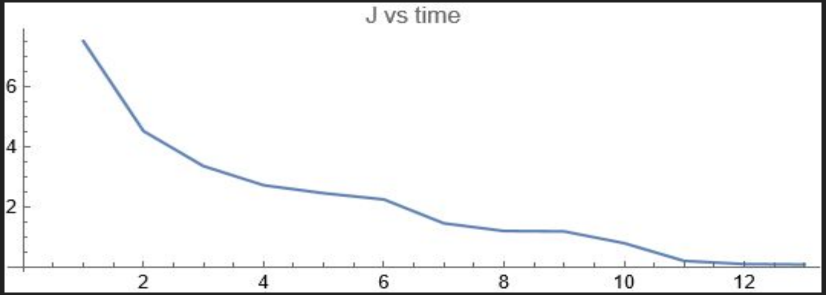


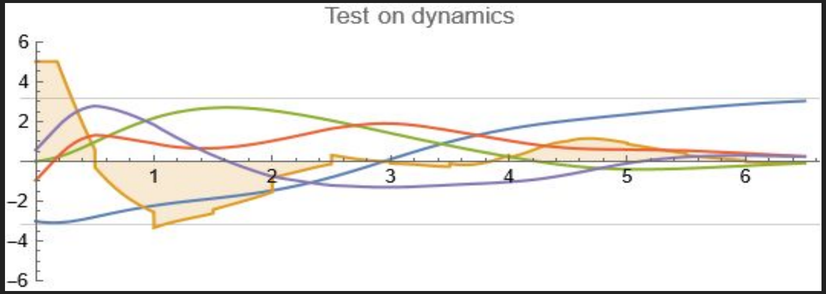


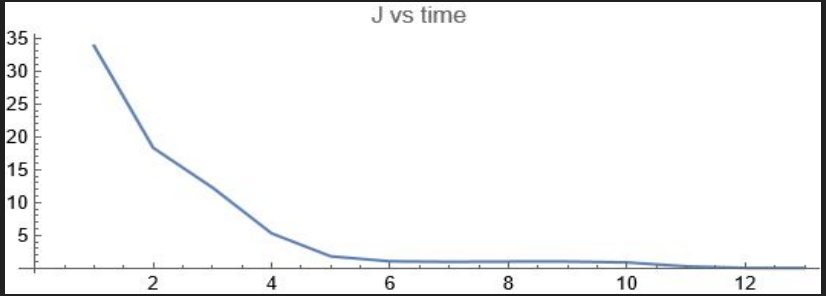


MPC with Parameter mismatch (AUsed = 0, ATrue = 0.2)









Future Work (Simulations)

1] Hammer Perturbations:

Start from {0,0,0,0} & at a Random Time Change $\dot{\theta}$ → $\dot{\theta} + \Delta\theta$ decide this later.

(Use an event triggered periodic recomputation)

Do this Randomly & save the Time Required to Stabilize the Pendulum.

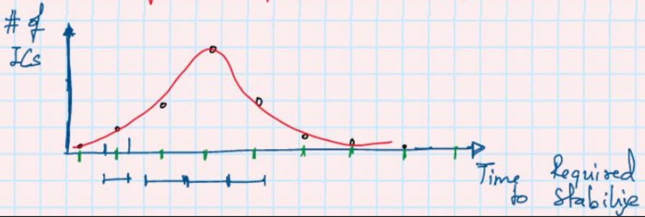
2] Parameter Mismatch

Compute for $A = 0$ & test for $A = 0.3$ or $A = 0.2$ → Do this for ICs = {0,0,0,0}

Computat^n time for $A = 0$ is much less (approx x2)

Periodic recomputations + Event triggered tend to stabilize the dynamics.

↳ Try a bunch of Random ICs & plot the distribution of Time Required to Stabilize



3] Choice of n when using an Event-trigerred Approach.