MASTER THESIS

EVALUATION OF MODEL PREDICTIVE CONTROL AND ADAPTIVE MODEL PREDICTIVE CONTROL FOR A SELECTED GENERALISABLE DAILY TASK INCLUDING UNCERTAINTIES

Elena París Quijada

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF BIOMECHANICAL ENGINEERING

EXAMINATION COMMITTEE
Dr. Edwin H.F. van Asseldonk
Dr. Ir. Mark Vlutters
Ir. Alex H.G. Overbeek
Dr. Hakan Koroglu

DOCUMENT NUMBER
BE - 972

UNIVERSITY OF TWENTE.

13-12-2023

CONTENTS

1 Introduction 3
1.1 Related Work . . . 4
1.2 Contributions . . . 5

2 Materials and Methods 5
2.1 Scenario Selection . . . 5
2.1.1 Top-Down Approach . . . 5
2.1.2 Bottom-Up Approach . . . 6
2.1.3 Outcome Scenario . . . 6
2.2 Control Method Selection . . . 6
2.2.1 Requirements . . . 7
2.3 Adaptive Model Predictive Control (AMPC) . . . 7
2.3.1 Optimisation Problem . . . 8
2.3.2 System Identification . . . 8
2.4 Simulations . . . 9
2.4.1 Experimental Setup . . . 9
2.4.2 Reaching Task . . . 9
2.4.3 Pulse Signal . . . 10
2.4.4 Task Without Predefined Path . . . 10
2.5 Robot Experiments . . . 11
2.5.1 Robot Parameters Identification . . . 11
2.5.2 Experiments Setup . . . 11

3 Results 11
3.1 Scenario Selection . . . 11
3.2 Control Method Selection . . . 13
3.3 Simulations . . . 13
3.3.1 Reaching Task . . . 13
3.3.2 Pulse Task . . . 14
3.3.3 Task Without Predefined Path . . . 15
3.4 Robot Experiments . . . 17

4 Conclusion 17

References 19

A First appendix 22

Evaluation of Model Predictive Control and Adaptive Model Predictive Control for a Selected Generalisable Daily Task Including Uncertainties

Elena París Quijada
Department of Biomechanical Engineering, University of Twente, The Netherlands

Abstract **- Robots need to interact with their surroundings, sometimes without having complete knowledge about it. In this research, different daily activities have been studied to find a general case that encompasses most of the situations. Subsequently, according to the selected scenario, Model Predictive Control and Adaptive Model Predictive Control methods are chosen and tested in simulation. The performance of the controllers is evaluated in different trials including model uncertainties and disturbances. Finally, the controllers are validated in a real case using a Franka Emika robot.**

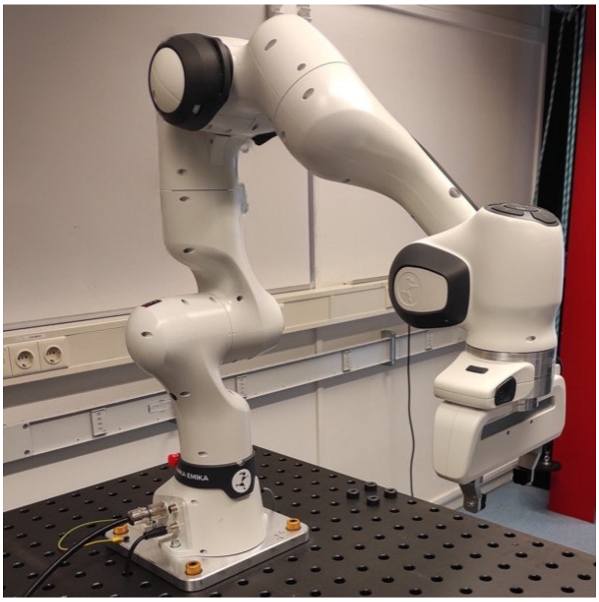

Figure 1.1: Franka Emika robot with 7 degrees of freedom in the chosen starting position. The robot is required to follow a rotational motion in the xy-plane, parallel to the mount.

1 Introduction

Robots have been widely used in industry during the past years. In healthcare, robots were first introduced for surgical assistance [1]. Lately, they have been included in numerous applications and healthcare settings such as rehabilitation or prosthetics. In particular, the lack of caregivers and the increase in healthcare costs have promoted the need for assistance robots [1, 2]. In the day-to-day work of nursing staff, back and shoulder injuries are common due to patient transfer and lifting. The urge to solve this problem has made assistive robots an interesting research field as their assistance could help with physically demanding duties [3]. In all these environments robots need to face a wide variety of situations and tasks which require interaction between the robot and its surroundings.

Disturbances and uncertainties are inevitable in many engineering systems due to measurement inaccuracy or modelling errors, e.g. when modelling a nonlinear system. This can affect both the performance and stability of the control systems [4, 5]. Thus, the dynamics of uncertain systems have long been and will continue to be one of the dominant themes in engineering applications [6].

Research has been done to develop control strategies for robots to adapt to different scenarios and situations. Some examples of used controllers are Disturbance-Observer-Based Control, Model Predictive Control, Optimal Control or Reinforcement Learning.

Some of these control strategies can be classified as model-based, which makes their performance highly dependent on system models and their predictions. Other strategies are model-free, meaning that a model of the environment is not required to determine the control law. Nevertheless, these control methods usually require a significant amount of data, which may not always be accessible. On the other hand, some control strategies allow the model to be updated at run-time, while others perform this step offline avoiding further modifications during the execution. Furthermore, not all control methods are robust enough against changes in the model or external disturbances [7].

The different properties of the control strategies can be considered as strong or weak points depending on the selected tasks or working requirements. This makes it challenging to determine the best approach for all kinds of scenarios.

The main aim of this research is to select and test a control method with the potential to be applicable across a wide range of scenarios where complete knowledge of the dynamic and kinematic properties is unavailable. An extensive evaluation of common daily tasks was conducted to get insight into the full spectrum of potential scenarios. Then, a case that incorporates as many of these likely scenarios as possible was selected. Subsequently, different control strategies were compared in order to identify the one that best aligns with the selected task and the specific requirements of the research. Finally, this research endeavours to establish the basis for future implementation of the controller in healthcare assistive robots.

1.1 Related Work

Different control strategies have been developed in order to deal with a wide range of scenarios and situations where interaction environment dynamics are not fully known. In these cases, typically a mathematical model of the system is needed. The strategies used to obtain these models from observed data are called system identification techniques [8] which include different algorithms.

Disturbance-Observer Based (DOB) is a control framework designed to estimate disturbances and/or uncertainties and then generate compensation in the control action using the computed estimates. It does not only account for disturbances from the external environment but also for uncertainties such as unmodeled dynamics and parameter perturbations. There exist variants such as Extended state observer which also estimates the states of a system [4, 5].

Sparse Identification of non-linear Dynamics (SINDy) are frameworks which identify non-linear dynamical systems from measurement data. These frameworks can be used when an imperfect model of the system is known as they identify the fewest terms in the model needed to explain the data. Even though they are not control frameworks, they can be combined with other algorithms, such as Model Predictive Control. Thus, the identification step can be performed fast enough to discover models in real-time [9, 10].

On the other hand, Adaptive Control estimates the unknown parameters of systems thanks to a parameter adaptation scheme. Then, the unknown parameters are replaced in a feedback controller with their estimates. Thus, the controller is adjusted for a system with parametric, structural and environmental uncertainties to achieve the desired system performance [11]. An example is Model Reference Adaptive Control where the controller adapts the system so it can behave as a certain reference model [12, 13].

Optimal control aims to find a control signal for a dynamic system that minimises a cost over a certain time interval. Optimisation of a system requires knowledge of a model of the controlled system, the performance criterion and constraints. There are different control techniques based on optimal control such as Reinforcement Learning, where the optimal behaviour depends on maximising a cumulative reward, or Model Predictive Control [14, 15].

Robust controllers are control strategies that guarantee an adequate level of performance within acceptable disturbance ranges or system parameter changes. There are different robust control strategies used in the development of controllers with stable and optimum response such as H-Infinity Control or Sliding Model Control. However, a poor tracking ability is usually shown [16, 17, 18].

Reinforcement Learning controllers are based on learning from experience while their behaviour is optimised based on some reward or punishment applied. This approach can work with systems where a complete knowledge of the model is unknown. Nevertheless, achieving the optimal solution require high computational resources [19, 20].

After comparing different control methods Model Predictive Control (MPC) and Adaptive MPC (AMPC) approaches have been selected. Regarding the adaptive strategies, Kalman Filter and Forgetting Factor approaches have been considered for the estimation of the model parameters [21]. Previous works have been done for the control of linear systems applying the mentioned controllers.

MPC describes a set of control methods that use a process model to predict the future behaviour of the controlled system. Thus, an optimal output is determined by solving a constrained optimisation problem. Typically, the system output is required to track a given reference for the prediction horizon. To some extend, it can handle with complex nonlinear systems. However, the performance can be highly dependent on accurate prediction models.

To avoid the dependency on the model, research focused on online model adaptation in MPC such as AMPC has been done [22, 23]. In particular, [24] developed an MPC controller with two adaptive schemes derived from online system identification and adaptive control to allow a manipulator to interact with unknown environments. Such control strategies were validated in door opening and object lifting tasks. However, it assumes that the environment can be described by a linear mass-spring-damper system, thus it does not consider nonlinearities.

In [25], an AMPC approach for a two-wheeled robot manipulator with varying mass is presented. The estimation of the model parameters is done by means of linear parameter varying modelling using a Kalman Filter algorithm. Other works have applied Kalman Filter for the estimation of the parameters in trajectory tracking using AMPC [26, 27]. Nevertheless, they have been only tested in specific cases of study without being extrapolated to other tasks.

Forgetting factor has been used together with AMPC approaches to improve tracking in quadrotors [28] or space robots [29]. Only the latest considers the presence of parameter uncertainties. In addition, the algorithm has been used to address the control of the joint trajectories of a robot considering nonlinear and time-varying dynamics [30]. It combines the classic MPC with a locally weighted learning approach.

In conclusion, there is relatively little work on AMPC approaches that can handle both linearities and nonlinearities in the system as well as uncertainties and disturbances.

1.2 Contributions

The main contribution of this work is to first, find a case of study from daily tasks that can be generalised to several different actions. Then, a control strategy able to control the selected tasks without the need to have an accurate model of the environment is determined and tested. This includes linear and nonlinear scenarios as well as different kinds of uncertainties and disturbances to represent interaction with unknown environments. In particular, the AMPC presented in this work combines the MPC strategy with parameter estimation to update the model depending on the scenario with a combination of algorithms that have not been addressed in previous research. Finally, the selected control strategy is tested in a real case using a Franka Emika robotic device. The procedure followed to answer the previous research questions is presented in Section 2 while the results for each of them are shown in Section 3.

2 Materials and Methods

2.1 Scenario Selection

There exists a wide variety of daily scenarios where the dynamical and kinematic properties of the environment are not completely known. In order to determine a specific case to study for this project, different approaches were considered.

2.1.1 Top-Down Approach

The first approach is called Top-Down. The main idea of this strategy is to determine common daily activities and reduce them to their

most basic components. First, a list of common daily tasks based on the literature was made [31, 32, 33, 34, 35]. Then, a second list was generated documenting the activities undertaken by a specific individual over a single day. Afterwards, a total of fifteen tasks were selected based on their recurrence in both lists. Thereafter, these tasks were divided into simpler actions for which the following properties were determined: dynamics, kinetics and when it is achieved. In addition, control methods and prior knowledge needed to perform the task were determined. These properties were selected to obtain a general idea of the composition of the actions. It is important to highlight that in this approach only the main control strategy that rules the task has been quantified. This main control strategy does not consider the prior knowledge or the goal condition but the simplest definition of the task.
Subsequently, the cumulative values of the different dynamics and control strategies were calculated to have a quantitative measure of the most common ones. Finally, the components with the highest outcome in the summation were selected. A summary of the main tasks with their related dynamics and control strategies is treated in Section 3 and can be seen in Table 3.1 while the table with the complete analysis can be found in Appendix A.

2.1.2 Bottom-Up Approach

The second strategy considered is named here as Bottom-Up. The central concept of this approach is to reconstruct the main task actions from the basic constituent elements based on the kinematics properties.
First, three out of the fifteen tasks determined in the Top-Down approach were selected. The selection process was based on the alignment between the tasks featured in the list determined in the Top-Down approach and the daily tasks mentioned in literature [31, 34]. The criteria for inclusion also focused on the ease of extrapolation or incorporation of the tasks into other tasks to ensure the selection of a more generalisable action.
Second, for each one of the tasks, different approaches to perform the tasks were considered based on the goal conditions and different prior knowledge. For each approach, initial and final kinematic states were defined in between the following options: force, position and velocity. Then, the increase or decrease of these conditions throughout the execution of the task was determined.
Third, a control strategy able to deal with the changes in the states was listed. Finally, as in the previous approach, the cumulative values of the control strategies were computed by counting their appearance in the tasks. A summary of the diverse states and control approaches considered is treated in Section 3 and is shown in Table 3.2 while the complete analysis can be found in Appendix A.

2.1.3 Outcome Scenario

Analysing the output of the two approaches, the most general case was obtained. The most common dynamics encountered in the tasks were mass and damping while the most common control paradigm was position control. Thus, the linear system shown in Equation 2.1 was considered. Where the value of the mass $m = 10\ kg$ and the friction $b = 10$ Ns/m. These values were selected based on similar studies [36].

$$F = m\ddot{x} + b\dot{x} \tag{2.1}$$

2.2 Control Method Selection

Considering the findings of the scenario selection, the main control goal is to make the mass of the aforementioned system follow a reference position. The controller will modify the value of the force exerted on the system so the generated position is as close as possible to the reference position.
To determine the control method needed to achieve the desired control goal, initially, a list of the main control strategies used in the control of robots according to literature was made [17, 18, 37, 38]. Subsequently, the strengths and limitations of each algorithm against the specific requirements of the research objectives were considered. These properties and studied controllers are listed in Table 2.1. It is important to note that the lines shown in some requirements indicate that these aspects cannot be ensured.

Table 2.1: Summary of the properties of the different control algorithms.

Properties	Control algorithm						
	Disturbance-Observer Based Control	SINDy-PI	Adaptive Control	Optimal Control	Model Predictive Control	Robust Control	Reinforcement Learning
Feedback	Yes	No	Yes	Yes	Yes	Yes	Yes
Model	Yes	Yes	Yes	Yes	Yes	Yes	Yes or No
Runtime model update	Yes	No	Yes	No	No	No	No
Non-linearities	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Online performance	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiple variables	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Stability	Yes	-	Yes	-	Yes	Yes	No
Computational efficiency	Medium	Low	Medium	Medium	Medium	Depends	Low
Disturbance rejection	Yes	No	Yes	No	Yes	Yes	No
Amount of data needed	Medium	Low	Low	Low	Low	Depends	High
Training	No	Yes	No	No	No	No	Yes
Generalize beyond training data	-	-	-	-	-	-	-
Combine with other control methods	Yes	Yes	Yes	Yes	Yes	Yes	Yes

2.2.1 Requirements

Some requirements arise from the outcomes of the scenario selection. These are the need to be able to deal with multiple variables and non-linearities and the presence of a model of the system. Regarding the requirement to have a model, this previous knowledge of the system does not need to be completely accurate. Thus, the controller is required to be able to update the model information.

Other properties are determined by the main aim of this research, which is to evaluate the performance when interacting with unknown environments. These include external disturbances or mismatches between the reality and the controller. Thus, disturbance rejection is considered a requirement. As shown in the literature [4], feedback controllers are designed so that the performance goals of systems can be achieved by attenuating disturbances. Thus, the presence of feedback or feedforward is considered a requirement.

On the other hand, as a future application, the controller is intended to be included in the healthcare environment. In this scenario, the controller needs to interact with the environment in real-time. This implies that the computations and refinement processes need to be done online. Furthermore, in real time not a big amount of data is always available. Therefore, control methods that require a small amount of data are preferred. To ensure optimal performance, a high computational efficiency is needed. Furthermore, for the safe inclusion of the controller in the healthcare environment stability is considered a requirement.

Finally, some controllers require a training phase which can be time-consuming and resource-intensive. This feature is impractical for the future implementation of the controller as it requires immediate and reliable performance.

After evaluating the mentioned controllers, both MPC and its extension with adaptive strategies, AMPC, have been selected.

2.3 Adaptive Model Predictive Control (AMPC)

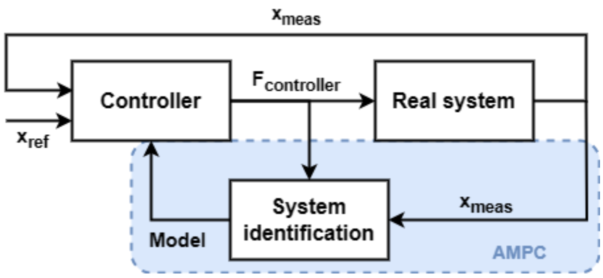

Figure 2.1: Control diagram for both controllers, MPC and AMPC. x_{ref} is the reference position, x_{meas} is the real position and $F_{contoller}$ is the force applied by the controller. Blue section corresponds to the identification step only present in the AMPC.

Figure 2.1 shows the control schema of both controllers. The signal x_{ref} represents the reference position, x_{meas} is the real position and $F_{contoller}$ is the force applied by the controller. Unlike MPC, AMPC uses the measured position and the force applied by the controller to estimate and update the parameters of the system model in real-time. This step is represented in blue.

2.3.1 Optimisation Problem

The MPC and the AMPC solve an optimisation problem, in this case, a quadratic program, to determine the values of the controlled variables. In this optimisation problem, a cost function is minimised. The cost function used is shown in Equation 2.2 [39].

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_\epsilon(z_k) \tag{2.2}$$

The term $J_y(z_k)$ corresponds to the tracking of the reference. It is defined in Equation 2.3 where w_i^y represents the weight associated with the reference tracking. The variables $y(k+i|k)$ and $r(k+i|k)$ are the predicted position value and the reference value at the ith prediction step while p is the prediction horizon.

$$J_y(z_k) = \sum_{i=1}^{p} \left\{ \frac{w_i^y}{s^y} \left[r(k+i|k) - y(k+i|k) \right] \right\}^2 \tag{2.3}$$

The term $J_u(z_k)$ refers to the controlled variable tracking and can be obtained as shown in Equation 2.4. Its associated weight is defined by w_i^u. To make sure that the controller does not generate force beyond the capacity of a physical device, this term includes penalisations for actions that require high forces. Finally, $u(k+i-1|k)$ corresponds to the value of the controlled variable in the control interval k and prediction step i.

$$J_u(z_k) = \sum_{i=0}^{p-1} \left\{ \frac{w_i^u}{s^u} \left[u(k+i|k) - u_{target}(k+i|k) \right] \right\}^2 \tag{2.4}$$

The variable $J_{\Delta u}(z_k)$ corresponds to the controlled variable change suppression. It can be computed as in Equation 2.5 where $w_i^{\Delta u}$ is the associated weight.

$$J_{\Delta u}(z_k) = \sum_{i=0}^{p-1} \left\{ \frac{w_i^{\Delta u}}{s^u} \left[u(k+i-1|k) - u(k+i|k) \right] \right\}^2 \tag{2.5}$$

Finally, $J_\epsilon(z_k)$ refers to the constraint violation for the current control interval, k where z_k represents the controlled variable adjustments that minimise the cost function.

After conducting a systematic exploration of various weight configurations and aligning with the recommendations proposed in [39] the default weights assigned to w_i^y, w_i^u, and $w_i^{\Delta u}$ in Equation 2.5 were updated. The final values are shown in Table 2.2.

Regarding the constraints, they are determined by the physical limits of the robot. Specifically, to restrict the exerted force to within acceptable limits, constraints on the controlled variable have been introduced according to Equation 2.6. Variables u_{min} and u_{max} are the lower and upper bounds while s^u refers to the scale factor for the controlled variable. The values are shown in Table 2.2 and were selected following the recommendations of [39].

$$\frac{u_{min}}{s^u} < \frac{u(k+i-1|k)}{s^u} < \frac{u_{max}}{s^u}, i = 1:p \tag{2.6}$$

where p is the prediction horizon.

Table 2.2: Optimisation weights

Variable	**Weight**
w_i^y	$1\ unitless$
w_i^u	$10\ unitless$
$w_i^{\Delta u}$	$1\ unitless$
u_{min}	$-100\ N$
u_{max}	$100\ N$
s^u	$200\ N$

2.3.2 System Identification

The identification step is made using recursive polynomial model estimation. Based on the input of the system and the measured output, the discrete-time input-output polynomial model of the system is estimated. In this case, the ARX model shown in Equation 2.7 is selected as it can handle systems with multiple inputs [21]. In addition, this model assumes that the current system output is a function of the previous system outputs and inputs. Variables $u(t)$ and $y(t)$ correspond to the input and the output of the system, force and position respectively. Variable q refers to the time-shift operator, n_k

corresponds to the input delay and $e(t)$ is the error. Finally, $A(q)$ and $B(q)$ are the polynomials whose parameters will be estimated.

$$A(q)y(t) = B(q)u(t - n_k) + e(t) \quad (2.7)$$

A common algorithm used in parameter estimation is Kalman Filter (KF). However, in [10] limitations in capturing nonlinear dynamics or structural changes are mentioned. Thus, not only Kalman Filter but also Forgetting Factor (FF) algorithm will be used in the parameter estimation process [21].

2.4 Simulations

2.4.1 Experimental Setup

The selected controllers are tested in simulation using both the Matlab and Simulink environments with the Model Predictive Control Toolbox. The configuration of the controller has been determined through a systematic process involving iterative experimentation and refinement. Thus, the values that ensure a better tracking of the reference were selected. The final settings are shown in Table 2.3. The system is composed of two inputs, force and disturbance force; and one output, position. According to the research requirements, unmeasured external disturbances have been considered.
For the identification of the parameters, the System Identification Toolbox from Matlab was used. The values for the process noise covariance for the Kalman Filter algorithm and the Forgetting Factor parameter need to be determined. These values, shown in Table 2.3 were selected after a process of iterative experimentation and refinement. The values that minimised the reference position tracking error were chosen.

Table 2.3: Controller settings

Settings	**Value**
Sampling time	$0.005\ s$
Prediction horizon	$100\ steps$
Control horizon	$5\ steps$
Forgetting factor	$1 - 5 * 10^{-3}\ unitless$
Process Noise Covariance	$0.01\ unitless$

The control methods are tested in three different simulated scenarios. These included trials with unknown mass, damping, components of the system model, external disturbances and nonlinear components. A summary of all the different tests can be seen in Table 2.4. The values of the mass and damping were selected based on the values used in similar studies [36]. Furthermore, they align with the expected application. When applying the controller to transport a patient in a wheelchair, it is essential to consider that adults can weigh between 40 to 120 kg. If the controller is only aware of the weight of the wheelchair, which on average is 10 kg, the total mass the controller has to interact with can be significantly larger, up to ten times greater. The reference values considered were $m = 10\ kg$ and $b = 10$ Ns/m.
To compare the behaviour and performance of the controllers for the different scenarios the root mean square of the tracking error (RMSE), rise time, settling time and overshoot are computed. The rise time was considered as the time the response takes to rise from 10% to 90% of the final value. Regarding the settling time, the selected threshold was 1%. In addition, the variation of the mentioned outcomes relative to the reference values, where no mismatches are considered, is computed. Then, the mean and standard deviation (SD) of these variations are calculated. Finally, to investigate the difference in the behaviour between the FF and KF approaches, the transfer functions based on the estimated parameters along the simulations are obtained.

2.4.2 Reaching Task

In the first simulation, named reaching task, the controller is required to follow the reference step signal of magnitude $10\ m$ applied at $t = 10$ s. A graphical representation of the motion is shown in 2.2. First, only different values of the modelled mass, m_{cp}, were considered while keeping the real mass, m_{rp}, constant.
Afterwards, the adaptability to external disturbances was tested. Two different disturbances were applied. The first one consisted of a constant force and the second one was an impulse. The latest was applied at $t = 15.5\ s$. Both were applied as positive and negative disturbances with magnitudes $30\ N$ and $-30\ N$ respectively.

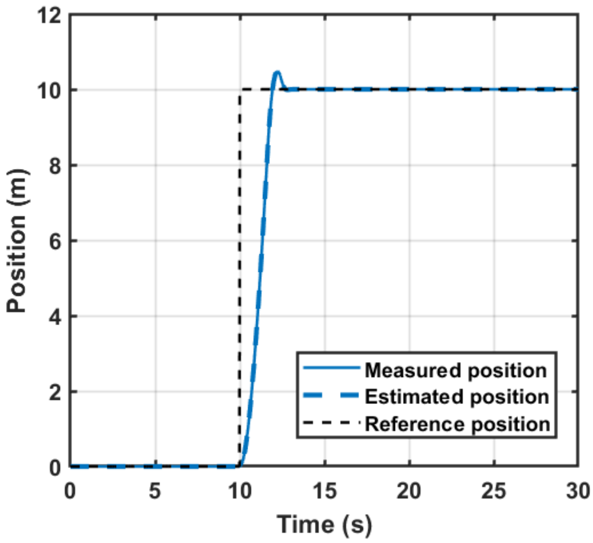

Figure 2.2: Overview of the reaching task for the MPC with the reference values.

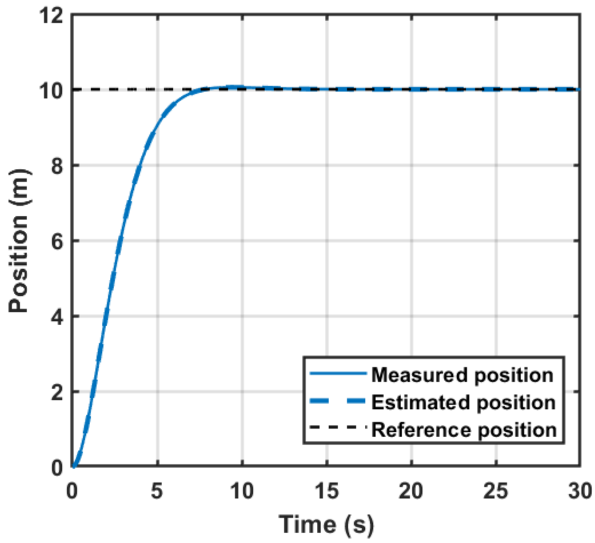

Figure 2.3: Overview of the task without predefined path for the MPC with the reference values.

2.4.3 Pulse Signal

For the second trial, the controller is required to follow a reference pulse signal of magnitude $10\ m$. The pulse starts at $t = 10\ s$ and ends at $t = 20\ s$. An example of the described motion can be seen in Figure 3.2. In addition to the scenarios with different masses, mismatches between the damping were tested modifying the value of the modelled damping, b_{cp}, while keeping the real value, b_{rp}, constant. Furthermore, model mismatches were introduced. The real system remained a mass-damper system, r_{IB}, while the modelled system was configured to operate under two different assumptions: one considering only a mass, c_I, and the other assuming a mass-spring-damper system, c_{IBK}.

2.4.4 Task Without Predefined Path

When humans perform object movement tasks, they do not prioritise following an exact path required to achieve the desired goal but a motion given by models such as metabolic energy minimization, minimum jerk or minimum torque change [40]. To emulate this behaviour, the controller is tasked with reaching and maintaining the goal position, $x = 10\ m$. Nevertheless, it is no longer provided with an explicit path to follow at each time step. Figure 2.3 shows a graphical description of the motion.
As in previous scenarios, the goal was tested for different values of the mass, damping and components in the system. Furthermore, nonlinearities in the damping were included through the displacement square damping model. The force generated due to friction can be calculated as shown in Equation 2.8, [41]. For these simulations, the values of mass and damping were $m = 10\ kg$ and $b = 10\ Ns/m$.

$$F_b = (sgn\dot{x}) * b * x^2 \tag{2.8}$$

Finally, simulations were performed varying the reference values of the environment mass and the damping at runtime. The parameters reached the lowest value at $t = 12\ s$.

Table 2.4: Summary of all the tested scenarios.

	Configuration	
	Controller	**Environment**
Unknown mass	m_{cp} (kg)	m_{rp} (kg)
	$1, 2, 5, 10,$ $20, 50, 100$	10
Unknown damping	b_{cp} (Ns/m)	b_{cp} (Ns/m)
	$0.1, 1, 2, 5, 10,$ $20, 50, 100$	10
Change in controller components		
cp_I: m_{cp} (kg)	10	$m_{cp} = 10kg$
$cp_{IBK}: m_{cp}, b_{cp}, k_{cp}$	$10kg, 10\frac{Ns}{m}, 10\frac{N}{m}$	$b_{cp} = 10\frac{Ns}{m}$
Nonlinear dynamics		
	Nonlinear b_{cp}	Linear b_{rp}
	Nonlinear b_{cp}	Nonlinear b_{rp}
	Linear b_{cp}	Nonlinear b_{rp}
Changing dynamics		
m (kg)	10 1	10 to 1
b (Ns/m)	10 0.1	10 to 0.1

2.5 Robot Experiments

Finally, the selected controllers were tested in a real setup. The robot employed is Franka Emika composed of a 7-degree of freedom (DOF) arm with a gripper as end-effector. The system is shown in Figure 1.1. It is programmed using the Simulink library provided by Franka, into which both the MPC and AMPC codes can be easily incorporated.

2.5.1 Robot Parameters Identification

To perform the experiments, the values of the parameters of the robot were identified. These parameters were only identified for the specific motion determined for the experiment. This motion was restricted to the xy-plane, parallel to the mount of the robot, and was generated by one of the joints keeping the other six joints fixed.

The robot was excited with a chirp signal in a range of frequencies from $0.01\ Hz$ up to $5\ Hz$. This signal was provided as the torque. The generated motion, velocity, and acceleration, due to that torque were measured. The frequency range was selected considering that, for greater frequencies, there was no motion associated with the input signal.

Finally, the values of the robot's mass and damping were obtained by applying least squares regression as shown in Equation 2.9. Vector $\vec{\tau}$ includes all the values of the torque and matrix X includes the values of the position, velocity and acceleration of the robot end-effector. Finally, vector $\vec{\beta}$ corresponds to the parameters that want to be estimated this is, the value of the mass, damping and spring constant.

$$\vec{\tau} = X * \vec{\beta} \tag{2.9}$$

2.5.2 Experiments Setup

The robot is programmed to follow a rotational motion of 60º in the xy-plane, parallel to the mount, from the chosen initial position shown in Figure 1.1. These values were selected to perform motion in a large range without exceeding the limits of the robot. Only the joint used for the identification process was involved in the motion.

In this case, only the MPC and AMPC FF were tested due to the better performance of the AMPC FF compared to the AMPC KF in the simulations. Three scenarios were tested by varying the controller mass considering it to be greater, smaller or equal to the real robot mass. Furthermore, a limit in the force applied by the controller was set in $-10\ N$ and $10\ N$ to ensure the integrity of the robot.

In addition, to avoid discontinuities in the torque, its range of variation was limited for the AMPC. This was decided instead of changing the weight associated with the variation in the controlled variable since the limitation established within the cost function is not a hard constraint. Thus, to get variations in the torque that the robot can accept the weight needs to be set excessively high, around 50, while the weight for the reference tracking needs to be exceptionally low, around 1. This worsens the performance of the controller as the reference is not followed anymore.

3 Results

3.1 Scenario Selection

The results of the Top-Down approach are shown in Table 3.1. The dynamics that obtained the highest value in the summation and therefore, the most common within all the tasks, are mass and damping. In addition, this damping can be linear or nonlinear. It is important to highlight that springs and dampers are components with a similar number of appearances in the tasks. However, only damping was considered as according to the computed cumulative values it appears in more actions than spring. Furthermore, in real systems, the motion of a mass is accompanied by some kind of damping. On the other hand, position control is the main control strategy as it exhibited the greatest cumulative value. Finally, most of the tasks require some prior knowledge which leads to the necessity for the controller to have a model of the system. However, it can be incomplete or wrong.

Regarding the Bottom-Up approach, the results are presented in Table 3.2. The most

Table 3.1: Summary of the main daily tasks with their related dynamics and control strategies of the Top-Down approach.

Case of study	Dynamics			Control			
	Mass	Damper	Spring	Impedance	Position	Force	Velocity
Open a lid	x	x	-	x	-	-	-
Close a lid	x	x	-	x	-	-	-
Open the tap	x	x	-	x	-	-	-
Prepare breakfast	x	-	x	-	x	x	-
Take notes	x	-	-	-	x	x	-
Open a window/door	x	x	-	x	-	-	-
Close a window/door	x	x	-	x	-	-	-
Throw out rubbish	x	-	-	x	-	-	-
Hand out cards	x	-	-	-	x	-	-
Answer a call	-	-	x	-	x	-	-
Eat a piece of steak	x	-	-	-	x	-	x
Cut a steak	x	x	x	x	-	-	x
Wash the dishes	x	-	x	-	x	-	-
Brush the teeth	x	x	x	-	x	-	-
Vacuum	x	x	-	-	x	-	-
Total	14	8	5	7	8	2	2

Table 3.2: Summary of the kinematic states and control methods of the three selected daily tasks for the Bottom-Up approach.

Case of study	Subactions	State variation	Considerations	Control strategy			
				Impedance	Position	Force	Velocity
1) Open a lid		$f_i > f_f$,$v_i < v_f$	-	x	-	-	-
		$f_i > f_f$, $x_i < x_f$	-	-	x	x	-
		$x_i < x_f$	The force needed to move the lid is known	-	x	-	-
2) Prepare breakfast	Open cupboard	-	Same as 3)				
	Take out mug	$x_i < x_f$	-	-	x	-	-
	Open fridge	-	Same as 3)				
	Take out milk	$x_i < x_f$	-	-	x	-	-
	Open milk bottle	-	Same as 1)				
	Pour milk	$x_i < x_f$	-	-	x	-	-
		$f_i < f_f$,$v_i > v_f$	-	x	-	-	-
	Open microwave	$f_i < f_f$,$x_i < x_f$	-	-	x	x	-
		$v_i > v_f$	Apply velocity to the button until the end of the motion range is reached	-	-	-	x
	Put mug in the microwave	$x_i < x_f$	-	-	x	-	-
3) Open a window/door		$f_i > f_f$,$v_i < v_f$	-	x	-	-	-
		$f_i > f_f$, $x_i < x_f$	-	-	x	x	-
		$x_i < x_f$	The force needed to move the object is known	-	x	-	-
		$f_i < f_f$,$v_i > v_f$	Task is completed when the object is completely open	x	-	-	-

quantified control strategy is position control which confirms the findings obtained in the Top-Down approach. In addition, almost all the tasks can be explained using combinations of different kinematic states. That determines the need for the control framework to deal with multiple specified goals.

Based on the previous findings similar tasks, such as opening a door, drawer, lid, window or jar include the most common dynamics and control strategies. Thus, instead of a specific task, the system described in Equation 2.1, which can be generalised in all the mentioned actions, was selected. Furthermore, by the nature of these tasks, both rotational and linear motion are involved. Linear motion is tested in the simulations due to its ease of implementation reducing the complexity of the simulation environment. On the other hand, rotational motion has been selected for performing the experiments on the robot.

3.2 Control Method Selection

Diverse control algorithms have similar properties and performance as depicted in Table 2.1. Regarding optimal control, it lacks some of the most important properties such as disturbance rejection or update of the model at runtime. Thus, it is not considered a suitable option.
In addition, robust control does not allow the update of the model at run time and it has been reported to not be an approach commonly used in industry [38] while MPC is.
Finally, MPC itself lacks the capability to update the internal system model. However, it offers sufficient flexibility to accommodate an additional control method with the specific function of model updating. In this context, both DOB and Adaptive Control can be combined with MPC. The resulting controller is able to update the system model and the performance of the MPC when used on its own is improved.
According to Table 2.1 the main difference between DOB and Adaptive control is that DOB needs a bigger amount of data to perform the estimations. Furthermore, the efficacy of DOB to deal with changes in mass and damping has been already tested as mentioned in [36] which leaves AMPC as a promising field of study. Additionally, implementations of Adaptive Control within MPC are already available in programming environments such as Matlab. All this leads to the selection of MPC and its extension with adaptive strategies, AMPC, as control methods.

3.3 Simulations

3.3.1 Reaching Task

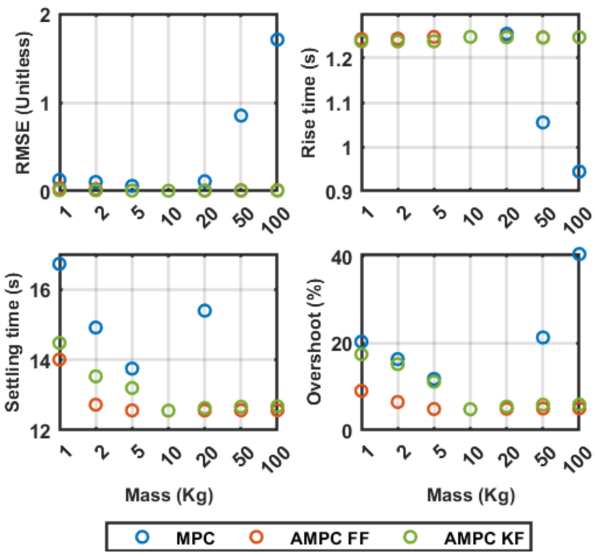

Figure 3.1: RMSE, rise time, settling time and overshoot for different values of the controller mass for the MPC, AMPC with FF and AMPC with KF controllers. Real mass is $m_{rp} = 10\ kg$.

The results for the trials with different values of the mass can be seen in Figure 3.1. MPC shows the biggest variation of the variables relative to the values obtained for $m = 10\ Kg$, which is the reference. This difference is especially large for the trials where the controller mass is bigger than the real mass. It is important to highlight that, in these cases, the scenario becomes unstable for the MPC. These instabilities arise from the fact that the controller considers the object to be heavier than it is. Thus, the controller applies a great amount of force to reach the target position which leads to overshooting. The repeated attempts to correct this mismatch with the reference create a cycle of instability. As it does not reach the desired end position, there is no settling time associated with these trials.

Table 3.3: RMSE, rise time, settling time and overshoot when applying external disturbances for the MPC, AMPC with FF and AMPC with KF controllers. Cases where the desired end position was not reached are marked with lines in the overshoot.

External disturbance	**RMSE**			**Rise time (s)**			**Settling time (s)**			**Overshoot (%)**		
	MPC	FF	KF	MPC	FF	KF	MPC	FF	KF	MPC	FF	KF
Positive Impulse	0.015	0.015	0.015	1.248	1.248	1.248	16.275	16.274	16.273	4.784	4.799	4.812
Negative Impulse	0.014	0.015	0.015	1.248	1.248	1.248	16.275	16.274	16.273	4.863	4.881	4.894
Positive Constant Force	0.082	0.083	0.082	1.550	1.550	1.550	12.503	12.503	13.060	-	-	-
Negative Constant Force	0.084	0.084	0.084	1.085	1.085	1.085	13.037	13.041	13.038	15.392	15.532	15.431

Regarding the AMPC with FF and AMPC with KF, both show similar behaviour between each other and relative to the reference. The AMPC with FF exhibits a closer performance to the reference in the settling time and overshoot. In addition, for each controller, it was computed the average duration during which the maximum force was applied. The calculated values are as follows: $7.401\ s$ for the MPC, $2.3954\ s$ AMPC FF, and $2.8374\ s$ for the AMPC KF.

The results of the simulations where external disturbances were applied are summarised in Table 3.3. It is observed that all controllers exhibited a similar performance. Remarkably, for the trial where a positive constant force was applied, none of the implemented control strategies reached the desired end position. In that case, the error between the final position and the desired reference position was consistently measured as $0.072\ m$, $0.073\ m$ and $0.073\ m$ for the MPC, the AMPC FF, and the AMPC KF, respectively.

3.3.2 Pulse Task

First, it is important to highlight that AMPC with KF algorithm presented unexpected behaviour in some of the trials. The controller was expected to show resembling behaviours both at the rise and the fall of the pulse signal. Furthermore, considering a proper adaptation of the algorithm the performance at the end of the pulse is expected to be improved as it happens in the results obtained for AMPC FF. However, as illustrated in Figure 3.2, for AMPC KF the performance does not improve but deteriorates.

The mean and SD of the RMSE variation relative to the reference values are computed. The outcomes for all the controllers and scenarios are depicted in Table 3.4. AMPC with FF shows the smallest values of the mean and SD while MPC presents notable higher values.

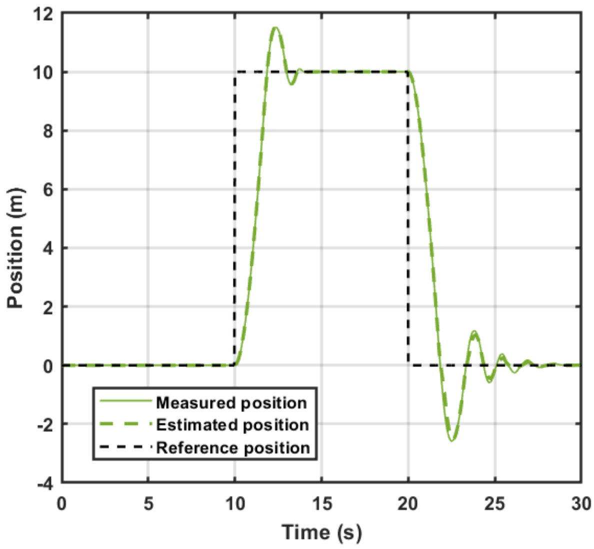

Figure 3.2: Measured, estimated and reference position of the AMPC KF controller for $m_{cp} = 2\ kg$.

Table 3.4: Mean and SD of the RMSE variation for MPC, AMPC FF and AMPC KF controllers, for different values of the mass, damping and controller system components.

	Varying Mass		**Varying Damping**		**Varying Components**	
	Mean	SD	Mean	SD	Mean	SD
MPC	102.40	130.27	74.43	112.96	69.60	75.24
AMPC FF	2.10	1.90	2.29	2.57	0.70	0.99
AMPC KF	5.50	8.20	4.97	0.40	8.03	0.00

On the other hand, the values of the overshoot are represented in the boxplot shown in Figure 3.3. The first subplot corresponds to the trials involving variations in the system mass. Subplot (b) represents the simulations with variations in the value of damping. The final subplot is associated with the model mismatches. For the AMPC with FF, the overshoot values remain at approximately the same level across all scenarios.

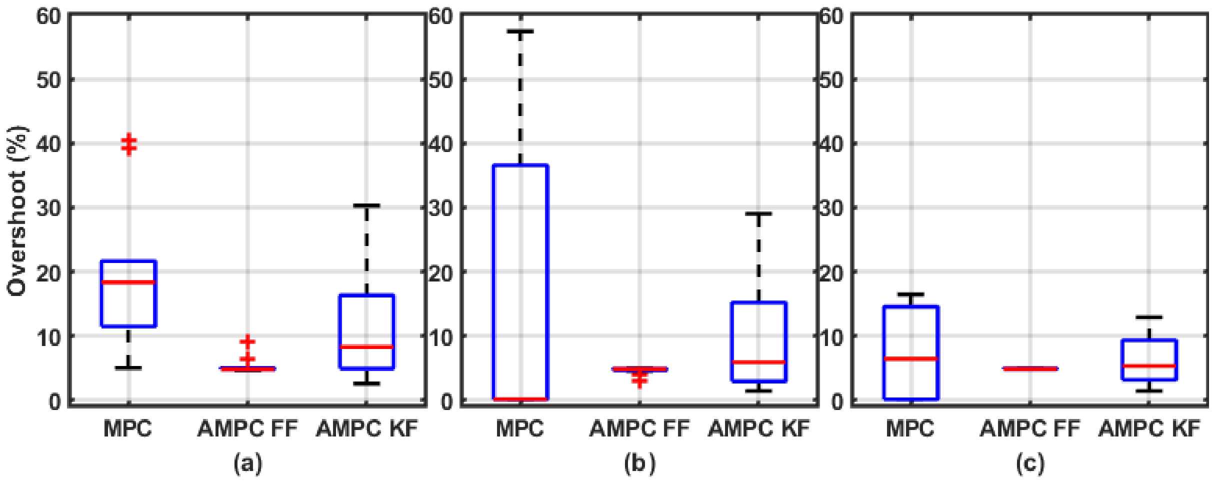

Figure 3.3: Overshoot (%) for the trials involving variation in a) the system mass, b) the damping of the system, and c) amount of system components for the MPC and AMPC FF controllers.

On the other hand, these values change significantly for the MPC and AMPC KF. It is important to note that, the output of the MPC became unstable in the trials $m_{cp} = 50\ kg$, $m_{cp} = 100\ kg$, $b_{cp} = 50$ Ns/m, $b_{cp} = 100$ Ns/m and cp_{IBK}.

3.3.3 Task Without Predefined Path

Table 3.5: Mean and SD of the RMSE, rise time, settling time and overshoot variation for MPC, AMPC FF and AMPC KF controllers, for different values of the mass and damping.

		MPC		AMPC FF		AMPC KF	
		Mean	SD	Mean	SD	Mean	SD
Varying mass	RMSE	14.83	10.91	8.25	4.08	3.08	6.82
	Rise time	0.69	2.05	-0.008	0.19	-0.40	0.15
	Settling time	0.48	0.65	0.27	0.23	0.72	0.75
	Overshoot	12.59	17.63	0.11	1.01	15.45	7.50
Varying damping	RMSE	32.50	33.73	8.36	3.65	0.36	0.24
	Rise time	0.28	0.58	0.027	0.21	-0.40	0.14
	Settling time	0.46	0.88	0.44	0.33	1.01	0.99
	Overshoot	-0.19	0.67	-0.25	0.82	17.50	8.83

For the trials with different values of mass and damping, the mean and SD of the RMSE, rise time, settling time and overshoot variations are shown in Table 3.5. MPC demonstrates the poorest performance as all the outcome parameters highly differ from the reference. In addition, the goal was not reached for the scenarios where $m_{cp} = 50\ kg$ and $m_{cp} = 100\ kg$. Regarding AMPC, for different mass values, the values of the RMSE and rise time are slightly bigger for AMPC with FF than for AMPC with KF. However, the settling time and overshoot present a greater difference between trials for the AMPC with KF than for the AMPC with FF. The same outcomes arise for the simulations with different damping values.

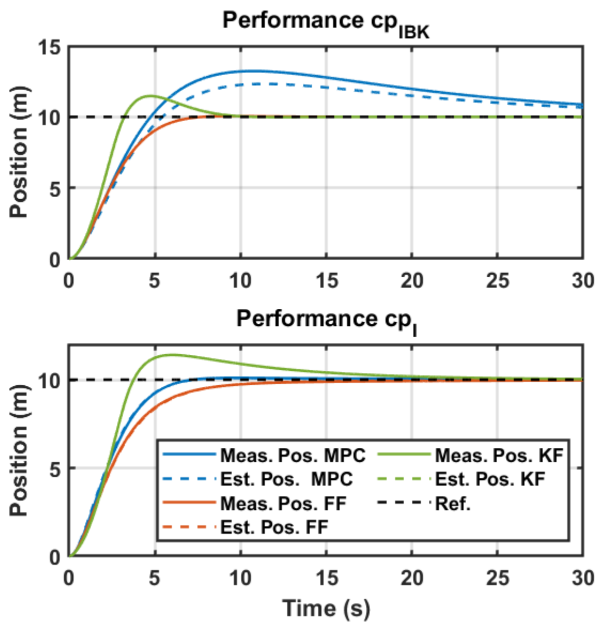

Figure 3.4: MPC, AMPC FF, and AMPC KF performance with different controller system components compared to the real mass-damper system.

The results of the model variations are summarised in Figure 3.4 where the measured and estimated position of all the controllers are shown. For the cp_{IBK}, the MPC presents a great error between the measured and the estimated position, 0.643, apart from never reaching the goal due to the overshoot, 21.717%. Regarding AMPC, both controllers, with KF and FF exhibit a similar performance in the two cases. However, the behaviour of the AMPC with FF is closer to the reference situation, Figure 2.3, where no overshoot occurs.

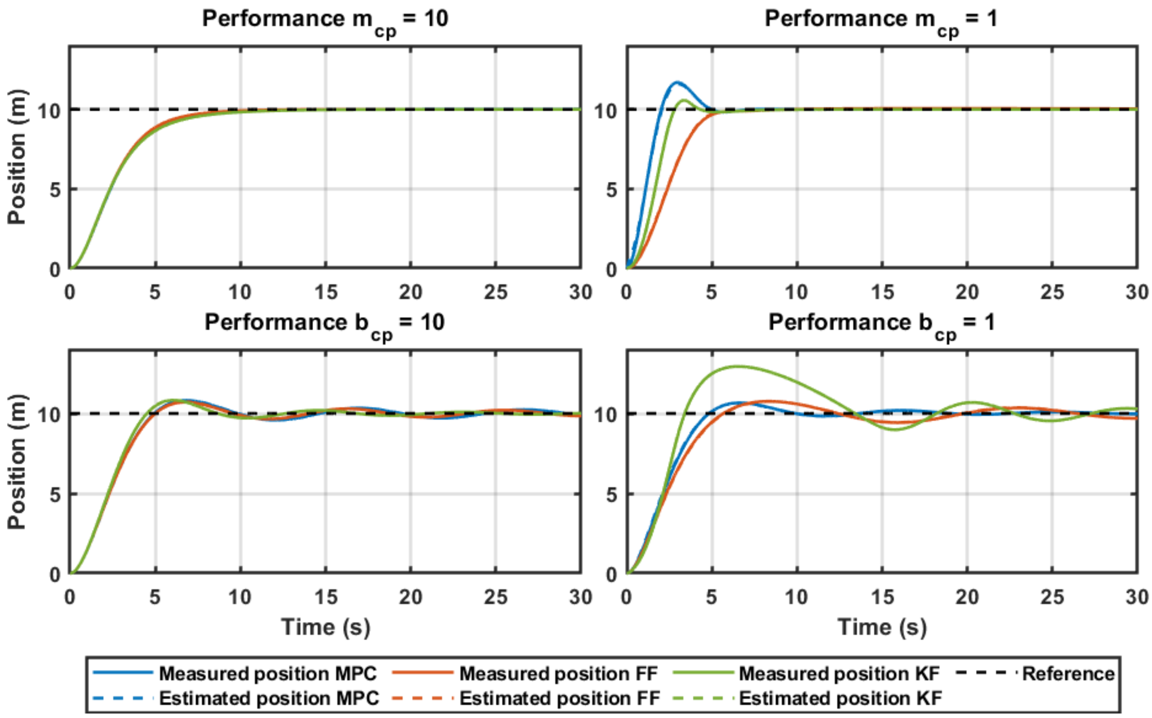

Figure 3.5: Measured, estimated and reference position for the trials where variations of mass and damping at runtime were added for MPC, AMPC with FF and AMPC with KF controllers.

Figure 3.5 depicts the measured, estimated and reference positions for the trials where changing dynamics were considered. AMPC with FF presents a more consistent performance along the four different trials. Regarding the rest of the controllers, MPC exhibits a behaviour more consistent than AMPC KF for the changing damping trials. However, it happens the opposite in the changing mass scenarios.

For the simulations where the controller considers nonlinear damping while the real damping is linear, all the controllers reached the goal. The values of the RMSE are 0.036, 0.024 and 0.003 for the MPC, AMPC FF and AMPC KF respectively. Although AMPC KF presents the smallest value of the RMSE, it is the only strategy that shows overshoot, 21.783%. This overshoot is not present in the simulations where both real and controller damping are linear, Figure 2.3. For trials with nonlinear damping in both, the controller and the real system, none of the controllers reached the goal. The difference between the final and reference position has been computed. These error values are consistent between the controllers being $0.346\ m$ for the MPC, $0.346\ m$ for the AMPC with FF and $0.352\ m$ for the AMPC KF. In addition, when only the real system considers nonlinear damping none of the controllers reached the goal. In this case, the error values are $0.346\ m$, $0.346\ m$ and $0.353\ m$ for the MPC, AMPC FF and AMPC KF respectively.

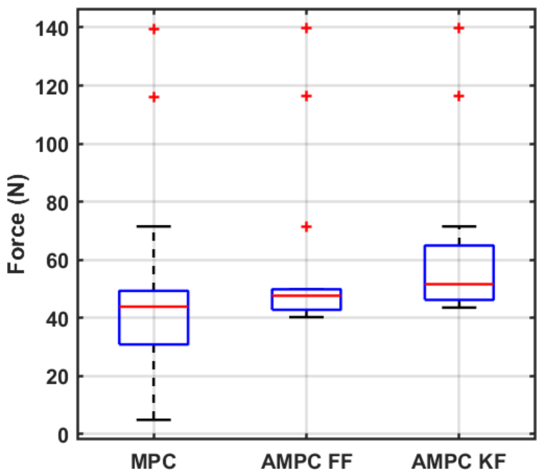

Figure 3.6: Maximum force applied by the controllers in all trials without predefined path.

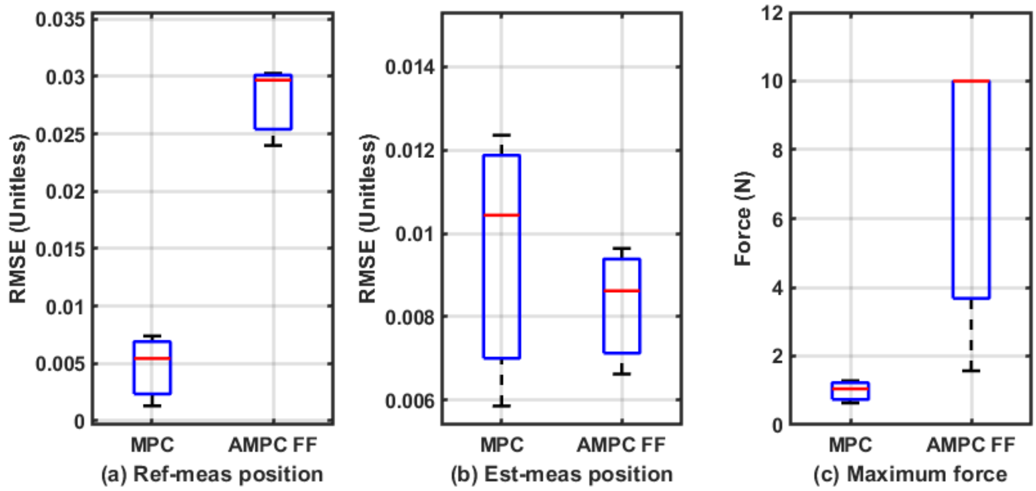

Figure 3.7: a) RMSE between the reference and measured position in all trials; b) RMSE between the estimated and the measured position in all trials; c) Maximum applied force in all trials.

The mean of the maximum force applied by the controllers in all the mentioned scenarios is summarised in Figure 3.6. AMPC with FF presents the lowest variability in the maximum force. In addition, its mean value is slightly bigger than the one of the MPC but lower than the AMPC KF.

Finally, the computed transfer functions based on the estimated parameters for the FF and KF approaches show that FF always estimates a mass-damper system. This is not always the case for the KF approach which sometimes estimates a mass-spring-damper system. Nevertheless, none of the methods converge to the real values of the mass and/or damping.

3.4 Robot Experiments

From the linear squares regression, the values of the mass and damping of the robot in the chosen motion were determined. The value of the mass is $2.5\ kg$ while the damping is $8.011\ Ns/m$. These values were considered the reference values.

The results of the experiments performed with the robot can be seen in Figure 3.7. First, the RMSE between the reference and the measured position has been calculated. AMPC with FF presents greater error than MPC following the reference position. Second, the RMSE between the estimated position by the controller and the measured position is shown. In this case, AMPC with FF has a smaller error relative to the actual position. Finally, the maximum force applied in all the trials is summarised. AMPC applies greater force in all the trials than MPC. As mentioned before, for the AMPC the change in the force was limited. This limitation has been included in the Simulink model but not as part of the configuration of the AMPC. Therefore, the controller is not aware that the force it is supposed to apply is being limited. As the torque cannot increase as fast as the controller is expecting, its capability to track the reference position more accurately is affected.

4 Conclusion

The goal of this study was to find a case of study from daily tasks that can be generalised to several different actions. Then, select a control strategy able to accurately perform the selected task without the need to have perfect knowledge about the system it is interacting with. Finally, the performance of the controller would be evaluated under different scenarios both, in simulation and a real case using a Franka Emika robotic device.

First, from the selection of the scenario, it was concluded that several common daily tasks with different goals are conformed by the same dynamics, and kinematics. In addition, they can be controlled by the same kind of control strategies. As a result of this, a simple scenario that can easily be extrapolated to different actions could be selected.

Second, based on the requirements of the selected case of study, MPC controller and its extension with AMPC using both KF and FF approaches for the parameter estimation were selected for their evaluation.

Regarding the simulations, the goal was reached in almost all the trials for AMPC approach but not for MPC, such as the cases considering mass values of 50 or 100 kg. Furthermore, MPC showed unstable performances for bigger values of the mass or for some trials of the pulse signal which cannot be accepted in the real application. In addition, the greater values of the RMSE obtained for the MPC compared to the ones of the AMPC show that the controller is not able to accurately follow the desired trajectory. Thus, it is demonstrated that the combination of MPC with adaptive strategies enables the controller to complete the desired tasks when uncertainties in the system model are present while the MPC approach on its own fails.

In particular, AMPC with FF shows a better performance than the KF approach. This is demonstrated as presents a more consistent performance between the trials. Furthermore, it also shows better adaptability as its behaviour is closer to the reference. This aligns with the statements found in the literature [10]. This difference in the performance is caused by the estimation algorithms used. The lack of convergence to the real values for both approaches confirms the fact that the algorithms compute the parameters that better relate the input and the output without caring about the actual values or composition of the system.

The lack of difference between the controllers in the error between the final position and the desired reference position for the trials with external disturbances applied underscores that none of the controllers account for external disturbances. Thus, it can be stated that none of the methods exhibit robust capabilities for disturbance rejection. Future works could approach the combination of the AMPC with controllers equipped with enhanced capabilities for the rejection of external disturbances, particularly in situations where these disturbances are completely unknown.

In addition, results showed no difference between the controllers in the error between the final position and the desired reference position when considering nonlinear damping in the real system. There does not appear to be a significantly superior controller in effectively adapting to nonlinear dynamics. On the other hand, the goal was reached when considering only nonlinear damping in the controller and linear damping in the real system. For a future application of the controller in a real application, a performance that does not account for external disturbances and that does not reach the goal when nonlinearities are present is considered to be not good enough. This behaviour can be caused by the fact that linear methods are used. Deeper exploration to improve the performance of the controllers can be done. This can include the use of control algorithms such as nonlinear MPC.

Regarding the experiments with the robot, the outcomes diverge from the conclusions drawn in the simulations.

According to the simulations, AMPC was expected to show a better performance than MPC. However, it only appears to be superior regarding the tracking of the measured position while showing a poorer tracking of the desired trajectory than the MPC. This discrepancy may arise from the physical limitations of the device as no great changes in the torque can be applied while in simulations this scenario may be feasible.

It is important to highlight that both MPC and AMPC showed the same variation of the force profile in the simulations. Thus, it was not expected that the torque experienced a substantial variation for the AMPC and not for the MPC. To improve the performance of the AMPC, alternative ways to regulate the variation in the torque applied to the robot can be studied. On the other hand, only changes in the value of the mass have been tested. Thus, future works could include the extension of the scenarios tested in the robotic device. This might involve introducing external known masses to the end-effector to test if greater values of mass would have a different impact on the controllers' performance.

REFERENCES

[1] Arshia Khan and Yumna Anwar. Robots in healthcare: A survey. In Kohei Arai and Supriya Kapoor, editors, *Advances in Computer Vision*, Advances in Intelligent Systems and Computing, pages 280–292. Springer International Publishing, 2020.

[2] Joseph, Azeta, Christian, Bolu, Abiodun, Abioye A., and Oyawale, Festus. A review on humanoid robotics in healthcare. *MATEC Web Conf.*, 153:02004, 2018.

[3] John Hu, Aaron Edsinger, Yi-Je Lim, Nick Donaldson, Mario Solano, Aaron Solochek, and Ronald Marchessault. An advanced medical robotic system augmenting healthcare capabilities - robotic nursing assistant. In *2011 IEEE International Conference on Robotics and Automation*, pages 6264–6269, 2011. ISSN: 1050-4729.

[4] Emre Sariyildiz, Roberto Oboe, and Kouhei Ohnishi. Disturbance observer-based robust control and its applications: 35th anniversary overview. *IEEE Transactions on Industrial Electronics*, 67(3):2042 – 2053, 2020.

[5] Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-observer-based control and related methods—an overview. *IEEE Transactions on Industrial Electronics*, 63(2):1083–1095, 2016. Conference Name: IEEE Transactions on Industrial Electronics.

[6] Vaneet Aggarwal and Mridul Agarwal. Control of uncertain systems. In Shimon Y. Nof, editor, *Springer Handbook of Automation*, Springer Handbooks, pages 189–204. Springer International Publishing, 2023.

[7] Daniela Selvi, Dario Piga, and Alberto Bemporad. Towards direct data-driven model-free design of optimal controllers. In *2018 European Control Conference (ECC)*, pages 2836–2841, 2018.

[8] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[9] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219):20180335, 2018. Publisher: Royal Society.

[10] Markus Quade, Markus Abel, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):063116, 2023.

[11] Ruiyun Qi, Gang Tao, and Bin Jiang. Adaptive control: A tutorial introduction. In Ruiyun Qi, Gang Tao, and Bin Jiang, editors, *Fuzzy System Identification and Adaptive Control*, Communications and Control Engineering, pages 55–74. Springer International Publishing, 2019.

[12] Jing Sun. Model reference adaptive control. In John Baillieul and Tariq Samad, editors, *Encyclopedia of Systems and Control*, pages 1–7. Springer, 2014.

[13] T. Senjyu, T. Kashiwagi, and K. Uezato. Position control of ultrasonic motors using MRAC with dead-zone compensation. *IEEE Transactions on Industrial Electronics*, 48(6):1278–1285, 2001. Conference Name: IEEE Transactions on Industrial Electronics.

[14] A Locatelli. Optimal control: An introduction. *Applied Mechanics Reviews*, 55(3):B48–B49, 2002.

[15] The MathWorks. What is optimal control?

[16] Ian R. Petersen and Roberto Tempo. Robust control of uncertain systems: Classical results and recent developments. *Automatica*, 50(5):1315–1335, 2014.

[17] Ronald Ping Man Chan, Karl A. Stol, and C. Roger Halkyard. Review of modelling and control of two-wheeled robots. *Annual Reviews in Control*, 37(1):89–103, 2013.

[18] Andrew Zulu and Samuel John. A review of control algorithms for autonomous quadrotors. *Open Journal of Applied Sciences*, 04(14):547–556, 2014.

[19] Peter Dayan and Yael Niv. Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology*, 18(2):185–196, 2008.

[20] Yisel Garí, David A. Monge, Elina Pacini, Cristian Mateos, and Carlos García Garino. Reinforcement learning-based application autoscaling in the cloud: A survey. *Engineering Applications of Artificial Intelligence*, 102:104288, 2023.

[21] Lennart Ljung. System identification toolbox reference.

[22] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, 2021.

[23] Johannes Köhler, Peter Kötting, Raffaele Soloperto, Frank Allgöwer, and Matthias A. Müller. A robust adaptive model predictive control framework for nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, 31(18):8725–8749, 2021.

[24] Maria Vittoria Minniti, Ruben Grandia, Kevin Fäh, Farbod Farshidian, and Marco Hutter. Model predictive robot-environment interaction control for mobile manipulation tasks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1651–1657, 2021. ISSN: 2577-087X.

[25] Mert Önkol and Coşku Kasnakoğlu. Adaptive model predictive control of a two-wheeled robot manipulator with varying mass. *Measurement and Control*, 51(1):38–56, 2018. Publisher: SAGE Publications Ltd.

[26] Fen Lin, Yuke Chen, Youqun Zhao, and Shaobo Wang. Path tracking of autonomous vehicle based on adaptive model predictive control. *International Journal of Advanced Robotic Systems*, 16(5):1729881419880089, 2019. Publisher: SAGE Publications.

[27] Akshaya Kumar Patra. Adaptive kalman filtering model predictive controller design for stabilizing and trajectory tracking of inverted pendulum. *Journal of The Institution of Engineers (India): Series B*, 101(6):677–688, 2020.

[28] Piwai N. Chikasha and Chioniso Dube. Adaptive model predictive control of a quadrotor. *IFAC-PapersOnLine*, 50(2):157–162, 2017.

[29] Chunting Jiao, Jun Yang, Xueqian Wang, and Bin Liang. Adaptive coordinated motion control with variable forgetting factor for a dual-arm space robot in post-capture of a noncooperative target. *International Journal of Advanced Robotic Systems*, 16(5):1729881419872342, 2019. Publisher: SAGE Publications.

[30] Christopher Lehnert and Gordon Wyeth. Locally weighted learning model predictive control for nonlinear and time varying dynamics. In *2013 IEEE International Conference on Robotics and Automation*, pages 2619–2625. IEEE, 2013.

[31] Colin M. Light, Paul H. Chappell, and Peter J. Kyberd. Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity. *Archives of Physical Medicine and Rehabilitation*, 83(6):776–783, 2002.

[32] M. C. M. Klotz, L. Kost, F. Braatz, V. Ewerbeck, D. Heitzmann, S. Gantz,

T. Dreher, and S. I. Wolf. Motion capture of the upper extremity during activities of daily living in patients with spastic hemiplegic cerebral palsy. *Gait & Posture*, 38(1):148–152, 2013.

[33] Matthias C. M. Klotz, Stefan van Drongelen, Oliver Rettig, Patrick Wenger, Simone Gantz, Thomas Dreher, and Sebastian I. Wolf. Motion analysis of the upper extremity in children with unilateral cerebral palsy—an assessment of six daily tasks. *Research in Developmental Disabilities*, 35(11):2950–2957, 2014.

[34] Susannah M. Engdahl and Deanna H. Gates. Reliability of upper limb movement quality metrics during everyday tasks. *Gait & Posture*, 71:253–260, 2019.

[35] Ellen Jaspers, Kaat Desloovere, Herman Bruyninckx, Guy Molenaers, Katrijn Klingels, and Hilde Feys. Review of quantitative measurements of upper limb movements in hemiplegic cerebral palsy. *Gait & Posture*, 30(4):395–404, 2009.

[36] Akira Shimada. *Disturbance Observer for Advanced Motion Control with MATLAB / Simulink*. Wiley-IEEE Press, 2023.

[37] Khelifa Baizid. A COMPREHENSIVE STATE-OF-THE-ART ON CONTROL OF INDUSTRIAL ARTICULATED ROBOTS. *Journal of the Balkan Tribological Association*, 20:499–521, 2014.

[38] Tariq Samad, Margret Bauer, Scott Bortoff, Stefano Di Cairano, Lorenzo Fagiano, Peter Fogh Odgaard, R. Russell Rhinehart, Ricardo Sánchez-Peña, Atanas Serbezov, Finn Ankersen, Philippe Goupil, Benyamin Grosman, Marcel Heertjes, Iven Mareels, and Raye Sosseh. Industry engagement with control research: Perspective and messages. *Annual Reviews in Control*, 49:1–14, 2020.

[39] Alberto Bemporad, N. Lawrence Ricker, and Manfred Morari. Model predictive control toolbox user's guide, 2023.

[40] Sylvain Miossec and Abderrahmane Kheddar. Human motion in cooperative tasks: Moving object case study. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 1509–1514, 2009.

[41] Chunchu Bhavani Prasad. The effects of nonlinear damping on the large deflection response of structures subjected to random excitation, 1987.

A First appendix

Tables of both the Top-Down and Bottom-Up approaches.

Case of study	Dynamics	Kinetics	Prior knowledge	Control	Goal condition
1.a) Open a bottle/can/jar (Control is the defining feature)	Mass (it can be negligible) + Damper (nonlinear)	Force Position	Force when opening for the first time Size of the lid	Increase the force while there is no movement → admittance controller	Release the lid. It is released when you are able to take the lid out without taking the jar
Considering the task as a whole (Prior knowledge)	-	Position	Position where the lid can be released	Position control (Force control)	Position the lid in the spot where it can be taken off.
Dividing the motion (goal condition): - First applying force until it starts moving	-	Force		Force control	Make the lid moving
- Move the lid to unscrew and take it out	-	Position		Position control (Force control)	Find the spot to take it out
1.b) Close a bottle/can/jar	Mass (it can be negligible) + Damper (nonlinear)	Force Position	Size of the lid (more force if it's bigger)	Increase the force while there is no movement → admittance controller	Fix the lid Minimum energy principle → applied force until you can't move but no more
2) Open the tap (Focusing on control method)	Mass (can be neglected) + nonlinear damping – Tap	Force (Position)	Force for open it to the maximum / Whole range of motion	Admittance controller	Water goes out of the tap
Considering the task as a whole focusing on the prior knowledge	-	Position	Position that allows water to go out	Position control (Force control)	Reach the position that allows water to go out with a certain pressure
Dividing the task (goal condition): - Release the tap	-	Force		Force control	Being able to move the tap
- Move it	-	Position		Position control	Water goes out

3) Prepare breakfast	Mass – bottle milk Mass – mug	-	Weight of the mug Weight of a full milk bottle		Put milk on a mug
- Open the cupboard (5a)	-	*See 5a*	Whole range of motion	*See 5a*	*See 5a*
- Take out the mug focusing on <u>control method</u>	-	Position		Admittance control	Place the mug in the table
- Take out the mug focusing on <u>prior knowledge</u>	-	Position	Weight of the mug Position where it have to be placed	Position control (Force control)	Place the mug in the table
- Open the fridge (5a)	-	*See 5a*	-	*See 5a*	*See 5a*
- Take out the milk	-	*See 3 take out the mug*	*See 3 take out the mug*	*See 3 take out the mug*	The bottle of milk is on the table
- Open the bottle of milk (1a)	-	*See 1a*	-	*See 1a*	*See 1a*
- Pour the milk	-	Position		Position control	When the mug is full
- Open the microwave	Spring (nonlinear) - button	Velocity		Force control for the button	When the door is open
- Put the mug in the microwave	-	*See 3 take out the mug*	*See 3 take out the mug*	*See 3 take out the mug*	The mug is inside the microwave
4) Take notes (handwriting)	Mass (linear) (No friction with the paper is considered)	Position	Force that breaks the tip of the pencil Size of the paper	Position control and force control (separate)	There is no more empty paper to write
5.a) Open a window/door <u>Control</u> - Move the handle - Move the door/window	Mass + non/linear damper	Position Force	-Whole range of motion -Kind of handle (door or window)	Admittance/ Impedance controller	The door/window is completely open

Considering the tasks as a whole (<u>prior knowledge</u>): - Move the handle - Move the door/window	-	Position	-Whole range of motion of handle/door /window	Position control (Force control)	Handle completely pressed or in right position to allow the movement of the door/window Door/window is completely open
Dividing both tasks (<u>goal condition</u>): - Force until movement	-	Force		Force control	The handle/door starts moving
- Keep moving	-	Position		Position control	Handle completely pressed or in right position to allow the movement of the door/window Door/window completely open
5.b) Close a window/door -Move the door/window	Mass + non/linear damper	Position	-Whole path of the door	Admittance/ Impedance controller	Neither air nor person/object can pass
6) Throw out the rubbish	Mass (linear)	Position			The bag is inside the rubbish bin
- Open the rubbish bin (1a)	-	*See 1a*	Position where the lid is completely open	*See 1a*	The lid is completely open
- Placing the bag in the bin		*See 3 take out the mug*	*See 3 take out the mug*	*See 3 take out the mug*	The bag is inside the bin
7) Hand out cards	Mass (linear)	Position	The size of the space where the cards need to be split	Position control	The cards are split
8) Answer a call	Damper (linear)	Velocity	Dimensions of the screen -Force needed to be detected	Velocity control	Minimum energy principle You go through half screen
9.1) Eat a steak	Fork – mass		Weight of the fork		The steak is in the mouth

	Steak – spring		Weight of the whole steak		
- Insert the fork in the steak		Velocity		Velocity control and position control	Fork completely inserted Minimum energy principle
- Move the steak+fork to the mouth	Steak + fork - mass	Position	Where does it need to be moved to	Position control	Steak in the mouth
9.2) Cut a steak	Knife – mass Steak – spring +damping	Velocity (Position, Force)	Cutting a raw steak	Velocity control and position control	The plate/chopping board is reached with the knife
10) Wash the dishes	Soap bottle- mass+ spring Scouring pad – spring Object to clean - mass	-	Dimensions of the scouring pad and the object		
- Open the soap (1a)	-	*See 1a*	*See 1a*	*See 1a*	*See 1a*
- Tilting the soap on the scouring pad	-	Position	The bottle needs to be tilted 180º	-Tilting: position control	Bottle is tilted 180º
- Put soap on the scouring pad	-	Force	Force to get soap	Force control	1/8 of the scouring pad has soap
- Rub the object	-	Velocity (Position, Force)	Force needed to clean the surface	Velocity control	Go through the surface at least once.
- Open the tap (2)	-	*See 2*	-	*See 2*	*See 2*
- Rinse	-	Position	-	Position control	Go through the surface at least once
11) Brush your teeth	Toothbrush – mass Toothpaste – mass + damper	-	Dimensions of the toothbrush and the toothpaste bottle.	-	-

	Teeth – infinite stiff spring				
- Open the toothpaste (1a)	-	*See 1a*	*See 1a*	*See 1a*	*See 1a*
-Put toothpaste on toothbrush	-	Position (Force)	Force to get toothpaste	Position control (Force control)	Whole surface has toothpaste. For the force needed → minimum energy principle
- Brush the teeth (same as 10)	-	Velocity (position, force)	Force needed to brush	Velocity control	Go through the surface at least once.
12) Vacuum **It is considered that the user doesn't apply force in the up/down direction**	Mass + nonlinear damper	Position	Size of the space to clean -Material of the surface -Weight of the vacuum so the applied force to move it is constant	Position control	The surface is clean To be more concrete → the vacuum goes through all the space once

<u>Case 1a and case 2: Open a lid/tap</u>

It can be extrapolated to open a rubbish bin (case 6).
Three approaches can be made.
The first one is considering that the <u>control</u> is admittance control (force control).
Other option is considering that the <u>prior knowledge</u>, in which point the lid can be released, is known. Thus, it can be considered as position control.
The other option is <u>dividing the task</u> in two so defining different <u>goal conditions</u>: force until it starts moving and keep moving it. In this case each subtask has its own control.

<u>Case 3: Subcase take out the mug/milk</u>
Can be extended to moving objects in general.
Considering known the <u>prior knowledge</u> the position where it needs to be placed it can be just position control.
If the type of <u>control</u> it is what it wants to be applied then admittance control can be considered.
Based on the <u>goal condition</u> it can be divided in force until the motion is started and then keeping the motion.

Case 3: Subcase pouring water/milk/soap: it can be divided in two subtasks putting up the object (force) and then tilt it (position). The components of the tasks are mass, force, and position. Other tasks that can relate with this one:

- Putting the rubbish bag in the rubbish bin
- Taking any object and move it.
- Open a window/door (ignoring the damping)
- Open a bottle (admittance control, ignoring the damping)

Case 5a: Open a door/window

It can be extrapolated to open cupboard, open fridge in case 3. The following approaches can be followed for the two subtasks, moving the handle and moving the door/window itself.

Considering the prior knowledge, position where the door/window is completely open.

Dividing the task based on different goal conditions: force until it starts moving and keep moving it.

The control is admittance control (force control).

BOTTOM-UP APPROACH

1.a) Open a bottle/can/jar (<u>Control</u> is the defining feature)	Mass (it can be negligible) + Damper (nonlinear)	Force Position	Force when opening for the first time Size of the lid	Increase the force while there is no movement → admittance controller	Release the lid. It is released when you are able to take the lid out without taking the jar
Considering the task as a whole (<u>Prior knowledge</u>)	-	Position	Position where the lid can be released	Position control (Force control)	Position the lid in the spot where it can be taken off.
<u>Goal condition</u>): - Force until it starts moving	-	Force		Force control	Make the lid moving
- Unscrew the lid	-	Position		Position control (Force control)	Find the spot to take it out

Option 1: Apply force until it starts moving and decrease the force while the velocity increases.

- $f_i > f_f$
- $v_i < v_f$

This can be seen as decreasing the impedance over time.

- $Z_i > Z_f$ → admittance control

Option 2: Apply force until it starts moving and decrease the force when reaching the desired position.

- $f_i > f_f$
- $x_i < x_f$

Option 3: supposing known the force needed to move the lid, move until reaching the desired position.

- $x_i < x_f$ → position control

5.a) Open a window/door Control - Move the handle - Move the door/window	Mass + non/linear damper	Position Force	-Whole range of motion -Kind of handle (door or window)	Admittance/ Impedance controller	The door/window is completely open
Considering the tasks as a whole (prior knowledge): - Move the handle - Move the door/window	-	Position	-Whole range of motion of handle/door/ window	Position control (Force control)	Handle completely pressed or in right position to allow the movement of the door/window Door/window is completely open
Dividing both tasks (goal condition): - Force until movement	-	Force		Force control	The handle/door starts moving
- Keep moving	-	Position		Position control	Handle completely pressed or in right position to allow the movement of the door/window Door/window completely open

Difference with open a lid: when is the task done. In this case there are two options, when you see the thing you want to take or there is enough space to enter/allow the air to enter or you open it completely.

Option 1: Apply force until it starts moving and decrease the force while the velocity increases.

- $f_i > f_f$
- $v_i < v_f$

This can be seen as decreasing the impedance over time.

- $Z_i > Z_f$ → admittance control

Option 2: Apply force until it starts moving and decrease the force when reaching the desired position.

- $f_i > f_f$
- $x_i < x_f$

Option 3: supposing known the force needed to move the lid, move until reaching the desired position.

- $x_i < x_f$ → position control

Option 4: The task is completed when the door/window is completely open.

- $f_i < f_f$
- $v_i > v_f$

This can be seen as increasing the impedance over time.

- $Z_i < Z_f$ → admittance control

3) Prepare breakfast	Mass – bottle milk Mass – mug	-	Weight of the mug Weight of a full milk bottle			Put milk on a mug
- Open the cupboard (5a)	-	*See 5a*	Whole range of motion	-	*See 5a*	*See 5a*
- Take out the mug focusing on <u>control method</u>	-	Position			Admittance control	Place the mug in the table
- Take out the mug focusing on <u>prior knowledge</u>	-	Position	Weight of the mug Position where it have to be placed		Position control (Force control)	Place the mug in the table
- Open the fridge (5a)	-	*See 5a*	-	-	*See 5a*	*See 5a*
- Take out the milk	-	*See 3 take out the mug*	*See 3 take out the mug*		*See 3 take out the mug*	The bottle of milk is on the table
- Open the bottle of milk (1a)	-	*See 1a*	-	-	*See 1a*	*See 1a*
- Pour the milk	-	Position			Position control	When the mug is full
- Open the microwave	Spring (nonlinear) - button	Velocity			Force control for the button	When the door is open
- Put the mug in the microwave	-	*See 3 take out the mug*	*See 3 take out the mug*		*See 3 take out the mug*	The mug is inside the microwave

Open the cupboard is the same as previous case.

Take out the mug:

- $x_i < x_f$ → position control.

Open the fridge is the same as previous case.

Take out the milk:

- $x_i < x_f$ → position control.

Open the bottle of milk is open a bottle.

Pour the milk: this example can be applied to any task that implies movement of an object.

- $x_i < x_f$ → position control

Open the microwave:

Option 1: make the button to move until you reach the end of the motion range.

- $f_i < f_f$
- $v_i > v_f$

Can be seen as increasing the impedance over time.

- $Z_i < Z_f$ → admittance control

Option 2: increase the force until reaching a position.

- $f_i < f_f$
- $x_i < x_f$

Option 3: only consider the velocity, so you apply some velocity to the button until it becomes 0 meaning you reached the end.

- $v_i > v_f$ → velocity control

Put the mug in the microwave is the same as taking out the mug from the cupboard.