SWINGING UP THE DOUBLE PENDULUM

Seminar talk at University of Deusto (CMC), Bilbao

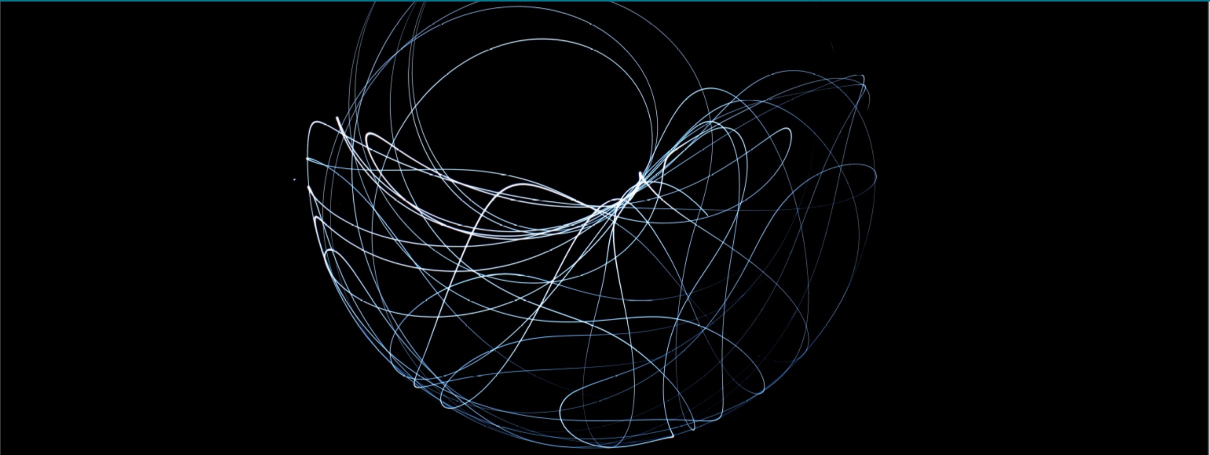
l'X
ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Arnaud Rippol

May 5th 2020

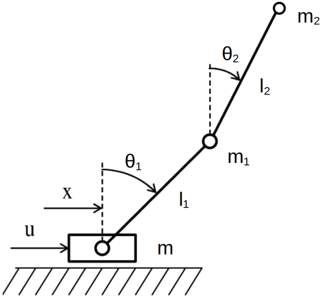
SUMMARY

1
INTRODUCTION
THE DOUBLE PENDULUM



THE SYSTEM

Notations



1. The state : $y = (x, \theta_1, \theta_2, v, \omega_1, \omega_2)$
2. Action : $+u$ (algebraic) on the horizontal acceleration of the cart.

Assumptions

Rods have no mass (hence no inertia), no elastic properties.

A CHAOTIC SYSTEM

Objective

From the downwards position (stable), swinging up the pendulum to the upward position (unstable), thanks to the action u.

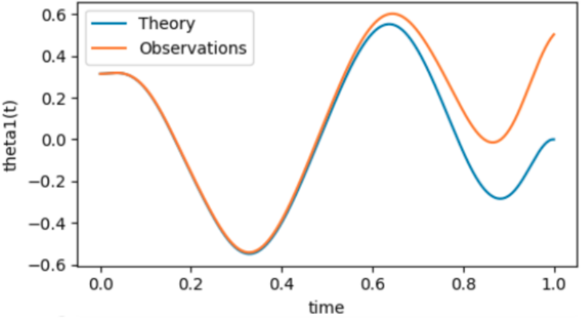
Difficulty

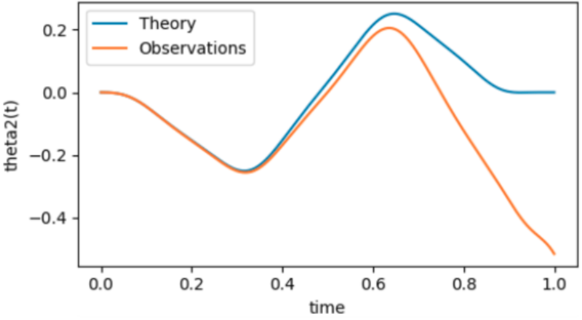
Chaotic aspect : small changes in conditions $\rightarrow$ substantial changes in short term.

Two strategies

1. Using optimal control theory ;
2. Using Reinforcement Learning.

2
OPTIMAL CONTROL THEORY





FINDING A GOOD CONTROL

Cost function

$$J(u) = \int_0^T |y(t)|^2 dt + \alpha \int_0^T u(t)^2 dt + \beta |y(T)|^2$$

with $\alpha, \beta > 0$ to adjust.

Add physical constrains

$\forall t \in [0, T], |u(t)| < u_{max}$

NEED FOR CLOSED LOOP CONTROL

Open loop control is not enough

- No garantee that the system will end exactly in a balanced state,
- Simulation using a different software : calculations are not exactly the same,
- Chaotic aspect : any small deviation leads to a loss of control.

IMPLEMENTING A FEEDBACK

State-Dependent Ricatti Equation

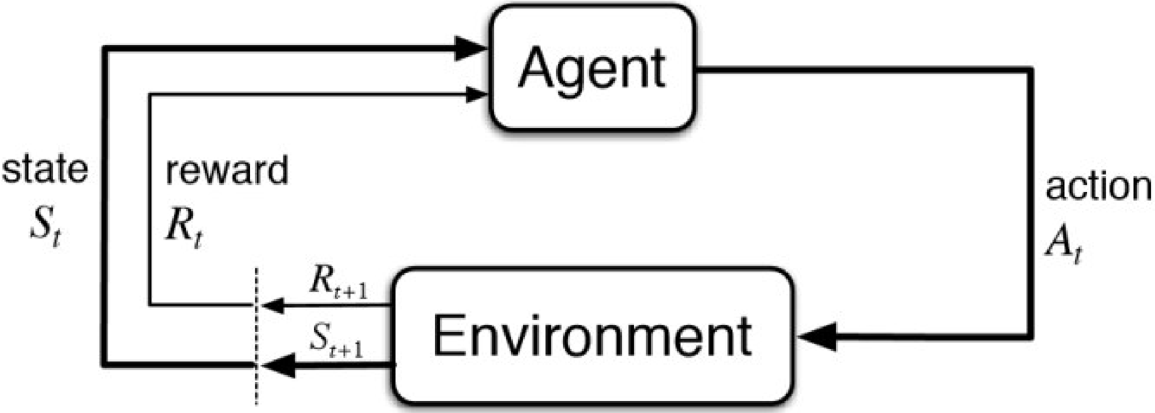
Uses the optimal trajectory $y^*(t)$ as a guideline.

1. Linearize the dynamics of the optimal trajectory equation at time t_m :

$$\dot{y} = f(y, u) \Rightarrow \dot{y}(t_m) = A_m\, y(t_m) + B_m\, u(t_m)$$

2. Apply the LQR theory to this new system between t_m and t_{m+1} : With cost $J = \int_{t_m}^{t_{m+1}} \Delta y^T Q \Delta y + u^T R u \, dt$, retrieve P solution of $A_m^T P + PA + Q - PBR^{-1}B^T P = 0$.
3. Use the following feedback : $u = u^* - B^T P R^{-1} \Delta y$

3
REINFORCEMENT LEARNING



REINFORCEMENT LEARNING ?

Context : an *agent* evolving in an *environment*, taking *actions* depending on its *state*, and receiving *rewards* based on its action and the environemnt.

Goal : select actions to maximize future rewards.

Definitions and notations

- $\mathcal{S}$ state space,
- $\mathcal{A}$ action space,
- $\mathbb{P}(s'|s,a)$ transition function,
- $\mathcal{R}(s,a,s')$ reward function,
- $\pi : \mathcal{S} \to \mathcal{A}$ a *policy*.

OBJECTIVE

Objective : find "best" policy $\pi \Rightarrow$ what should be maximized ? Next reward ? Need to focus on the future / the cumulative rewards.

Definition

- The **Return** at time t : $R_t = \sum_{i=0}^{\infty} \gamma^i r_{i+t+1}$ with $\gamma \in (0, 1]$ a discount factor.
- The **Value function** : $V^\pi(s) = \mathbb{E}[R_t | s_t = s]$
- The **Action-Value function** : $Q^\pi(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a]$

Example

The Greedy policy consists in $a_t = \pi(s_t) = \arg\max_{a \in \mathcal{A}} Q(s_t, a)$

AN EXAMPLE : Q-LEARNING

Main difficulty : finding Q^π or V^π due to the expectation in their difference $\Rightarrow \mathcal{S}$ and $\mathcal{A}$ can be very big ! Need to learn them.
In theory, $Q(s_t, a) = r_t + \gamma Q(s'_t, a)$, but not the case if we use empirical value for Q.

$$Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in \mathcal{A}} Q^t(s'_t, a) - Q^t(s_t, a_t))$$

for $t \leftarrow 0$ *to* $T-1$ **do**
  $s_t \leftarrow \text{StateChoice}\,;\ a_t \leftarrow \text{ActionChoice}$
  $(s'_t, r_t) \leftarrow \text{Simulate}(s_t, a_t)$
  $Q^{t+1} \leftarrow Q_t$
  $Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in \mathcal{A}} Q^t(s'_t, a) - Q^t(s_t, a_t))$
end

PILCO

The dynamic of the problem :

$$f(x_t) = y_{t+1} - y_t, \text{ with } x_t = (y_t, u_t) \in \mathbb{R}^7$$

Gaussian Process

We will assume that f is a *Gaussian Process* :

- $\forall (x_1, \ldots, x_n), (f(x_1), \ldots, f(x_n))$ is a Gaussian vector,
- $m(x) := \mathbb{E}[f(x)]$ is the mean function,
- $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ is the covariate function, or *kernel*.

We assume that the kernel is *Squared Exponential* :
$k(x, x') = \alpha^2 \exp\left(-\frac{1}{2}(x - x')^T \Lambda (x - x')\right)$, with α and Λ to determine.

Idea : for given y_t, u_t, we have $y_{t+1} \sim f(y_t, u_t)$.

PILCO - POLICY AND COST

Policy

We define the policy of the model as followed :

$$\pi(y,\theta) = \sum_{i=1}^{N} \omega_i \phi_i(y), \text{ where } \phi_i(y) = \exp(-\frac{1}{2}(y-\mu_i)^T \Lambda^{-1}(y-\mu_i))$$

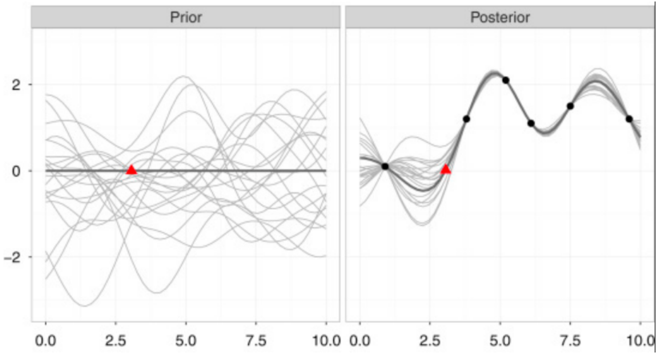
with $\theta = (\omega_i, \Lambda, \mu_i)_{1\leq i \leq N}$

Cost

Cost function of one state : $c(y) = 1 - \exp(-||y||^2/\sigma_c^2)$
Cost of one policy : $J(\theta) = \sum_{t=1}^{T} \mathbb{E}[c(y_t)]$ where the distribution of y_t is computed recursively : $y_{t+1} \sim f(y_t, \pi(y_t, \theta))$

PILCO - FIRST ROLLOUT

At first, no information on the behavior of f. To gain data, random rollout : random actions $(u_t)_{0\leq t<T} \to$ we reccord the data $f(y_t, u_t) = y_{t+1}$.
With this information, we reduce the space in which f can be.



PILCO - ALGORITHM

The algorithm

- With the new information on f, compute $J(\theta)$ (Difficult from a mathematical point of view, need to approximate),
- Minimize J : get $\theta^* = \arg\min J(\theta)$ (Gradient descent),
- With new θ (i.e. new policy), new rollout,
- More data $\rightarrow$ more precise f.

Repeat until the target is reached.

CONCLUSION

Further works

- Finalizing the implementation of both approaches,
- A comparison between the two approaches : speed, resistance to noise...

Questions ?

REFERENCES

- Tayfun Çimen, "State-Dependent Riccati Equation (SDRE) Control : A Survey" in Proceedings of the 17th World CongressThe International Federation of Automatic Control
- M.P. Deisenroth and C.E. Rasmussen, “PILCO : A Model-Based and Data-Efficient Approach to PolicySearch” in Proceedings of the 28th International Conference on Machine Learning
- Michael Hessea, Julia Timmermanna, Eyke Hüllermeierb and AnsgarTrächtlera, "A Reinforcement Learning Strategy for the Swing-Up of the Double Pendulum on a Cart" in Proceedings of 4th International Conference on System-Integrated Intelligence