ILC Exam Report

Implementation of f-ILC

Lorenzo Paiola — lorenzo.paiola@iit.it

Outline

Overview

- Brief summary of Iterative Learning Control
- Implementation of "Iterative Learning in Functional Space for Non-Square Linear Systems" by *C. Della Santina* and *F. Angelini*[1].
- Julia[2] Code found at https://github.com/PaioPaio/ILC_exam

Iterative Learning Control in a Nutshell

Iterative Learning Control[3] (ILC) generally concerns the control of a repeated task. It does so by:

- Closing the loop in the **Iteration Domain** rather than directly time
- Learning just the **Feed-Forward Input**

Remark

ILC assumes that only the initial state is the same at each iteration, no assumptions are made about the terminal state.

System Set up

LTI Continuous Time System

$\dot{x}_j = Ax_j + Bu_j, \quad y_j = Cx_j$ with $x_j \in \mathbb{R}^n, u_j \in \mathbb{R}^l, y_j \in \mathbb{R}^m$

This system is **Iterated** and j indicates the repetition index.

Iterative Learning Control in a Nutshell

Usual Policies

P-type	D-type	I-type
$u_{j+1} = u_j + Le_j$	$u_{j+1} = u_j + L\dot{e}_j$	$u_{j+1} = u_j + L\left(e_{j+1} - e_j\right)$

General Rule

$$u_{j+1} = u_j + \sum_{k=0}^{r} L_k e_j^{(k)} \qquad e_j = \overline{y} - y_j, \quad L_k \in \mathbb{R}^{l \times m}$$

Where r is the relative degree of the system, $e_j^{(k)}$ is the k-th derivative. Notice that we are working with functions.

What is Hard ?

- Usually the system is **non-square**, i.e. $l \neq m$, more interesting is the case where the system is underactuated $l < m$.
- We **sample** $\overline{y}$ only at a finite number of time instants $\{T^1, ..., T^o\}$ and so we have no access to its derivatives or even the true function $\overline{y}$.
- As specified before, we're working with functions and learning functions directly is hard. Moreover, we abuse notation and omit the proper definition of functions and just specify the output domain, eg $y(t) : \mathbb{R}_+ \to \mathbb{R}^m$ becomes $y \in \mathbb{R}^m$.

Functional ILC

> The idea is to not learn u_j directly at each iteration, but to learn some vector of weights α_j that is multiplied to a library of base functions π_j.

What we want is

$$\lim_{j \to \infty} y_j(T^k) = \overline{y}^k, \quad \forall k \in 1...o \tag{1}$$

where $\overline{y}^k$ is the sampled output at time T^k.

Functional ILC - Main Theorem

For a functional basis $\pi = [\pi^1 ... \pi^o] \in \mathbb{R}^{l \times mo}$ and a control function constructed as

$$u_j(t) = \pi\alpha_j, \quad \alpha_{j+1} = \alpha_j + L \begin{bmatrix} \overline{y}^1 - y_j(T^1) \\ \vdots \\ \overline{y}^o - y_j(T^o) \end{bmatrix} \in \mathbb{R}^{mo} \tag{2}$$

Equation 1 is achieved if $\rho(I - LH) < 1$, with

$$H = \begin{bmatrix} \int_0^{T^1} Ce^{A(T^1-\tau)} B\pi(\tau) d\tau \\ \vdots \\ \int_0^{T^o} Ce^{A(T^o-\tau)} B\pi(\tau) d\tau \end{bmatrix} \in \mathbb{R}^{mo \times mo}. \tag{3}$$

Functional ILC - π choice

Any choice of π for which H is full rank is fine.

The paper[1] gives a possible construction that satisfies this condition:

$$\pi^i(t) = \begin{cases} B^\top e^{A^\top (T^i - t)} C^\top & \text{if } T^{i-1} \leq t \leq T^i, \\ 0 \in \mathbb{R}^{l \times m} & \text{otherwise.} \end{cases} \tag{4}$$

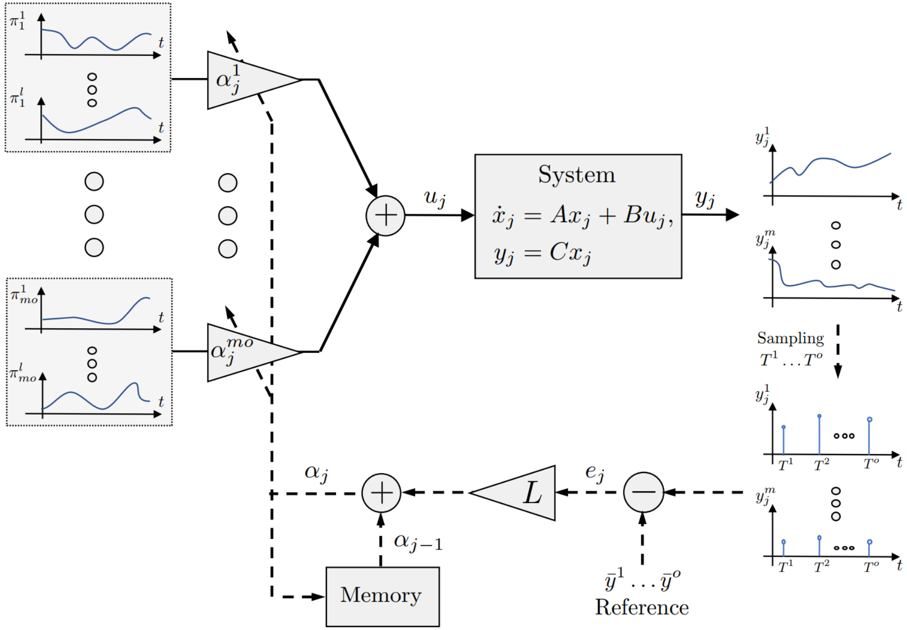
In the case in which the control can be applied only in a limited timespan, the basis

$$\pi^i(t) = \begin{cases} B^\top e^{A^\top (T^i - t)} C^\top & \text{if } T^{i-1} \leq t \leq T^{i-1} + d\tilde{T} \\ 0 \in \mathbb{R}^{l \times m} & \text{otherwise} \end{cases} \tag{5}$$

is appropriate.

The learning matrix is instead set to $L = \left(H^\top H + S\right)^{-1} H^\top \in \mathbb{R}^{mo \times mo}, \quad S = I \cdot 10^{-2}.$

Functional ILC - Summary

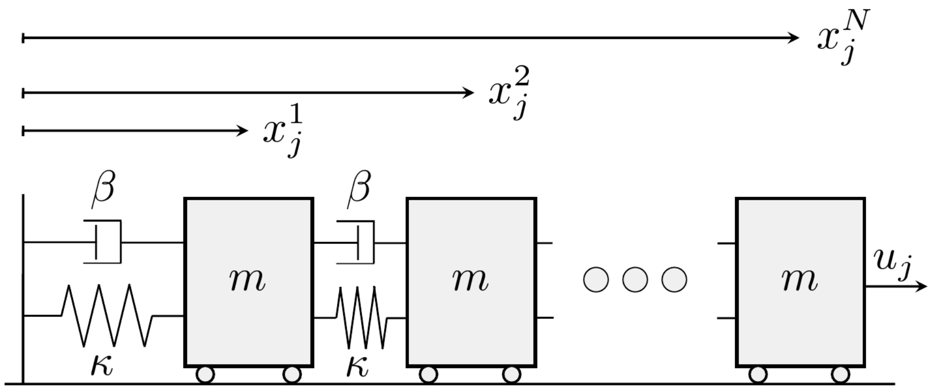


fILC Structure

- $\begin{bmatrix} \alpha_j^1 \\ \vdots \\ \alpha_j^{mo} \end{bmatrix}$ vector of weights updated at each iteration j
- l basis functions for each weight
- Reference given at discrete set of sampled times $\{T^1, ..., T^o\}$, $(T^0 = 0)$
- $L \in \mathbb{R}^{mo \times mo}$ learning matrix s.t. $\rho(I - LH) < 1$

Examples

Carts



Mass-Spring-Damper system actuated by a force on the last cart. Uses Equation 4.

Basketball in the wind

Juggle a **Basketball** in place a few times and then try a free throw. Uses Equation 5.

The details of the systems can be found directly in the paper[1].

Results - MSD, 5 Carts

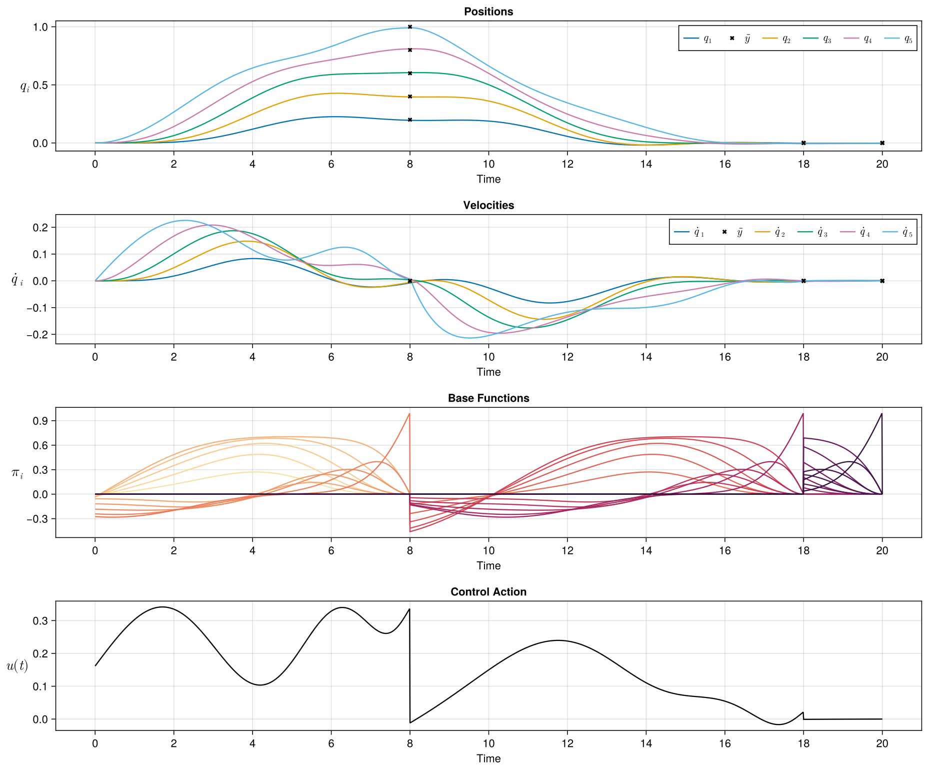


Figure 1: 30 Iterations

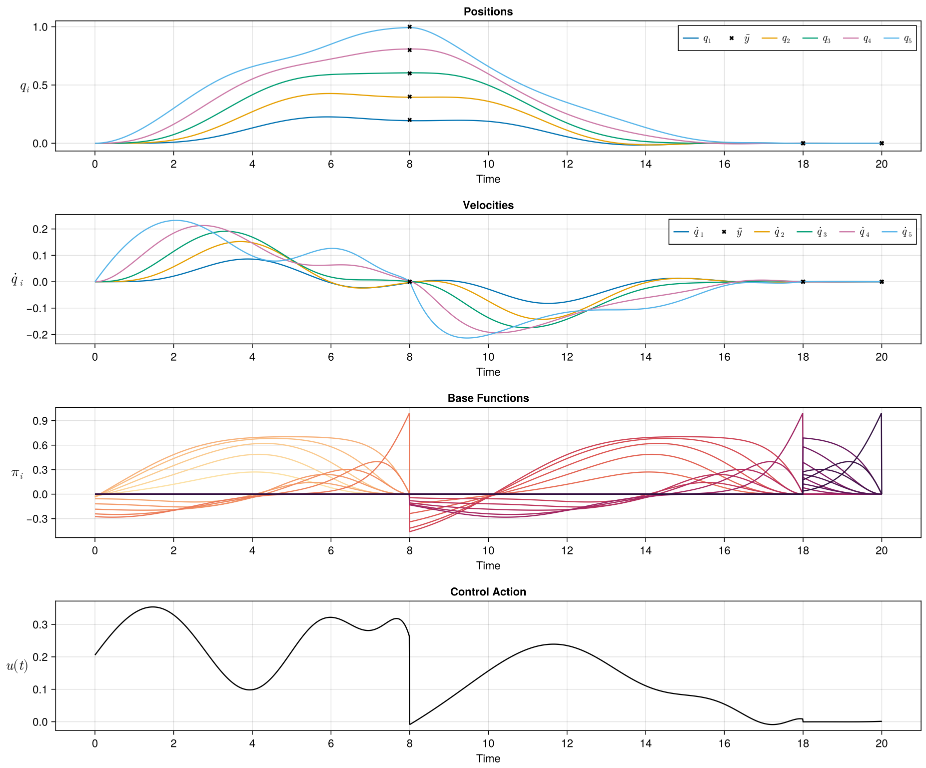


Figure 2: 300 Iterations - Not much changes

Results - MSD, 8 Carts

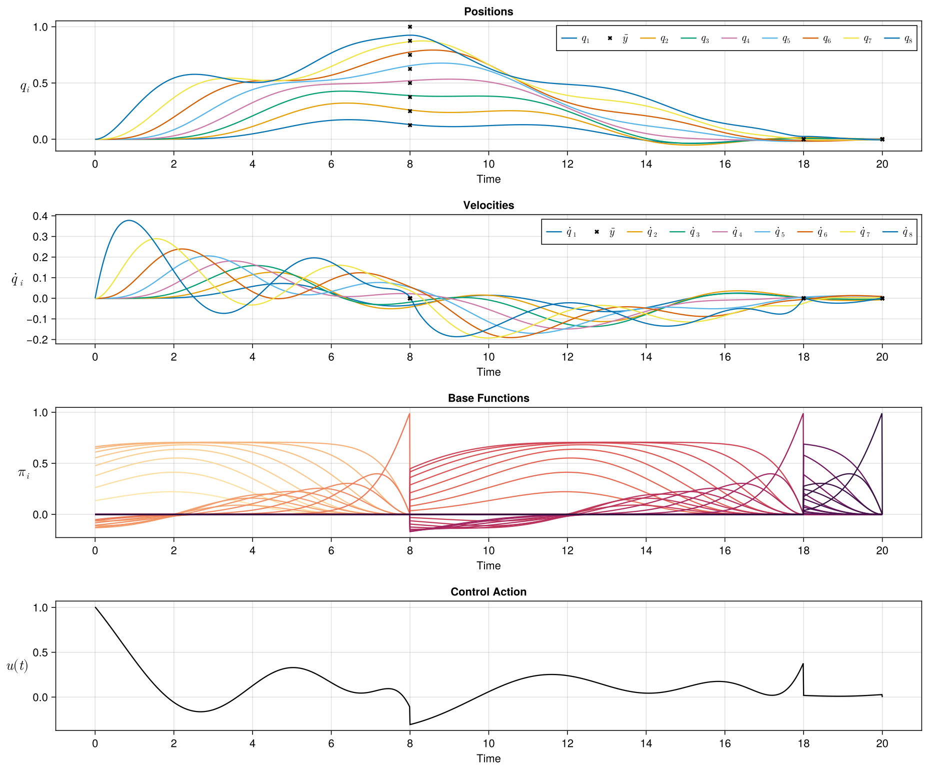


Figure 3: 30 Iterations - Can't perform to specification

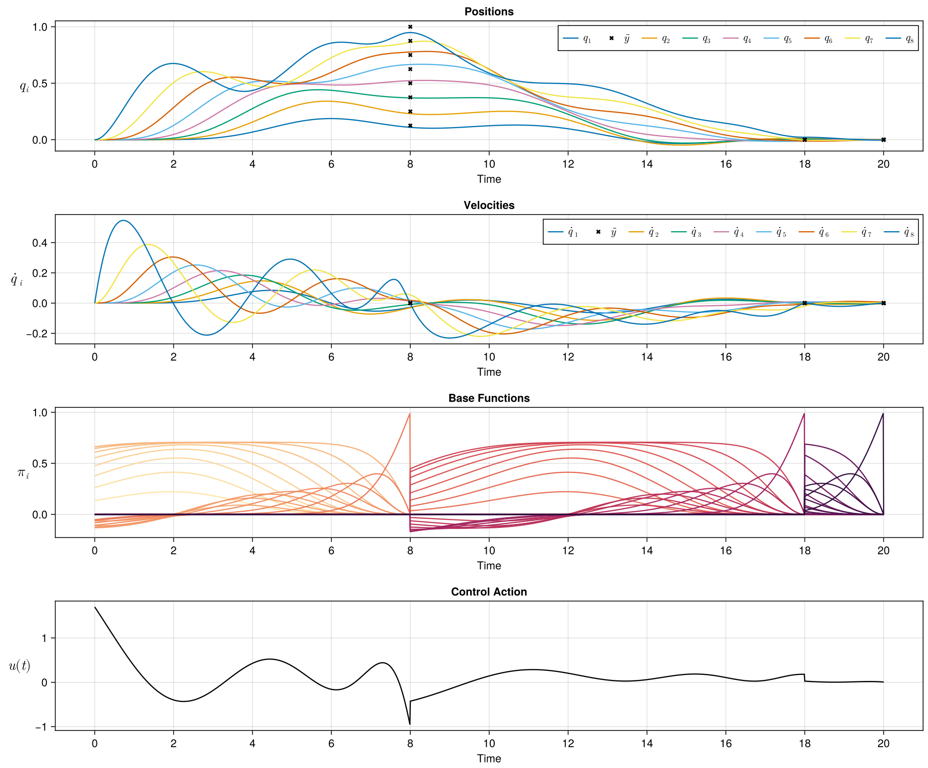


Figure 4: 300 Iterations - Better, still not perfect

Results - Basketball, $d\tilde{T}$ = 0.2s

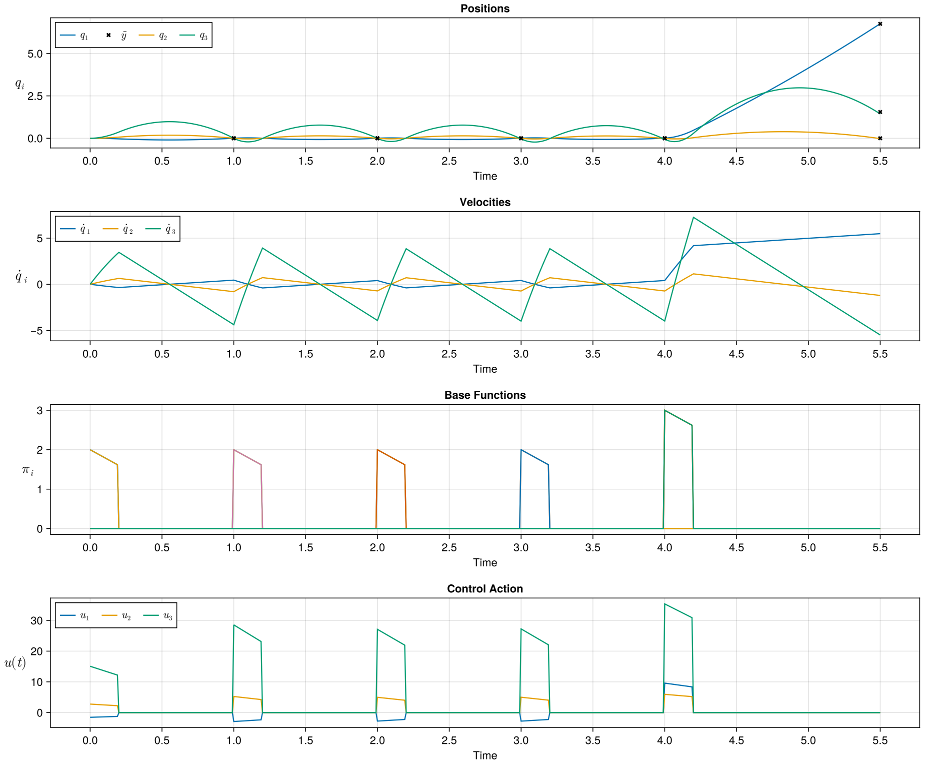


Figure 5: 8 Iterations

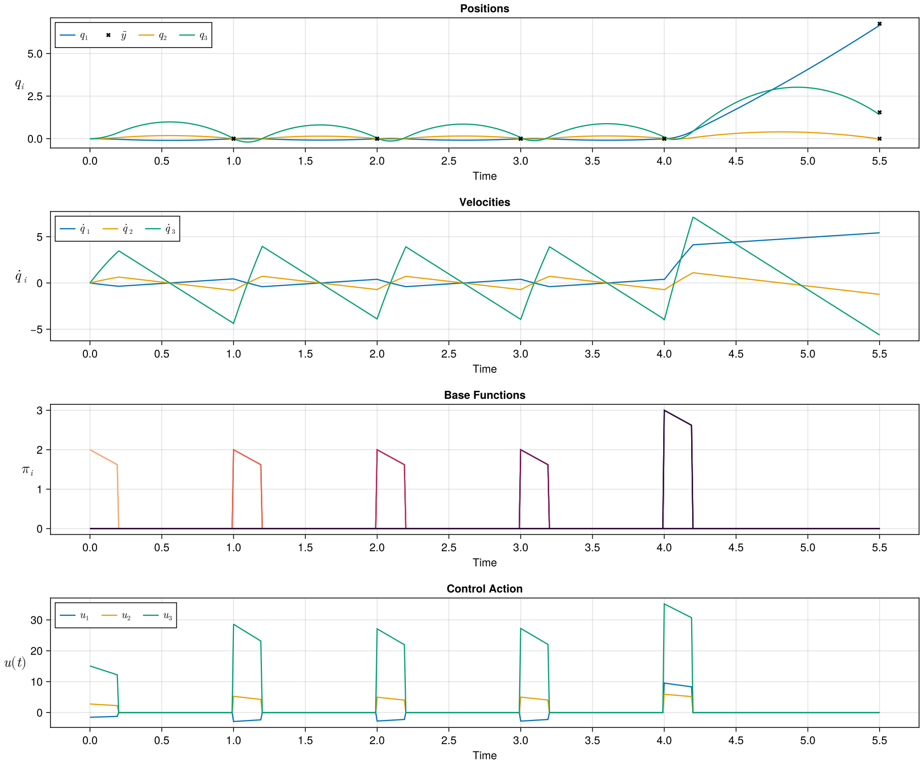


Figure 6: 100 Iterations

Results - Basketball, $d\tilde{T}$ = 0.5s

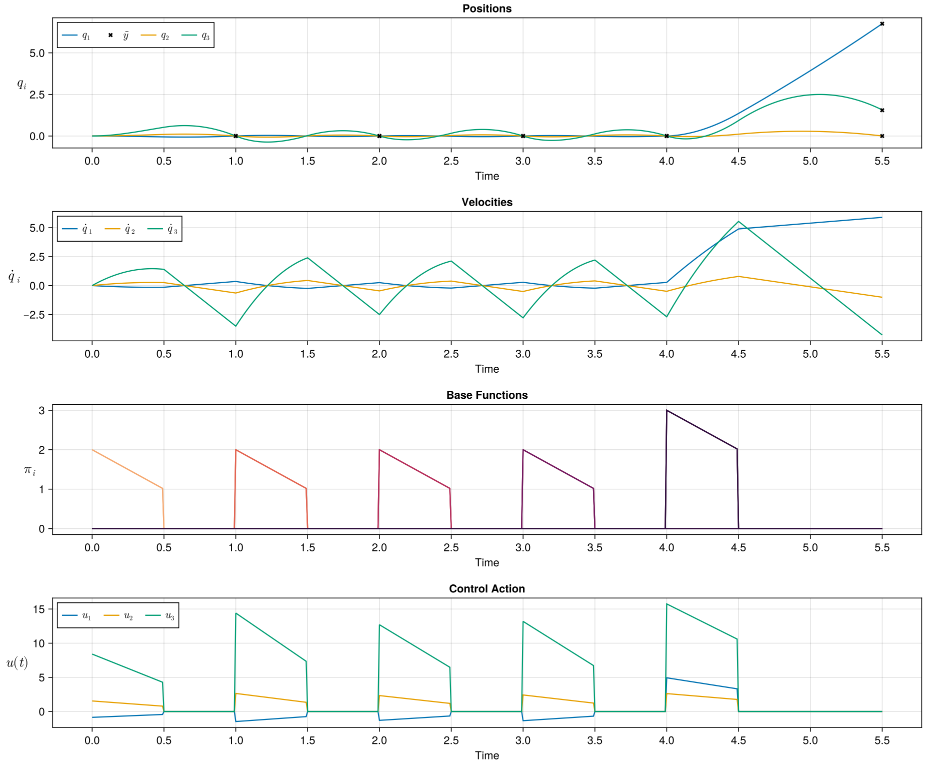


Figure 7: 8 Iterations

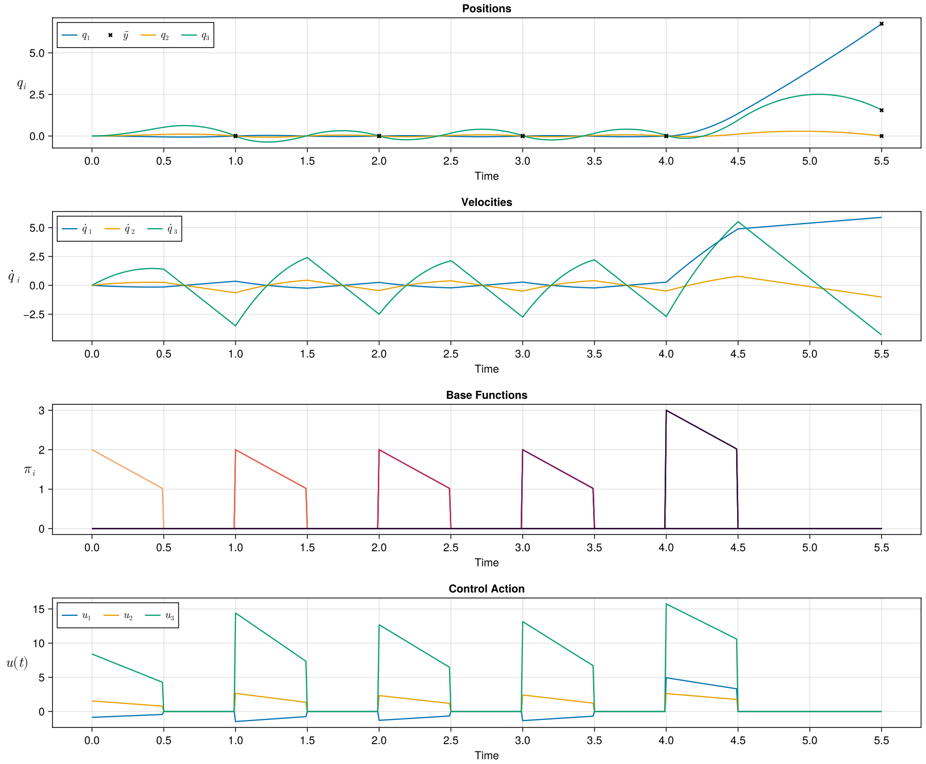


Figure 8: 100 Iterations

Remarks

- If the number of carts is > 8 fILC struggles to converge, for higher numbers it outrights fails even after $\sim 10^3$ Iterations
- The success of the scheme in the basketball example is dependent on the integrator used to simulate the system.
  - This is most likely due to the step-like nature of the constrained basis π_i rendering the linear ODE stiff
  - Strangely even integrators made for stiff ODEs fails if the method order is set too high or the timestep adaptive
  - Integrators that worked were: Heun, ROCK2, Ralston

Bibliography

1. Della Santina, C. & Angelini, F. Iterative Learning in Functional Space for Non-Square Linear Systems. in *2021 60th IEEE Conference on Decision and Control (CDC)* 5858–5863 (IEEE, Austin, TX, USA, 2021). doi:10.1109/CDC45484.2021.9683673

2. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Review* **59,** 65–98 (2017)

3. Bristow, D., Tharayil, M. & Alleyne, A. A Survey of Iterative Learning Control. *IEEE Control Systems* **26,** 96–114 (2006)