POLITECNICO
DI MILANO

Machine Learning for Mechanical Systems

Bayesian Optimization for Robot Arm Control

Authors:
Enrico Bregni
Francesco Pisacane
Michele Taboni

Date:
July 10, 2025

1 Introduction

A robot arm with 4 sections and 7 joints is controlled by means of the following control architecture:

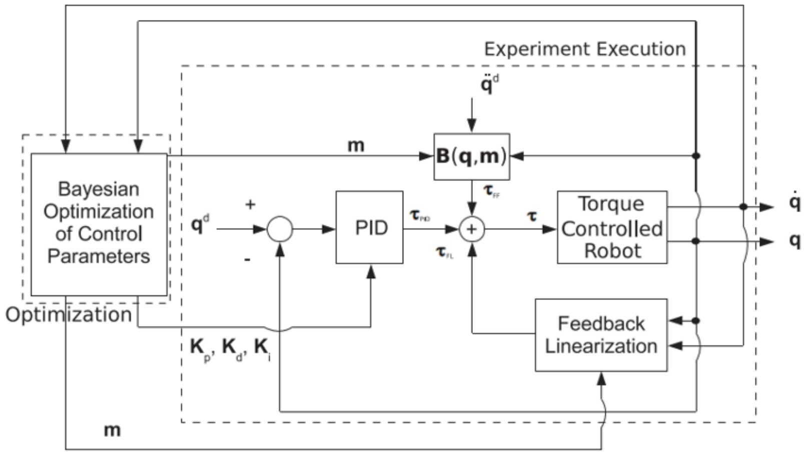

Figure 1: Control architecture

The parameters of the controllers are the PID gains K_p, K_i, K_d for each of the 7 joints and the masses of the individual sections which are used for the feedback linearization and feedforward control. These add up to a total of 25 parameters.

The goal is to optimize the tracking performance i.e. minimize the tracking error. To this end, there are two level of complexity:

1. **PID gains optimization**: the robot is given a set of gains and a position-speed trajectory to follow for each joint. An experiment is conducted measuring the real trajectories from which an error can be calculated:

$$|trajectory_{ideal} - trajectory_{real}|$$

   It is possible to optimize the gains by minimizing the error function.

2. **Masses estimation**: The real robot has hollow section with non trivial mass distribution. It is possible to setup a simplified model of the robot with simple bar joins, however equivalent masses for these sections need to be estimated to guarantee the same inertial behavior. For masses estimation, the gains are set to constant values. The real robot and its digital twin are given identical trajectories to follow and the error is computed as:

$$|trajectory_{real} - trajectory_{digital-twin}|$$

   This because we expect poor tracking performance with respect to the ideal trajectory but what is of interest is that the two robots show the same behavior.

To increase the time efficiency of the simulation we want each experiment to be as short as possible but still hold physical validity. To this end, a shorter transient is desirable so to reach the steady state faster. This condition can be favored with an appropriate choice of reference trajectory. A smoother reference with null initial derivative is used in order to limit initial torque and consequently transient duration:

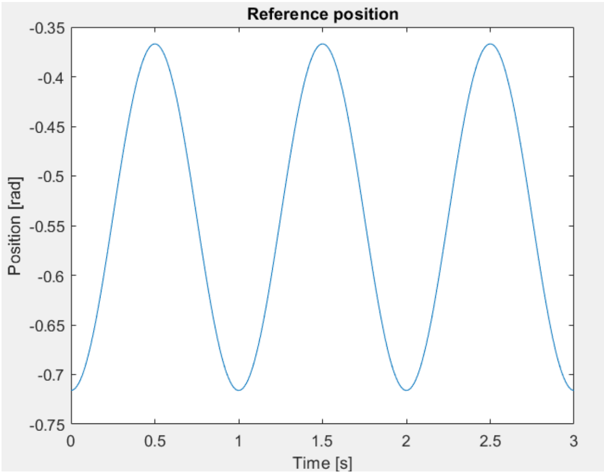

Figure 2: Reference trajectory

This trajectory is derived in time to obtain the velocity reference. They are then used for each joint.

2 Bayesian Optimization Implementation

The previously mentioned experiments require a minimum duration of a few seconds in order for the robot to reach steady state. Due to this fact, it is time consuming (resource expensive) to perform a large number of experiments.

When experiments are expensive to perform, **Bayesian Optimization** is a valid solution. This algorithm tries to mimic the (unknown) explicit *Cost Function* of the problem by means of a *Surrogate Function* which leverages the use of the *Gaussian Process*. This returns the expected value of the surrogate function and its statistical distribution given a set of samples obtained from experiments. Bayesian Optimization limits the number of experiments by selecting the parameters for the next experiment based on a smart strategy. The strategy is governed by an *Acquisition Function*.

2.1 Cost Function Definition

As we have said before, the first step is to define the *Cost Function*. Since the explicit one is not trivial to be found we use an implicit formulation. The final decision is:

$$J = (\max_t |q_{\text{err}}| + \max_t |\dot{q}_{\text{err}}| + \frac{1}{n_t}\sum_{j=1}^{n_t} |q_{\text{err}}| + \frac{1}{n_t}\sum_{j=1}^{n_t} |\dot{q}_{\text{err}}|) + L$$

In the software implementation in order to reduce the total optimization time, an experiment which shows unstable behavior should be terminated immediately. In the implementation, "instability" is defined as exceeding a preset threshold on the instantaneous angle error with respect to the reference trajectory. We want to penalize unstable parameter configurations however we would like to distinguish configuration for which instability is reached immediately or further in the experiment. To this end, a time dependent penalty factor **L** is introduced:

$$L = 1000\left(1 - \frac{(1-p)}{\log_{10}(T)}\,\log_{10}(t)\right) \qquad (1)$$
$$p = 0.01$$

- T: total experiment time
- t: time variable
- p: final scaling for logarithmic penalty

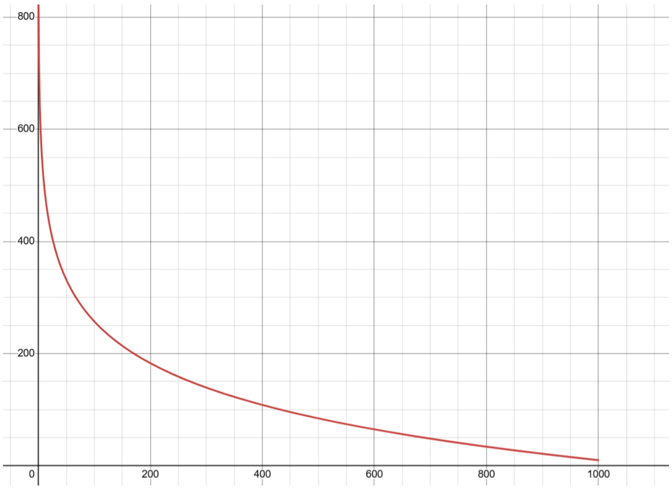

Figure 3: Example of L(t) with T = 1000

2.2 Acquisition Functions

After defining the Cost Function we have to define the tool that guides the Bayesian Optimization, the *Acquisition Function*. This function determines the parameters to use for the next experiment. Many acquisition functions exits, some aims at investigating the behavior of the surrogate function around the area with higher uncertainty (**exploration** focused) while other aim at performing experiments around the area where the expected value is more promising (**exploitation** focused).

In our case we have decided to use at first *exploration*, to (again) "explore" the whole parameters' space. Then, after several simulation, we finally use *exploitation* focusing the optimization around the best point found during the latter phase. This is the best approach to avoid settling in local-minimum. The MATLAB implementation of the following acquisition functions:

Lower Confidence Bound is used for exploration:

$$\mathrm{LCB}(x) = 2\sigma_Q(x) - \mu_Q(x).$$

Expected Improvement is used for exploitation:

$$EI(x, Q) = \mathbb{E}_Q\left[\max\left(0, \mu_Q(x_{\text{best}}) - f(x)\right)\right].$$

3 PID Parameters Estimation (all joints combined)

Focusing only on the PID gains, a first solution approach is to perform a Bayesian Optimization on all of the gains simultaneously which amount to a total of 21 unknowns. This is already close to the upper limit of number of parameters for which BO is expected to perform appropriately hence expectations for the results are limited.

The optimization is conducted with 350 exploration and 140 exploitation iterations. The evaluations of the Objective Function for successive iterations are shown in the following figure:

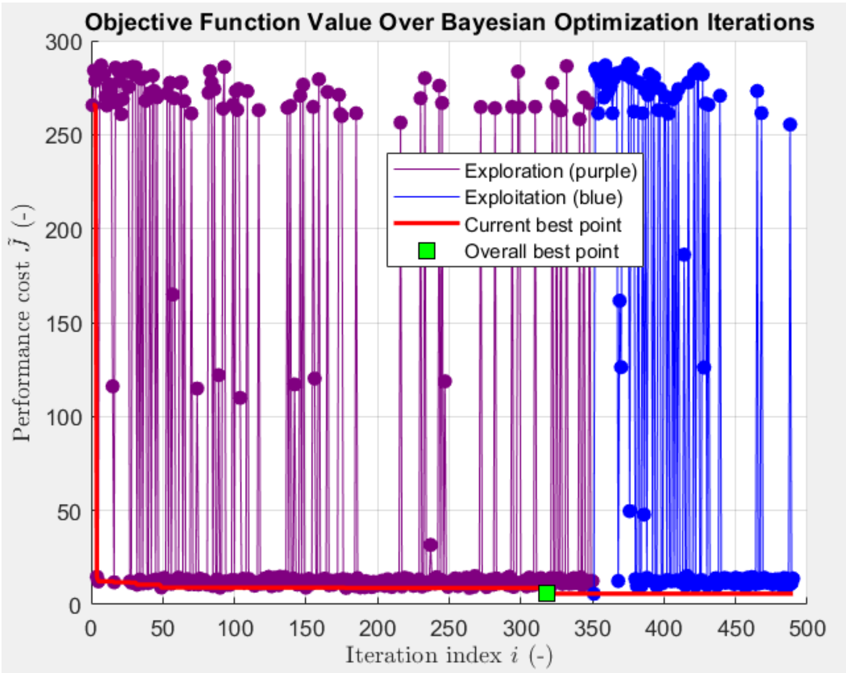

Figure 4: Objective function evaluations over iterations PID gains simultaneously

During the entire optimization, it can be noticed that the objective function has frequent increases. This is to be expected when dealing with a high dimension parameter space, a slight variation in few parameters can cause a drastic change in performance.

The effectiveness of the process can be evaluated by observing the tracking performances of the tuning.

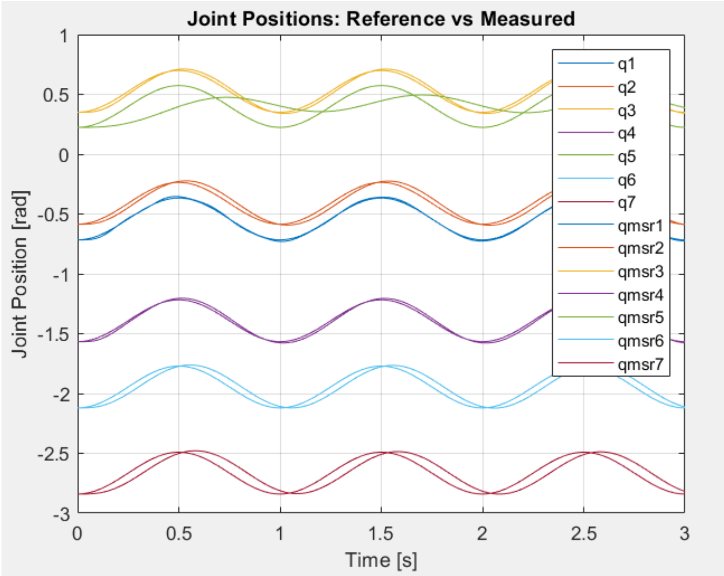

Figure 5: Reference position VS Measured position for PID gains simultaneously

Overall the robot seems to follow the trajectory however visible problems can be noticed for joints 5 and 7.

The same comparison can be done for velocity:

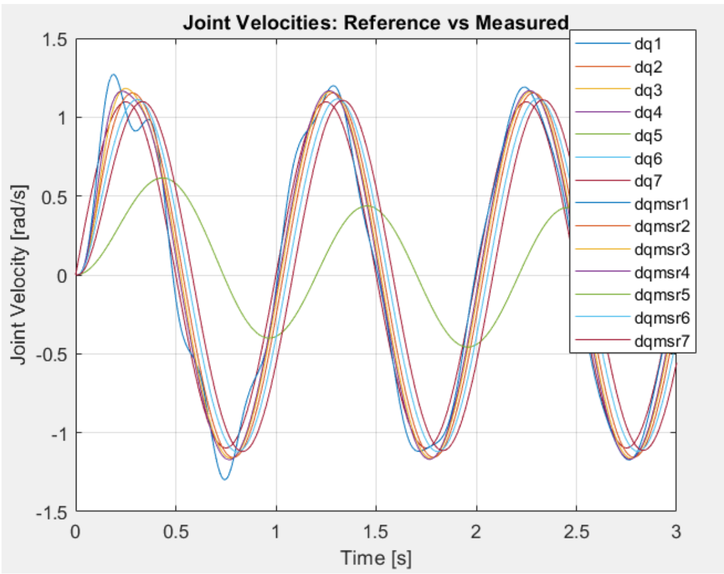

Figure 6: Reference speed VS Measured speed for PID gains simultaneously

As it can be observed from Figure 6, the poor tracking performances for joint 5 also carries on in the velocity while a noticeable transient for joint 1 is present.

Lastly the torques given by the PID controller can be plotted:

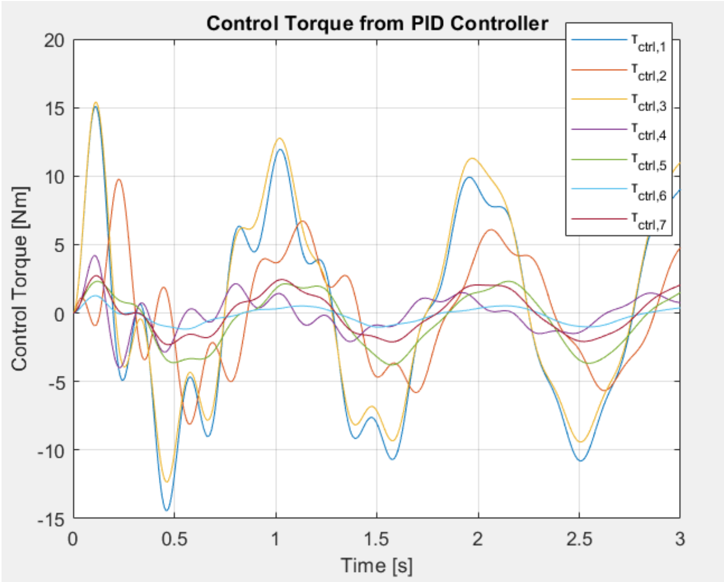

Figure 7: PID torque for PID gains simultaneously

The torque profiles hint, despite not clearly, to a periodic behavior which is what is expected from an harmonic trajectory. They are expected to be in phase however this does not seem evident in the graph.

	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6	Joint7
K_p	760.52	833.52	655.43	786.65	813.16	948.94	595.13
K_i	61.21	724.55	224.73	880.10	645.28	739.94	628.19
K_d	1.73	30.23	15.00	14.14	315.34	54.19	49.44

Table 1: Values for the gain parameters for all the joints together.

4 Masses Estimation (all sections combined)

As mentioned in the Introduction (1), the masses are estimated for a digital twin of the robot (in our case 4 section hence 4 masses). A first approach is to conduct a Bayesian Optimization for the 4 masses simultaneously. The optimization was conducted with 50 exploration and 15 exploitation iterations.

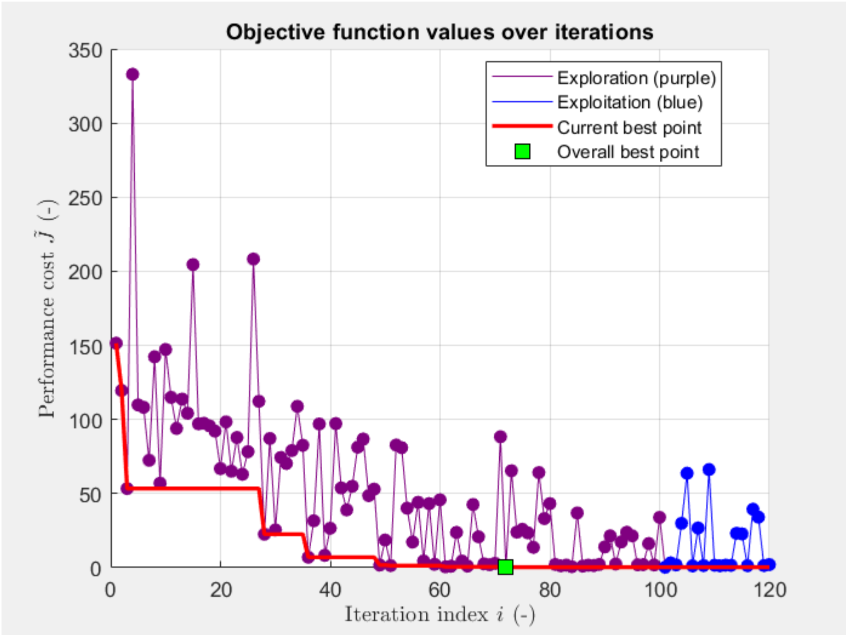

Figure 8: Objective function evaluations over iterations

mass 1	mass 2	mass 3	mass 4
0.8934	2.7079	5.0291	2.3230

Table 2: Estimated masses [Kg] , optimized simultaneously

To asses the validity of the estimation is it possible to compare the compensation torque with the torque on the joint due to gravity. If the masses are estimated correctly the two should match.

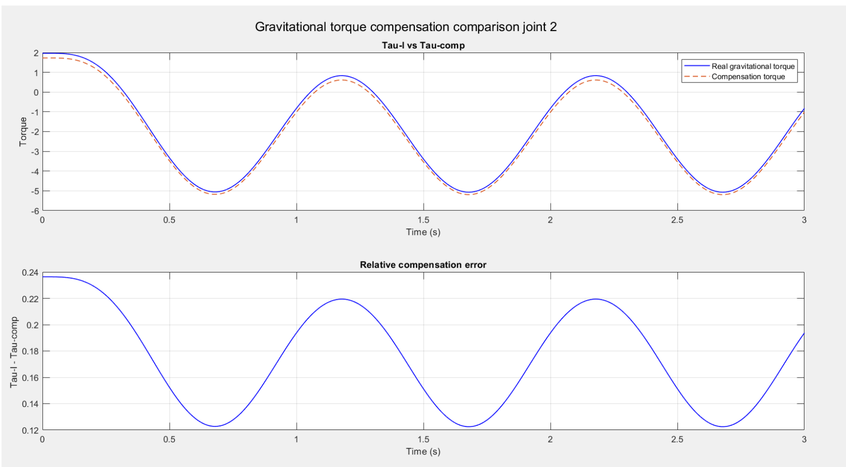

Figure 9: Comparison gravitational torque vs compensation torque joint 2

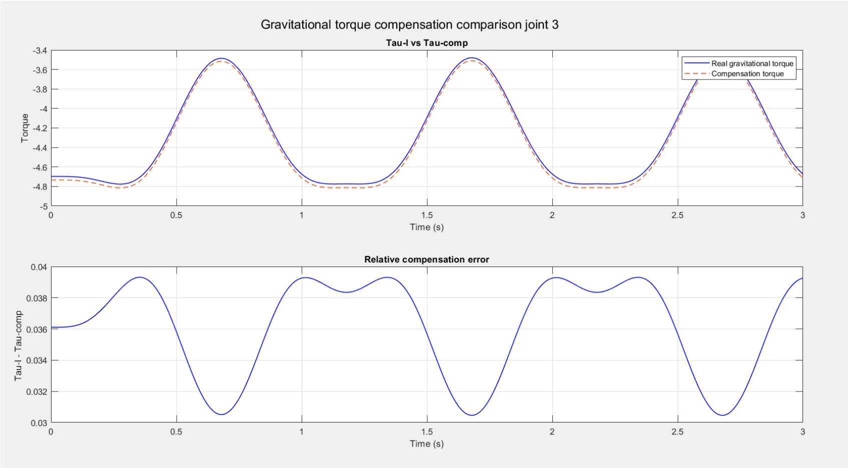

Figure 10: Comparison gravitational torque vs compensation torque joint 3

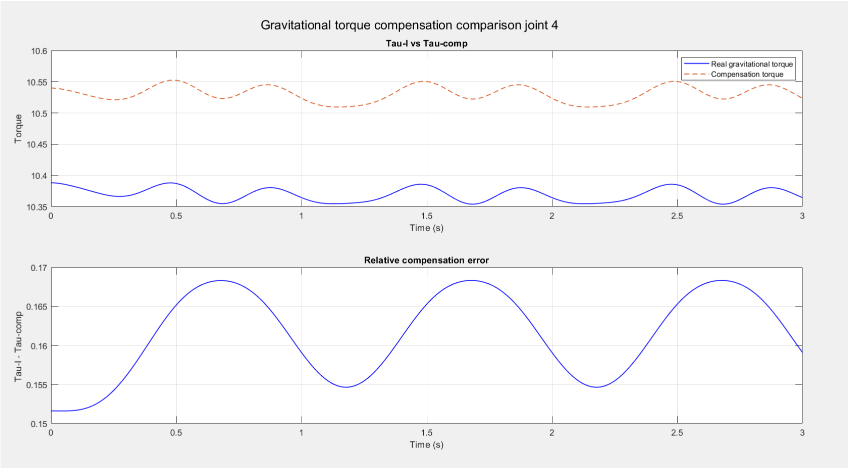

Figure 11: Comparison gravitational torque vs compensation torque joint 4

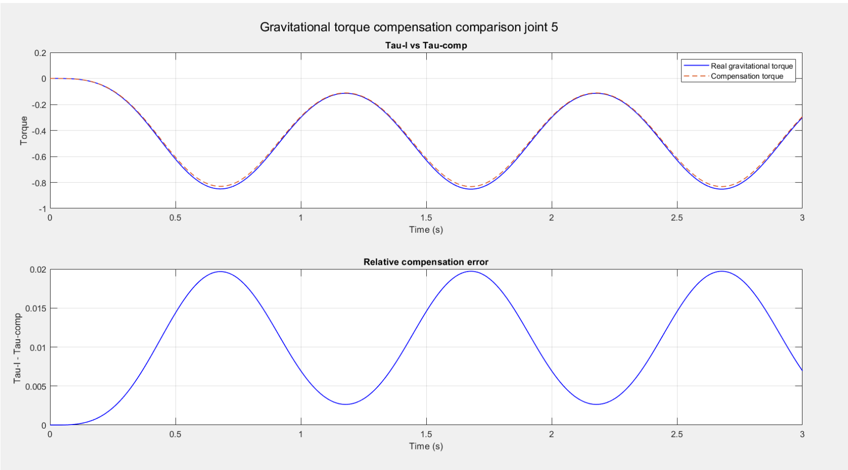

Figure 12: Comparison gravitational torque vs compensation torque joint 5

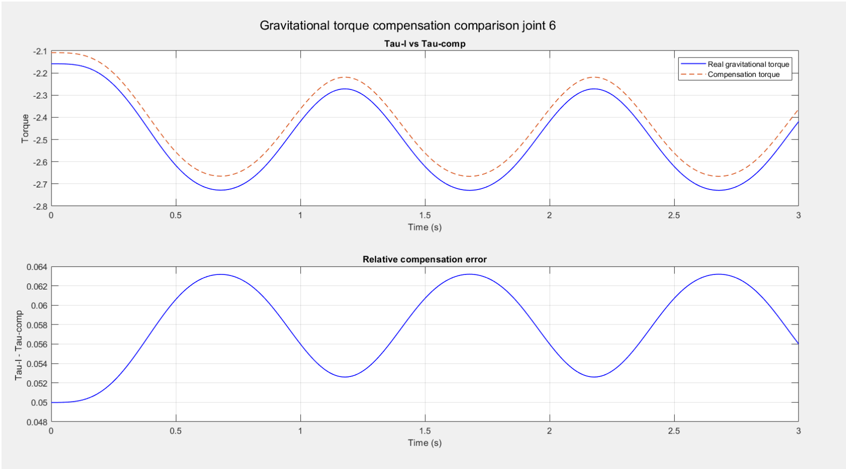

Figure 13: Comparison gravitational torque vs compensation torque joint 6

As it is clear from the above figures, the compensation seems to perform according to expectation but it is left to wonder if further improvement is possible. Is this the best way to approach the problem?

5 Masses Estimation (section by section)

Thinking physically at the problem, it is clear that many masses configurations can lead to very similar behavior with respect to the gravitational load, especially since we are dealing with a single gain parameters configuration. We expect then to have a problem for which many local minima are present. Such a problem benefits from an exploration focused approach which is what was implemented previously and lead in good results.

However, the problem is separable in 4 individual one-dimensional sub-problems i.e. the masses can be estimated one by one. This is true only by implementing a specific solution strategy: estimating the masses of the joint sequentially starting from the end effector all the way down the arm. This strategy works as every joint moves only the sections of the robot that comes "after" it (from the joint towards the end effector) hence by locking the whole arm and moving only the joint before the end effector, the latter can be estimated. The end effector mass can then be set as a constant and the mass of the section before it, let to be the new single unknown of the next optimization.

With section 4 being the end effector and section 1 being the closest to the base.
Pseudo algorithm:

- set mass 4 as the single unknown
- lock all joints beside the joint moving section 4
- estimate mass 4 with 1D Bayesian Optimization (also simpler algorithm could be used)
- set the estimated mass 4 as a constant of the model
- set mass 3 as the single unknown
- lock all joints beside the joint moving section 3 (which also moves section 4 as it comes after)

- estimate mass 3 with 1D Bayesian Optimization
- set the estimated mass 3 as a constant of the model
- loop until mass 1 is estimated

The reduction to 1D sub-problems makes each mass estimation univocal with respect to the behavior of the robot hence we are guaranteed to estimate the true mass of the section. Each sub-problem can be solved with a reduced number of iterations and it could be also solved with simpler algorithms than Bayesian Optimization.

Following are illustrated the evaluations of the objective function for the 4 mass estimations. 15 exploration and 15 exploitations iterations are used.

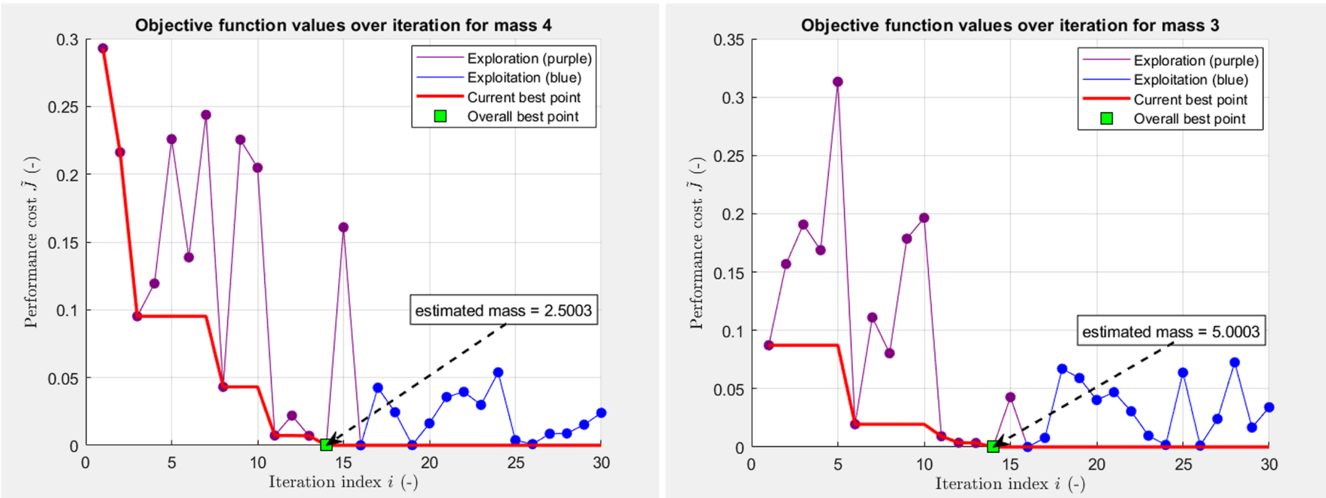

Figure 14: Objective function evaluations over iterations masses 4 and 3

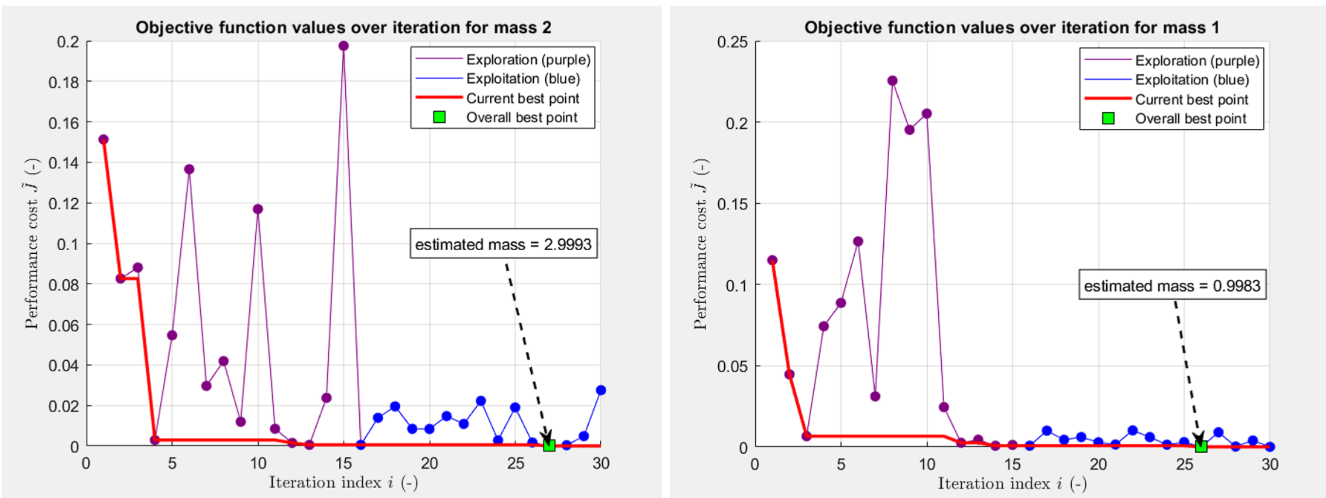

Figure 15: Objective function evaluations over iterations masses 2 and 1

Again the validity of the estimation can be assessed comparing the compensating torque with the torque on the joint due to gravity.

mass 1	mass 2	mass 3	mass 4
0.9983	2.9993	5.0003	2.5003

Table 3: Estimated masses [Kg] , optimized simultaneously

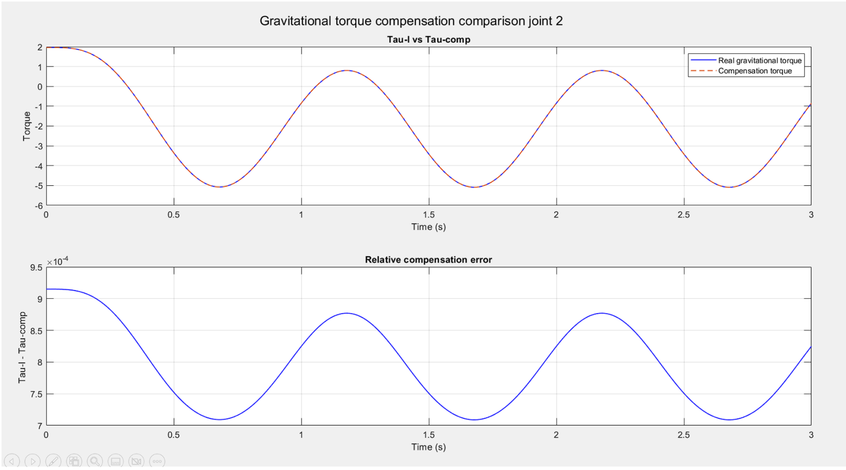

Figure 16: Comparison gravitational torque vs compensation torque joint 2

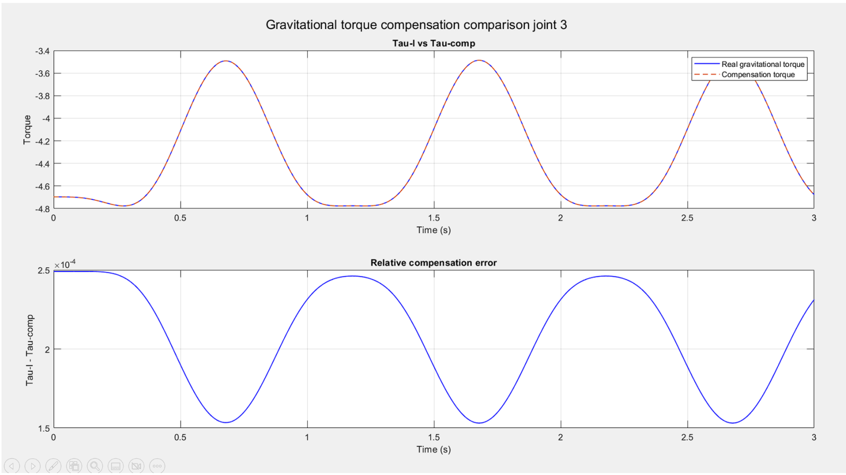

Figure 17: Comparison gravitational torque vs compensation torque joint 3

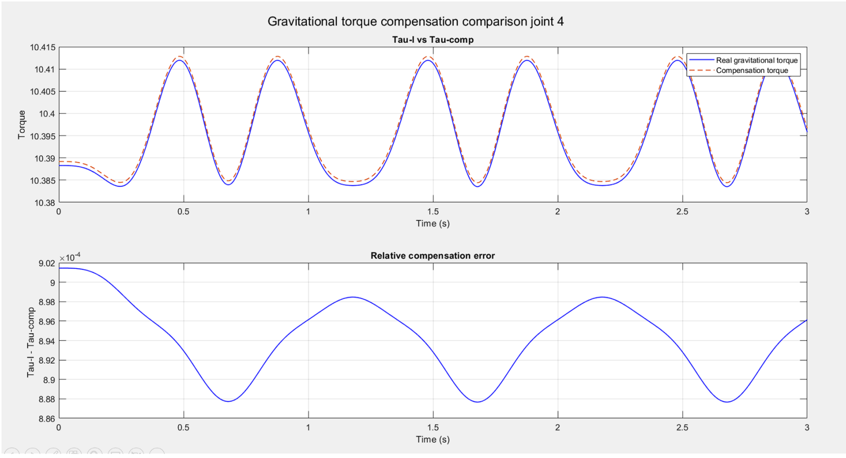

Figure 18: Comparison gravitational torque vs compensation torque joint 4

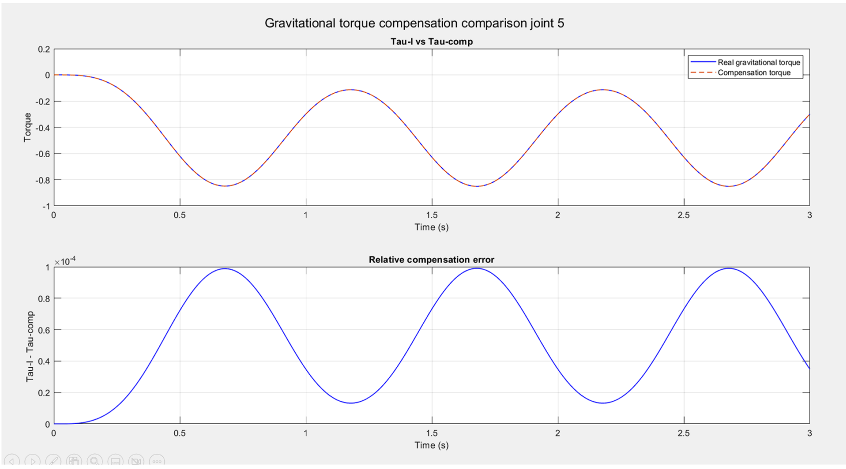

Figure 19: Comparison gravitational torque vs compensation torque joint 5

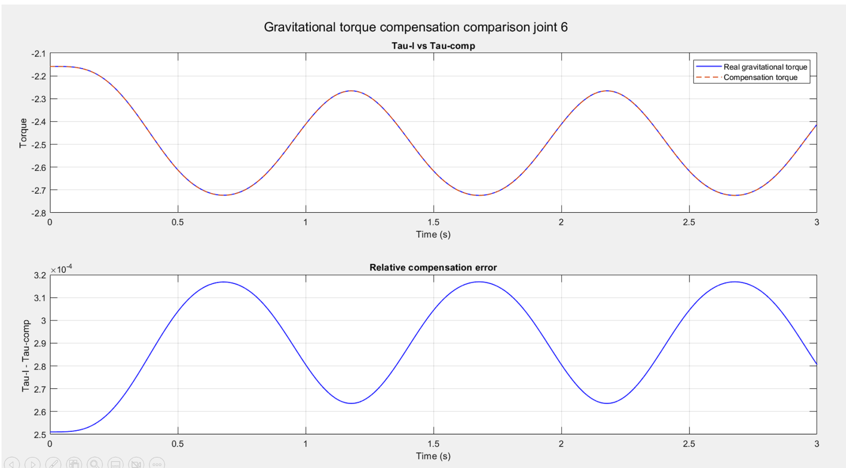

Figure 20: Comparison gravitational torque vs compensation torque joint 6

As it can be seen from the above figures, the more physical based approach proved to lead to better results. The matching is nearly perfect with errors in the order of 10^{-4} Nm.

6 PID Parameters Estimation (joint by joint)

As it was done previously for the masses it is also possible to perform the PID gains optimization joint by joint. Thus, instead of optimizing 21 parameters all together, which is already in the neighborhood of the maximum capabilities of Bayesian Optimization, 7 individual optimization with 3 parameters each are performed.

The Acquisition Functions and the Cost Functions used are the same as the ones inmplemented in *PID Parameters Estimation (all joints combined)* (3). Then two simulations were performed varying the upper bound for the proportional gains:

- $K_p \leq 10000$
- $K_p \leq 1000$

These two will be presented in the following pages.

6.1 $K_p \leq 10000$

The first optimizations set is performed with a relatively high upper bound for proportional gain in order to see what the estimated values will converge to, with greater freedom. The optimization is done using 50 iterations for exploration and 20 for the exploitation. As said before this optimization should be done for all the 7 joints.

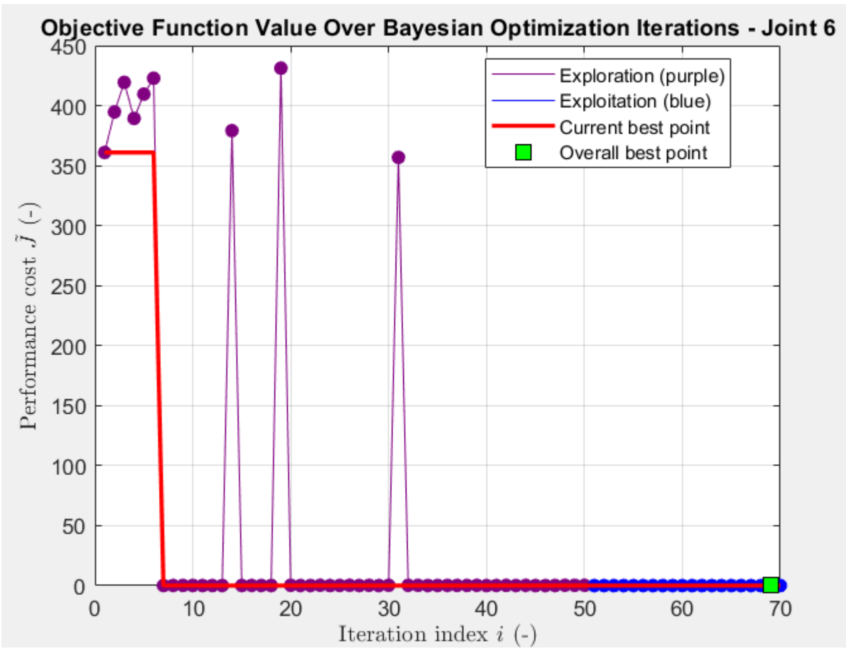

Figure 21: Objective function evaluations for joint 6 over iterations

The results found after this optimization are shown in the following figures. Firstly, excellent tracking performances are observable for all the joints:

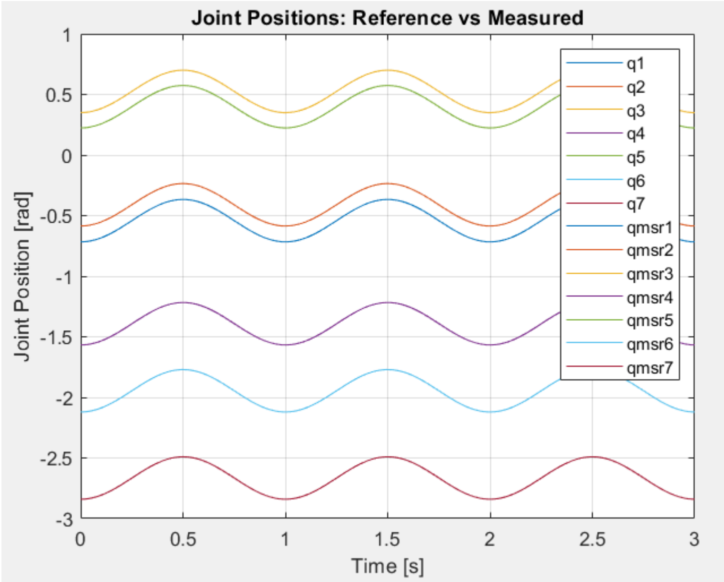

Figure 22: Reference position VS Measured position

The same comparison can be done for velocity:

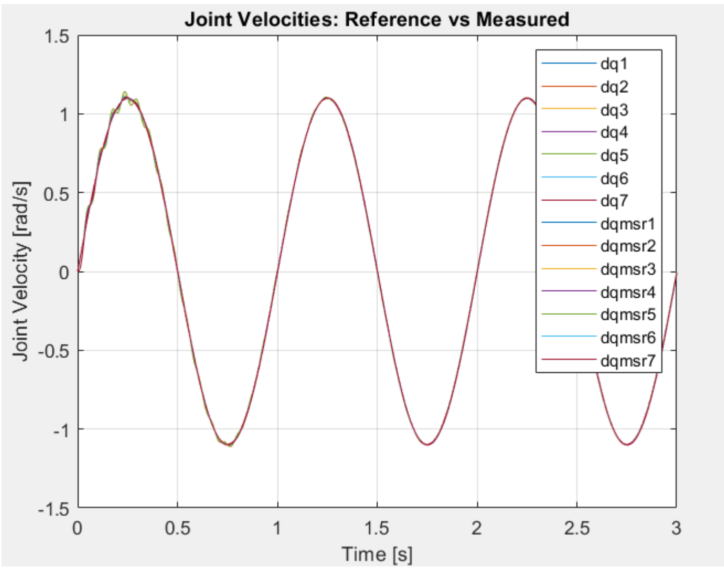

Figure 23: Reference speed VS Measured speed

As it can be observed from Figure 23, a more evident initial oscillation is present in the speed compared to position. This is due to the fact that small oscillations in the position response, for an harmonic response, are naturally amplified by the derivative over time.

Lastly the torques given by the PID controller can be plotted:

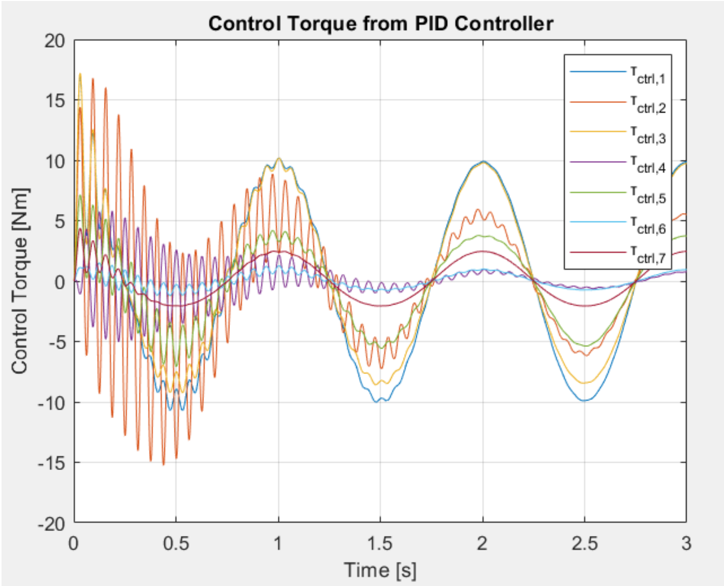

Figure 24: PID torque

Moreover, the estimated values for the proportional gains can be listed in a table:

K_{p1}	K_{p2}	K_{p3}	K_{p4}	K_{p5}	K_{p6}	K_{p7}
9999.16	9999.89	9999.48	9999.17	9999.32	9999.05	9851.83

It is observable how all the proportional gains tend to saturate to the upper bound of 10000. This is an expected result, since an higher proportional gain maximizes the tracking performances. This is due to the fact that higher K_p are associated with a stiffer response which although mathematically favorable has severe implication in real world implementation. It can cause high frequency oscillations as in Figure 23 and torque spikes, see Figure 24. These can contribute to hardware wear and even cause critical failure hence it was decided to reduce the upper bound for the proportional gain, in order to have a more realistic result.

The results obtained are the following:

	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6	Joint7
K_p	9999.16	9999.89	9999.48	9999.17	9999.32	9999.05	9851.83
K_i	266.86	21.10	413.93	207.35	172.87	503.49	99.80
K_d	15.59	3.94	10.52	26.14	4.31	13.54	18.65

Table 4: Values for the PID gain for each joint, $K_p \leq 10000$.

6.2 $K_p \leq 1000$

For the reasons stated previously, the upper bound for the proportional is reduced to 1000. Now, following the same path shown for the case of $K_p \leq 10000$, we start by showing the evolution of the objective function over the iterations:

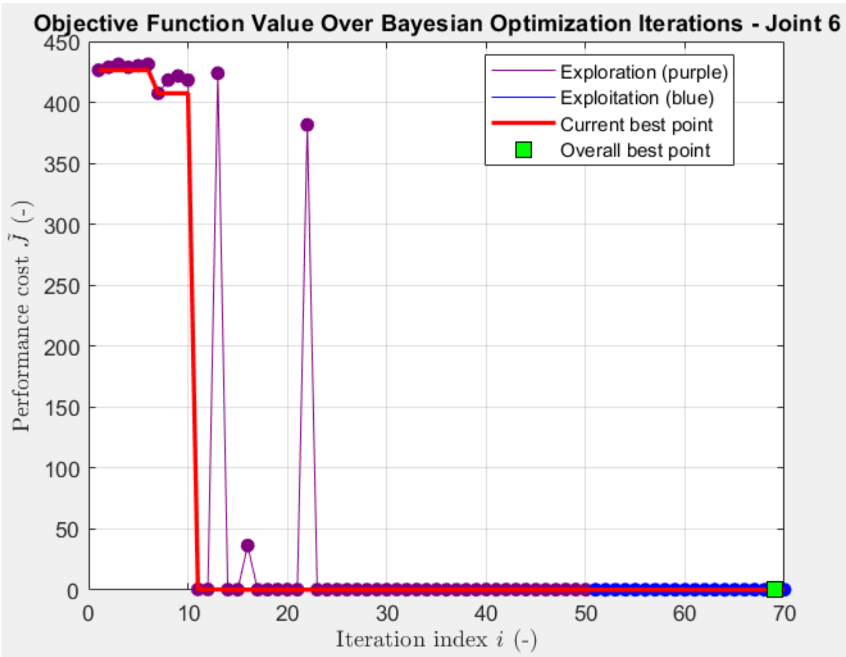

Figure 25: Objective function evaluations for joint 6 over iterations

Then, the position and the speed comparison between the reference and the measurement are shown:

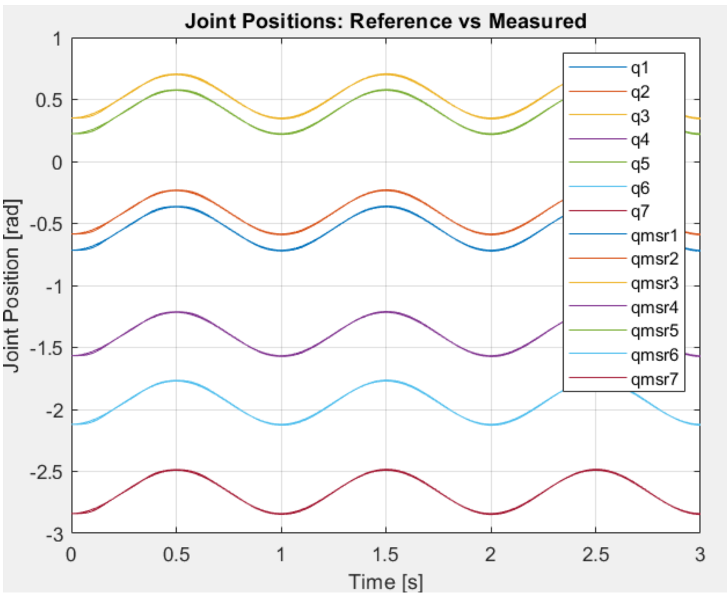

Figure 26: Reference position VS Measured position

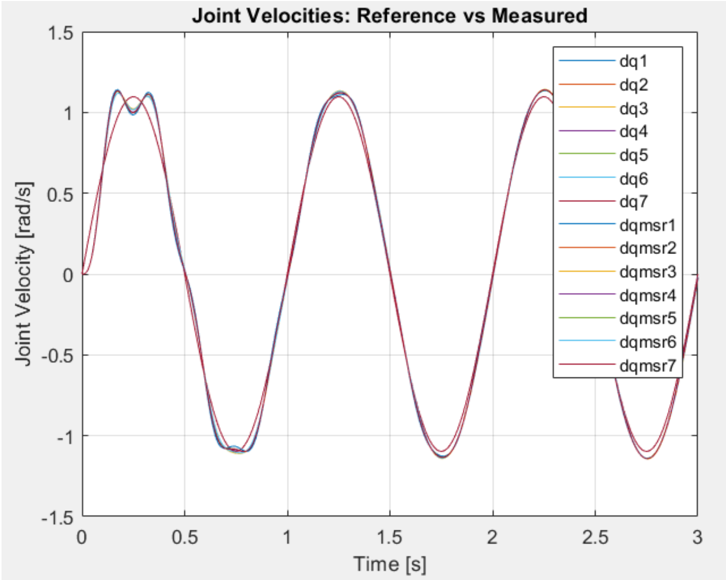

Figure 27: Reference speed VS Measured speed

Compared to the previous case, it is visible that now the difference between reference and measurement is slightly higher, due to the fact that lower K_p are used leading to worse performances. Accordingly, lower frequencies oscillations are present in both speed and torque.

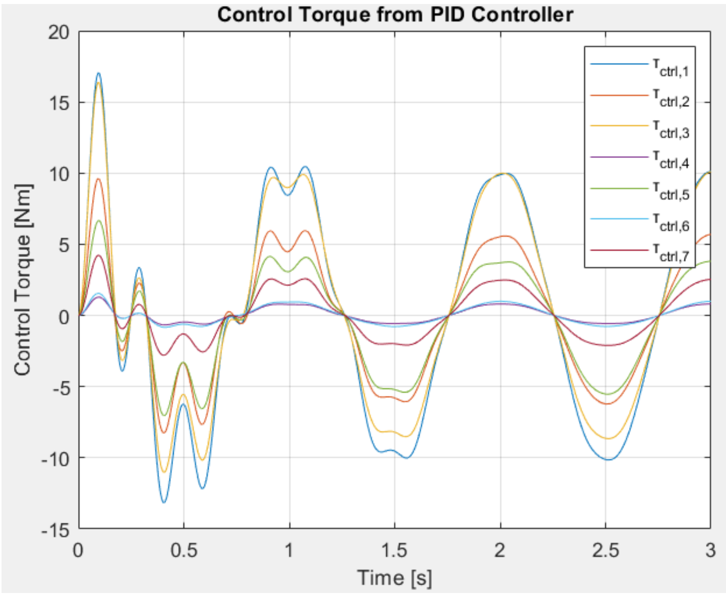

Figure 28: PID torque

Here the periodicity is evident in the torque profiles moreover, they are all in phase which follows expectations unlike the result for the gains optimized simultaneously seen in Figure 7.

K_{p1}	K_{p2}	K_{p3}	K_{p4}	K_{p5}	K_{p6}	K_{p7}
999.91	999.96	999.83	999.91	999.98	999.92	999.93

As expected we still have saturation of the proportional gains to their upper bound. Then, the results obtained are the following:

	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6	Joint7
K_p	999.91	999.96	999.83	999.91	999.98	999.92	999.93
K_i	545.66	69.64	483.14	311.17	25.66	52.07	547.30
K_d	3.47	3.72	5.46	4.14	3.71	4.84	4.18

Table 5: Values for the PID gain for each joint, $K_p \leq 1000$.

7 Conclusions

In this work, Bayesian Optimization was successfully applied to estimate the parameters of mechanical systems, with a particular focus on gravitational torque compensation and PID gain tuning. The comparison between gravitational torque and compensation torque across joints demonstrated that the physically informed approach yielded highly accurate results, achieving errors on the order of 10^{-4} Nm.

Regarding PID parameter estimation, the strategy of optimizing gains joint by joint proved effective. Two scenarios were investigated: with an upper bound on the proportional gain K_p set to 10000 and to 1000, respectively. In both cases, the optimization consistently drove the proportional gains to their maximum allowable values, highlighting the tendency of the algorithm to favor higher K_p values for improved tracking. However, this also resulted in increased oscillations and potential hardware concerns, as seen in the speed and torque responses.

By lowering the upper bound to 1000, a trade-off was achieved where tracking performance slightly degraded, but the system responses became smoother and more realistic for practical implementation. Overall, these results underscore the importance of carefully selecting constraints during optimization to balance performance and physical feasibility. The application of Bayesian Optimization demonstrated not only its capability to handle high-dimensional parameter spaces but also its flexibility in accommodating engineering constraints.

List of Figures

List of Tables