POLITECNICO DI TORINO

Master degree in Mechatronic Engineering

Master Degree Thesis

Trajectory Planning and Control of High-Speed Pick and Place in Industrial Processes

Supervisor
prof. Alessandro Rizzo

Candidate
Arianna Loi
240482

Tutor
Istituto Italiano di Tecnologia
dott. Giovanni Gerardo Muscolo

December 2018

Summary

The manipulation of bottles in high speed conditions is a serious issue in industrial processes. The objective of this thesis is to develop novel systems to pick bottles from a high-speed conveyor and to place them on another conveyor with a predefined position and orientation. Each cycle of pick and place of bottles must be performed within a maximum time. The solution to this problem has been requested by a company and a joint patent on the proposed solutions is in a pending status.
In a first step, the optimization of the present technology is proposed. In a second step, three different new solu tions have been studied. For each proposed solution some novel manipulators and control systems have been studied and simulated. The comparison among control outputs and a detailed analysis in order to satisfy the objective of the thesis permitted us to define the novel system and the control architecture proposed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem statement

This thesis work was carried on in collaboration with the IIT (Italian Institute of Tecnology) and is more specifically related to a project involving the italian company Fameccanica. The project deals with different kinds of industrial production line machinery offered by the company to the industry market, that need to be optimized concerning the necessity of high speed manipulation and processing of objects and materials.
The real issue on which this research is based is the transfer of bottles from a conveyor belt to a screw conveyor positioned on a second belt. The general scheme of the starting condition for the developement of the task handling is represented in Fig.1.1.
The bottles travel at speed that may vary between 0.4 and 1 m/s. The flow rate of the bottles should be kept constant for all the phases of the process and their travelling speed must be constant as well.
The initial condition consists in the bottles laying on their side on the first conveyor; the objective of the manipulator is to place the bottles in vertical position on the second conveyor, which is coupled with a screw conveyor that holds the bottles in position while they are being carried to their next destination.
The main problems that arise in this set-up are: the unknown position of the bottle on its initial condition, since the vibration of the conveyor makes it slide and rotate on the moving plane; the orientation of the bottle head that is not constant, while in the final position all the bottle heads must be directed upwards.
Currently the problems above are handled by an industrial robotic arm, redundant and provided of a gripper that is able to rotate the bottle and put it in vertical position. The main problem of this solution is the impossibility to

transfer the whole bottle flow at very high speed, due to the kinematic limits of the robot and of the gripper joint motors. As consequence, some residual flow remains on the first conveyor belt and cannot be manipulated with a single robotic device.
The second problem is the impossibility to transfer all the bottles to the screw conveyor with the present gripper, due to its mechanichal structure.
The purpose of this study is to elaborate and simulate new solutions to this problem, in particular aiming to fulfill the need to keep the flow rate of the items constant. Eventually the simulated results will be compared to the optimized existing technology in terms of speed of the overall process and efficiency in satisfying the problems listed above.

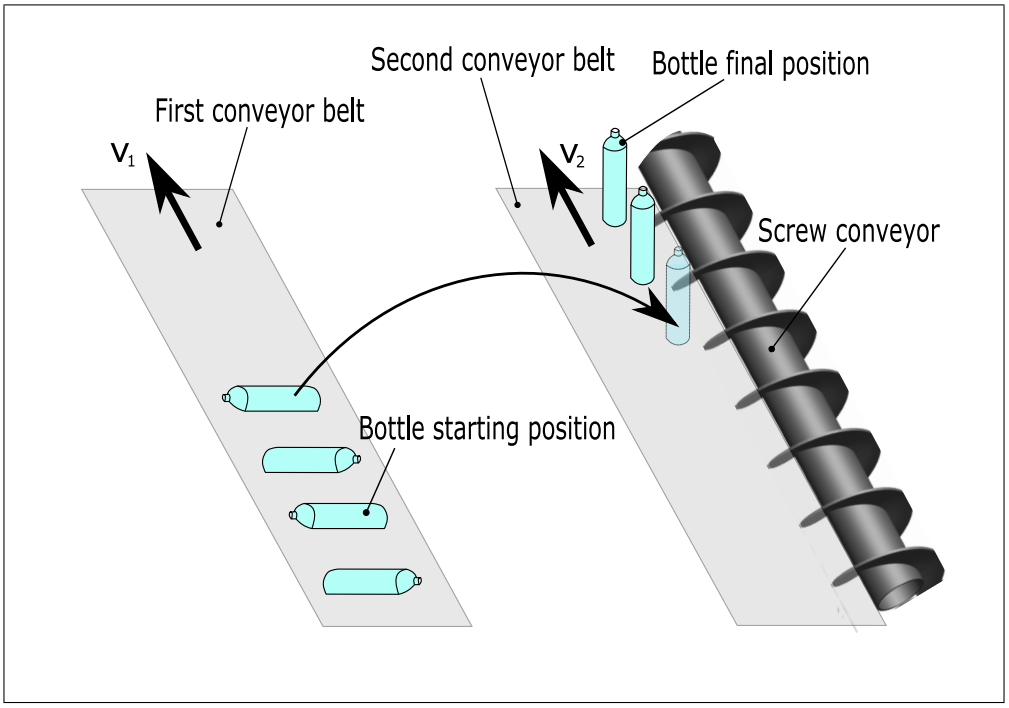


Figure 1.1: Layout of the prcess: bottles with different orientation need to be transferred to the second conveyor/screw conveyor apparatus

1.2 State of the art

The first approach studied in this thesis is the solution currently adopted by the company, consisting in an industrial robot provided of a gripper that has

the ability to grab the bottles and rotate them to put them in vertical position, all with the same orientation, i.e. with the head upwards. The rotation of the gripper has been tested to obtain the minimum rotation time of 0.8 seconds to complete the task. By the way the big ostacle in the time minimization of the whole operation is the transfer of the bottle from a first conveyor belt to another conveyor belt, which involves none of the gripper joints, but all the other joints of the robot and is limited by their maximum speeds. To verify the limits of the robotic solution, more in general and not specifically speaking of the gripper used, an optimization of the trajectory of the robot has been done in this thesis.

1.2.1 Trajectory optimization in industrial robotics

Trajectory optimization in robotics has been widely studied and spreads among different aims, e.g. time-minimization, jerk-minimization, obstacle avoidance, etc.
If planning is done in the operational space, the time law associated to the geometric path is defined first [1]; only afterwards an inverse kinematics algorithm is used to get a time sequence of joint configurations corresponding to the geometric trajectory. This class of methods for trajectory planning is suitable when the path and time law in the operational space must be well defined, for example in presence of obstacles, and there are specific values needed for velocity and acceleration of the TCP (tool center point). An example of task that could need trajectory planning in the operational space is welding.
The drawbacks of this approach are the large computation time and, most of all, the difficulty or impossibility to solve the inverse kinematics problem in the neighbourhood of singular configurations, when the robot is redundant. For the manipulation of objects, generally the trajectory planning process is carried on in the joint space. Firstly the desired geometric path is planned in the task space, then inverse kinematics is computed for a set of points that typically consist in the extremal points for point-to-point motion, or it includes also some intermediate points when it's needed; then the time law is generated in the joint space by interpolating the joint variables samples obtained from inverse kinematics.
The easiest and most naive point-to-point planning is the so-called trapezoidal velocity profile trajectory (Fig.1.2). It is well visble that the acceleration profile is not a continuous profile. The derivative of acceleration, that is jerk, is very high in the discontinuities, and this is an unwanted phenomenon. This is why the best way to avoid discontinuities is to choose a polinomial interpolation method (Fig.1.3).

The optimization process takes place in the stage when the time law for the geometrical path or for the joint variables points is generated.

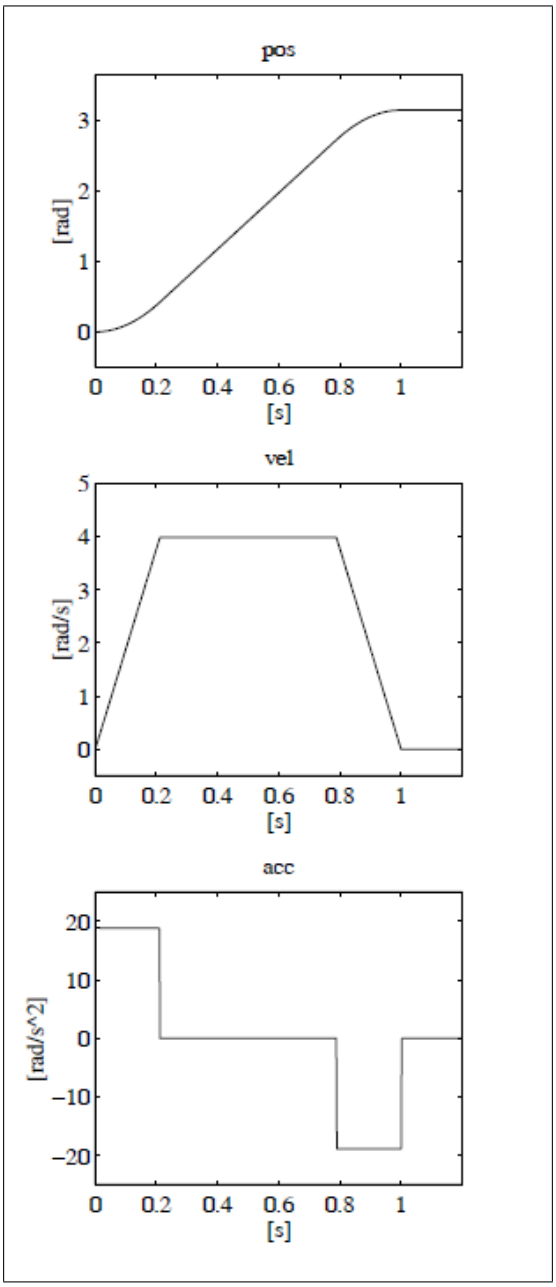


Figure 1.2: Image courtesy of [1]. Time history of position, velocity and acceleration with a trapezoidal velocity profile timing law

Usually the optimization problem involves a cost function to be minimized, also called objective function, a set of constraints and an algorithm to find the global minimum. The choice of such parameters is determined by the variables that

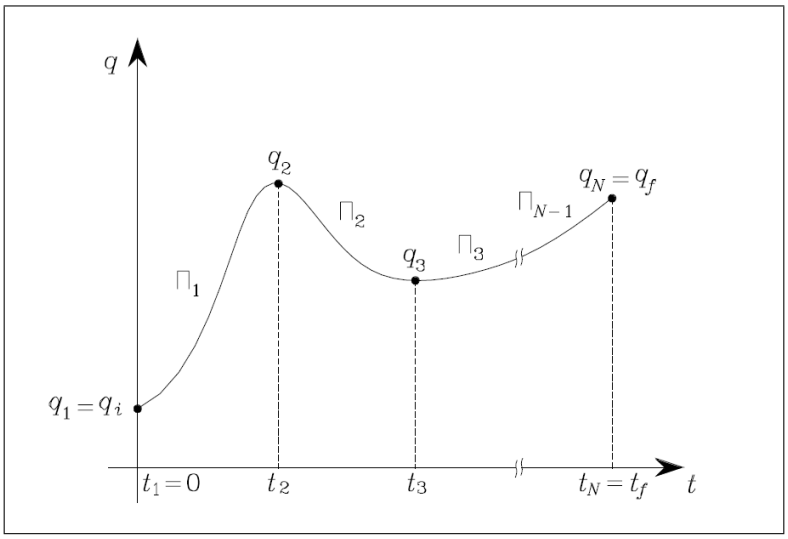


Figure 1.3: Image courtesy of [1]. Characterization of a trajectory on a given path obtained through interpolating polynomials.

must be minimized and the environmental constraints such as obstacles or the maximum speed, acceleration and torque of the joint motors.
In other cases, when the objective is simple and no particular constraints are present, one can rely on well studied techniques that have already been proved to work well in these cases, with no need of formulating a specific optimization problem. An example of this kind of methods are third order and fifth order splines and B-splines, that are piecewise polynomial curves, i.e. parametric representations that approximate a set of points in the plane. In the case of trajectory planning, the x-axis of the plane is the time axis and the y-axis is the axis of the values taken by the joint position, therefore it is necessary to specify the time intervals that separate the points and the algorithm will aim to generate a curve whose derivatives will define velocity, acceleration and jerk of each joint in each time instant.

In [2] the trajectory planning is developed through the formulation of an optimization problem where the objective function is composed by two terms: one accounting for the execution time, the other for the jerk. A trade-off can be found between the minimization of these two parameters by choosing the suitable weights. A solution is based on 5th order B-splines, namely linear combinations of polynomials recursively computed, where the parameters to tune are the coefficients of the base functions, called control points.
The objective function and the set of kinematic constraints must be translated in terms of the control points. To run the algorithm it is also necessary to find

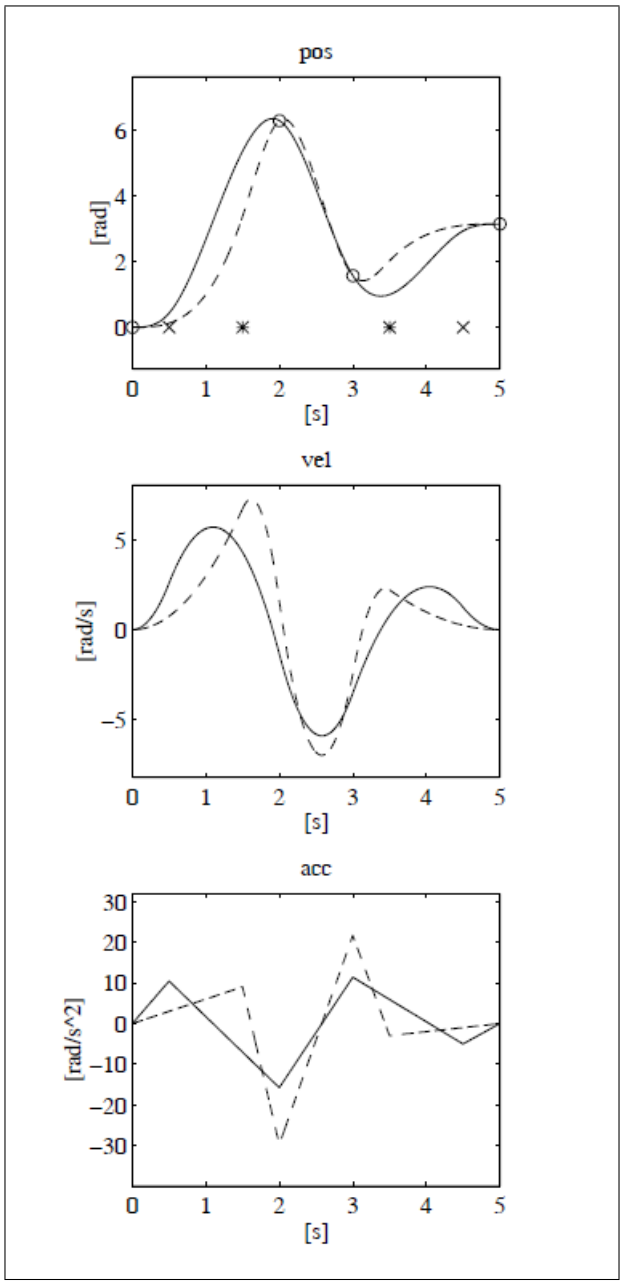


Figure 1.4: Image courtesy of [1].Time history of position, velocity and acceleration with a timing law of cubic splines for two different pairs of virtual points.

an initial solution to start with; the choice of the initial solution is important since it will affect the execution time and the final result.

In [3] the objective of the work is to compare the performance of two methods for trajectory planning that were previously studied by the authors. Both the procedures are based on the usual approach of the optimization problem with a cost function to minimize. The two compared solutions consist in the choice of modeling the trajectory with cubic splines and 5th order B-splines respectively. The test results in better performance of the cubic splines in terms of kinematic variables, hence this solution is more suitable if the task imposes strict constraints on the maximum value of velocity, acceleration and jerk of the joints. On the other hand, 5th order B-splines were found to lend smoothness to the acceleration and jerk profiles, helping to reduce vibrations and errors.

In [4] the adopted technique is the sine-jerk profile in the joint space. The previously known sine-jerk approach is optimized in this paper by relaxing the constraint of synchronized phases (acceleration,constant velocity, deceleration) for all the joints; this is done by solving the optimization problem separately for each joint, therefore it allows to consider the case of a robot whose joints have different characteristics and constraints. The total execution time is set to be equal for all the joints, but the single acceleration times are considered different for each joint. The result is a smooth motion profile with a fast total operation time.

The study in [5] deals with the problem of kinematic singularities in trajectories planned in the operational space, especially in the case of on-line planning, since in this situation critical points cannot be checked in advance. The proposed solution is a trajectory scaling system, whose core is a scaling filter able to provide a feasible output. The system receives a nominal trajectory in form of curvilinear coordinate, this must be smoothly scaled taking into account the constraints firstly defined in the operational and configuratiuon space and subsequently translated in the curvilinear coordinate space. The algorithm checks the feasibility of the signal and, if this condition is not met, it computes the best feasible approximation of the input. In the event of approaching a singularity, the system scales the bounds on velocity in order to keep the tracking of the path.

In [6] the traditional velocity profile in robots for pick and place applications is the well known trapezoidal velocity profile. The main problem with the traditional approach is the non-smoothness of the displacement and subsequent discontinuity in the acceleration profile, causing jerks and vibrations in the actuators. Instead of using the standard three-segment path, the kind of trajectory proposed in this paper follows an elliptical path, which is defined in the Cartesian space by a parametric vector function (Fig.1.6). Then the parametrized curve is represented in terms of the mod sine cam law equation, that can be designed with two parameters specifying the duration of constant

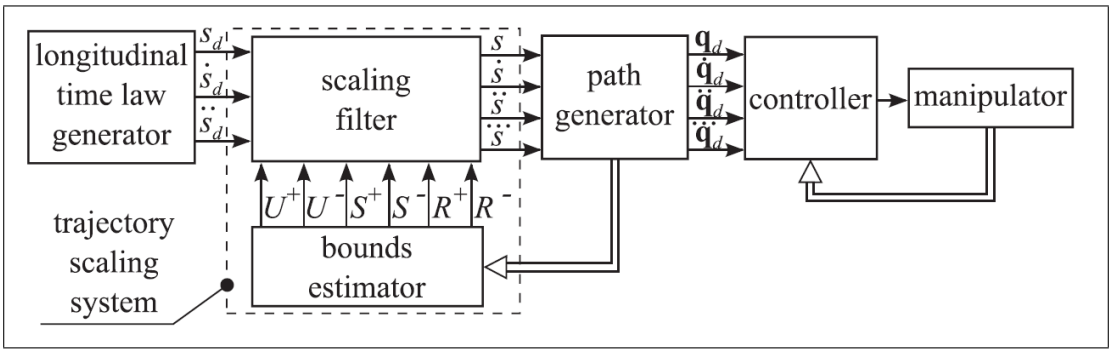


Figure 1.5: Image courtesy of [5]. A trajectory scaling system is placed between the longitudinal time law generator and the path generator in order to achieve feasible trajectories.

velocity segments and the degree of asymmetry. This last property has been proved to be particularly useful in high speed conditions, when the torques on the actuators present more abrupt changes.

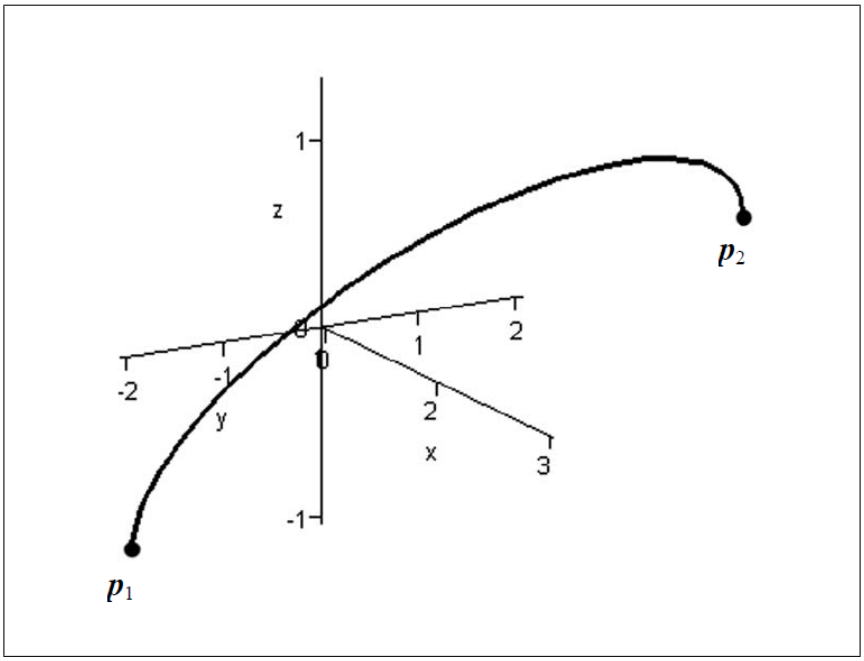
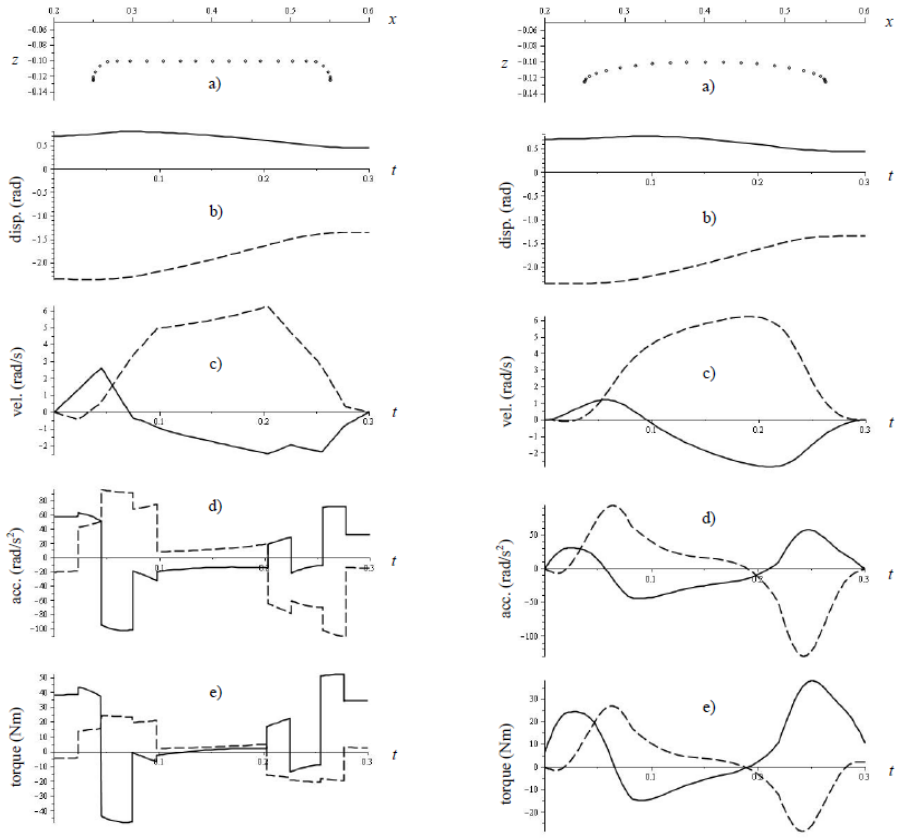


Figure 1.6: Image courtesy of [6]. Elliptical pick and place path generated by equation (1) between two arbitrary points p1 and p2.

In [7] the article describes an algorithm for detecting, locating and picking



(a) Rectangular pick and place path with trapezoidal velocity profile plots for joints 1 (solid) and 2 (dashed). a) Geometric path, b) Displacement, c) Velocity, d) Acceleration, e) Torque.

(b) Elliptical pick and place path with Mod Sine motion profile plots for joints 1 (solid) and 2 (dashed). a) Geometric path, b) Displacement, c) Velocity, d) Acceleration, e) Torque.

Figure 1.7: Images courtesy of [6]. Comparison between rectangular pick-and-place path and its elliptical equivalent.

objects on a moving belt thanks to a stationary matrix video camera. A belt variable is a relative transformation that defines a moving reference frame attached to the belt. An object completely inside the conveyor belt window is detected by a photocell that activates robotic vision to aquire an image of the object. The gripper frame and the object frame are represented with respect to the base frame. A destination for the robot is computed and changed dynamically basing on the transformation relating the vision frame to the base frame, which is updated every 8 milliseconds with the data received by the belt

encoder. Then the tracking procedure starts and the item is picked on-the-fly.

In [8] the study proposes a solution for trajectory planning that starts from the generation of a geometric path represented by a piecewise parametric function. The system constraints on the torque of the actuators are translated in contraints on velocity and acceleration, then the time minimum trajectory is compute through an algorithm that aims to limit jerks along the path. The first step is the generation of the accelerating and decelerating trajectories from the initial and final point o the path respectively. This is done by forward integration until the limit on maximum jerk or acceleration is reached. The acceleration continues untile the curve of maximum path velocity previously computed is reached, then the trajectory stays along the curve until some other constraints on acceleration are satisfied. At a certain point the two trajectories, backward and forward, intersect.

The algorithm proposed in [9] consists in two phases: firstly the minimum-time trajectory is planned in the joint space, by means of the Particle Swarm Optimization (PSO) integrated in the optimal trajectory optimization algorithm, given an initial set of feasible cartesian knots that will be converted through inverse kinematics. Once the trajectory is obtained in form of cubic splines, the second phase deals with the problem of the typical high jerk at the initial and final point of the spline. The optimization problem is formulated to minimize the time of each segment between two knots, then the cubic spline is replaced at the first and last point of the trajectory with 7th order polynomial which were proved to be the best solution for jerk containment.

The problem to be solved in [10] is the trajectory planning and control of an industrial robotic arm. The proposed method belongs to the direct methods class, i.e. it is based on the discretization of the dynamic variables and the associated parameter optimization problem. In this case, the joint evolution vector is parametrized through a clamped cubic spline model with a set of uniformly distributed free nodes.
A cost function is introduced to take into account both transfer time and actuator efforts, with relative weight coefficients. This solution allows to include in the optimization program also the physical constraints concerning the joints, such as torques, jerks, accelerations etc. The optimization problem is non-linear and it's solved with the Sequential Quadratic Programming approach. In the case of grasping mobile objects, additional constraints are used to set the position of the end effector equal to the object position at grasping time and the velocity of the TCP is the same as the one of the object.

In [11], starting from the direct and inverse kinematic model of the robot, the control is designed and the optimal trajectory is computed. The parameters of each joint are identified by means of a least squares estimator. An algorithm

is presented to compute the time-optimal trajectory. It consists in a cubic spline whose intermediate points are updated and added by the iteration of the algorithm. The new intermediate nodes are computed directly from the minimization of a function that takes into account the total pick-and-place time. The control is then implemented through a simple PID torque controller.

In [12] the problem of classifying objects with different geometry and unknown position on a moving conveyor belt has been solved with the use of an optical device consisting in a two-dimensional source of structured light and a CCD camera. The height of each object deviates the laser from the reference line on the belt.
Data has been processed by a fuzzy logic module to distinguish different objects basing on their height, then the objects belonging to a set of items with known geometrical characteristics have been identified by means of a multilayer perceptron neural network.

In [13] the evaluated parameters are gripping rate and efficiency. After demonstrating that a non-redundant device is not a good solution when speaking of a stochastic flow of items, the article focuses on redundant kinematic devices, whose flexibility allows a varying gripping time that depends on the position of the gripping.
The strategy improvement consists, for the FIFO method, in working close to the equilibrium position of the system, since the FIFO method is known to be unstable expecially in overload condition. The shortest processing time method has been optimized by giving priority to those objects in the downstream area when the STP object is still in the upstream area.

The problem of optimizing the pick-and-place task can involve more than one robot. In this case each robot must choose the objects to pick, that are not going to be picked by other robots, therefore the whole process must be coordinated and becomes more complicated.
The article [14] proposes a solution based on noncooperative dynamic games in an environment with more than one robot performing the same pick-and-place task. Each robot aims to minimize its individual cost function, but the individual function takes into account also the actions of the neighboring robots. The algorithm of the game has three criteria of decision: the robot should pick up the closest objects first, then it considers picking up the objects that are more distant from the neighbors and finally, the robot should continue picking up objects from the current working area. The process is repeated until it converges to a Nash equilibrium or to a time-constrained decision.

In [15] a simulator has been developed with the use of MATLAB, with the objective to test and compare different layouts of pick-and-place operations in packaging processes, where boxes are filled with the items and then leave the

working space on a second conveyor belt.
The counter-current configuration for the conveyor belts, which was thought to be the best performing, has been proved to be less convenient than the alternative: a co-current configuration. In fact, it happens that the dropouts in the proposed co-current scheme have been reduced under stress condition. Different strategies have been tested for each configuration; in the optimal one, a buffer is used, but only when the box or the unit to be picked are in the second part of the robot workspace.

In [16] the study deals with the optimization of the flow of objects on a conveyor belt in the case of multirobot systems.
A set of candidate part-dispatching rules is defined,including the most known FIFO, SPT and LPT, as well as specifc serial (SR), shortest and longest distance. The method for choosing the best combination of rules is called GRASP. It starts from a partial solution, gradually integrated by creating a restricted candidate list obtained by maximization of the largest incremental increase in the objective function. The next step is to estimate the minimum-maximal part flow. In order to do this, different samples of patterns of items on the conveyor must be considered and the maximal part flow must be estimated for each sample under a given combination of rules. In this way it is possible to obtain mean and variance of the maximal part flow, which is considered to follow a normal distribution.

In [17] the objective is to minimize the decision time in the choice of the scheduling rules during pick-and-place operations, in order to make the system more efficient, with almost real-time decisions. The metaheuristics applied in the problem are: ant colony optimization (ACO), particle swarm optimization (PSO) and genetic algorithms (GA).
The ACO gives a combination of scheduling strategies chosen among a set that involves the usual ones (FIFO,STP, etc.). The points chosen by the ants for pherormone release coincide in this application with the queuing strategies. In the GA the combination of scheduling rules is represented by an individual, each rule is a gene: firstly a set of random individuals is generated, the next generation is produced through crossover and mutation and so on. In the PSO the solution encoding is the same as for the GA but the algorithm is different. Eventually the best solution in the tests has been the ACO algorithm.

In [18] the proposed solution is an adaptive controller for picking a moving object based on the prediction of the object velocity and position in the gripping moment. The objective is to bypass problems linked to long processing times of the aquired images of the object and to find a compensation to modelling errors.
The model of the object velocity is estimated considering the error with the

least squares approach. Data about position is obtained through camera vision samplings. A self-tuning adaptive control is implemented, including a trajectory planner that uses the known trapezoidal velocity profile. The gripper position control is designed by means of a multivariable ARX model whose parameters are estimated on-line, in order to avoid the complex dynamics of the manipulator.

In [19] the control problem in this paper concerns nonlinear systems which are linear with respect to the control input. The proposed method is the sliding modes one: the states must reach and follow the sliding mode functions in the state space, i.e. some linear or non-linear combinations of the states designed as references. Firstly the control design is discussed in a general form, focusing on the problem of providing a continuous control input to the system, since some plants (e.g. all motion control systems) can't accept a discontinuous control input. The procedure to guarantee the stability of the closed-loop system is based on the Lyapunov functions criteria.
Later, the problem is restricted to the control of robotic manipulators, showing how the solution is suitable for different tasks such as tracking, impedance control and force control, with a minimum modification in the sliding mode manifold.

The main problem of picking a moving object consists in the prediction of its position in the future instants in order to track its path until the gripping moment. In some situations it is difficult to get updated and reliable information about the target motion through a vision system, for example when the speed is high, given that image processing algorithms introduce long time-delays.
In [20] the proposed solution is based on a Kalman filter to estimate the state and uses generalized predictive control (GPC).
In the Kalman filter, information about the position in the previous instants with respect to the present time is given by a visual feedback. The command input is computed at each time instant. The optimal command input is the weighted average of a suitable number of previous command inputs, in order to limit the effect of large corruption in some samples of data.
Similarly, in [47] a Kalman filter is used to predict the target object position in the next time instant, allowing the recomputation of the new robot trajectory as the object is moving. The procedure is shown in the diagram in Fig.1.8. This method belongs to the class called "APPE", which stands for Active Prediction, Planning and Execution, a particular case of "PPE" strategies (Prediction, Planning and Execution) where the process of prediction, planning and execution is repeated many times over time and the control input that is the robot trajectory is updated for every cycle.

Another problem that arises in many robotic environments is the presence

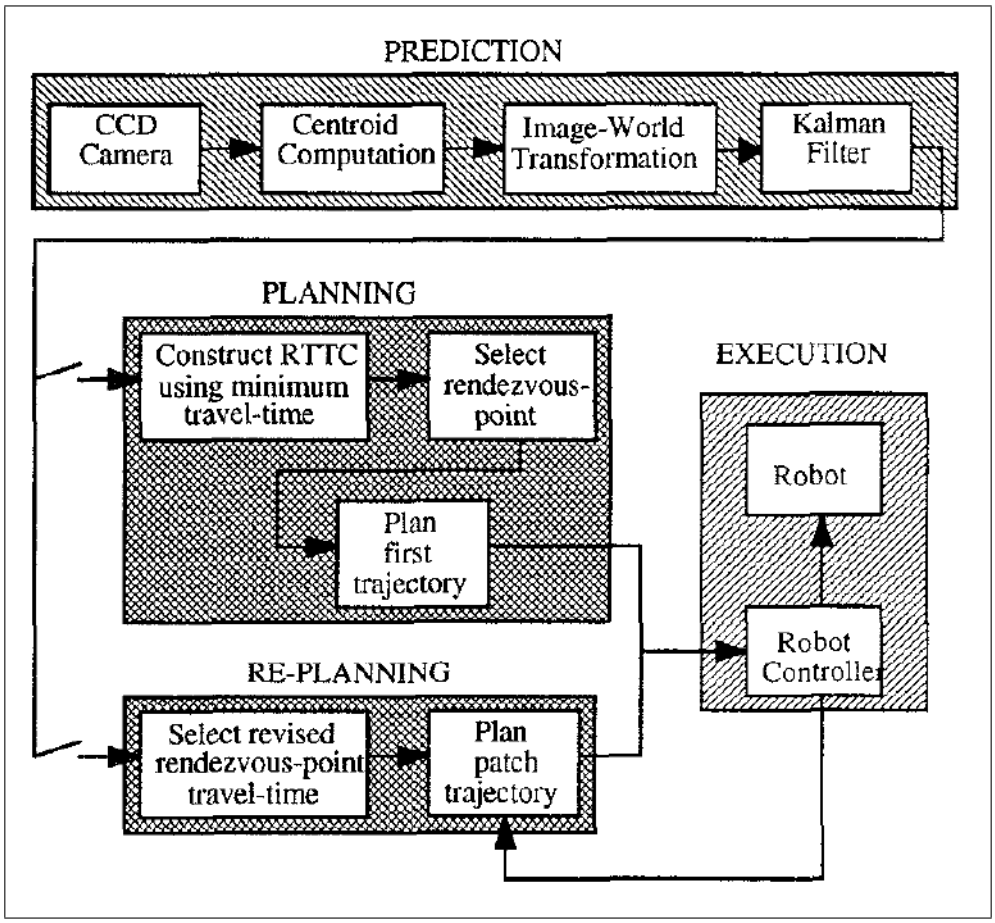


Figure 1.8: Image courtesy of [47]. APPE system implementation.

of obstacles that limit the robot workspace, which may be fixed or moving obstacles. If the obstacle is fixed, the problem is easier to handle because it can be solved by redefining the robot workspace in its control environment and limitating the trajectory generation to a smaller portion of the geometrical space. If the obstacle is not a constant factor, but is for example a moving object that enters the robot's workspace suddenly, it has to be identified through a vision system and avoided in the alorithm of the robot trajectory computation.
In [21] the study presents a method to handle the presence of obstacles in the trajectory of redundant manipulators. The obstacle is modeled as a sphere and the problem of avoiding the object is formulated in the form of a bilevel optimization problem. In the first level the new position of the robot is computed so that the obstacles are avoided, in the second level the solution is improved by maximizing the manipulability of the robot. The objective functions are

dependent on the sum of the distances between the center of each obstacle and each joint, but also on the distance of the robot from the final position and on its manipulability.
The solution is obtained through a bi-genetic algorithm, where the first and second level of the optimization are represented by a "leader" algorithm and a "follower" respectively; both of them are based on the operators used in genetic algorithms: crossover, mutation and selection.

In [22] the procedure described by the paper consists in a combination between trajectory planning in the Cartesian space and planning in the joint space. Firstly the set of samples of the desired path is interpolated with a cubic spline whose parameters are obtained by imposing the initial and final velocity and acceleration to zero. Inverse kinematics gives the same trajectory in the joint space, where the intermediate points are interpolated with a 7th order B-spline. The control points of the spline are used to translate the constraints on kinematic parameters. The minimum execution time is the object of the optimization problem solved with the SQP method. The advantage of using cubic and septuple splines is valid as long as there is an accurate choice of the samples density in the path, that should not be too high or too small.

In [23] the study proposes the use of multi-objective genetic algorithms to solve the inverse kinematics in case of redundancy. The considered algorithm uses a "non-dominated" sorting procedure, which means that the genetic population is divided into sets where the solutions are non-dominating, namely no solution performs better than the others in one or more objectives and as good as the others in the remaining objectives. The aim is to find a trade-off between the best solutions for different objective functions: the first one minimizes the distance between the current joint position and the initial joint position in order to make the joint configurations repetitive, the second one minimizes the error between the actual position and the desired position of the end effector. This method is a good solution also for obstacles avoidance

In [24] the presented algorithm is a mathematical solution for the minimax problem of a trajectory planning procedure whose main objective is jerk-minimization. The optimization formulation is: find the minimum among the possible maximum jerks at each spline time, given the constraint on the total time, which is fixed. The value of the jerk is constant since a cubic spline has been used for the interpolation of the points obtained with inverse kinematics. The best condition to exploit the algorithm is the off-line planning, expecially in automation systems where the total operation time is constrained.

1.2.2 Soft and flexible robotic manipulators

In the second part of the present work, the investigation of the problem of time-minimization lead to the developement of new ideas to solve the base issue without using industrial robotic arms, but looking for a more compact, cheap and fast solution. The orientation of this part of the work is towards the soft and flexible robotics field, touching also the topics of biomimetics, finding inspiration in the fastest animal that can grab a pray in nature: the chameleon.

In [25] a mathematical model of the ballistic mechanism of the chameleon tongue is developed. The main processes involved are the activation of the accelerator muscles to produce mechanical energy, the storage of elastic energy in the intralingual sheaths and the release of this energy resulting in the ballistic projection. The parts of the tongue are modeled as concentric cylinders.The intralingual sheaths are made of collagen and represent the elastic part of the system.
The phases of the overall process can be resumed in the rest position starting point, then the contraction of the accelerator muscle squeezes the sheaths in a loaded position and a further contraction of the muscle cause the sheaths to extend telescopically and fire the internal bone. The mechanical equilibrium of the system is computed considering the sum of the energy density of the intralingual sheats and the energy density of the accelerator muscle, then the energy must be minimized for some given reference parameters and contraction in order to solve the problem. The dynamic model of the energy release considers the extensible tongue as a series of connected springs and a force that pushes them in the axial direction.

In [26] the study develops four different bioinspired solutions, each aiming to imitate different aspects of the ballistic projection of the tongue of a chameleon in the process of catching a prey. The implementation of the experiments and its modelling is based on the concept of coilgun. The point on the tip of the tongue is realized through a magnet which is fired by a reel actuated by the current supply. The magnet is fixed on a short elastomer, which is connected in turn to a cotton string, in Fig.1.11 the phases of the shooting system can be observed.
The retraction of the magnet is due both to the elastomer and to a DC motor that rewinds the string and brings the tip close to the initial position. In the original variant of this experiment the string was substituted by a longer fixed elastomer without DC motor, and a further variant used the coilgun-DC motor-elastomer structure and added some wings to demonstrate the possibility of exploiting aerodynamic effects to control the displacement of the tip.

In [27] the two prototyped manipulators are similar: the first has the aim of quick capturing a moving target through the shooting of an end effector that

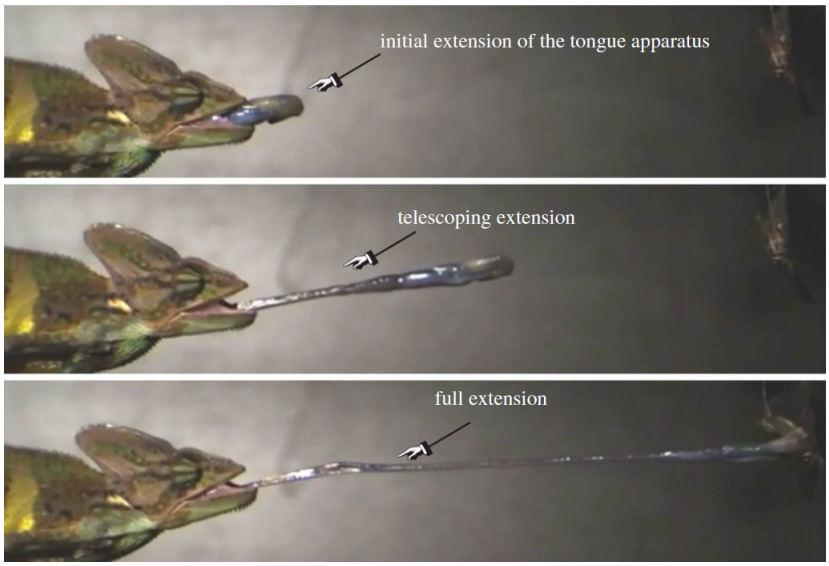


Figure 1.9: Chameleon catching the pray. Images courtesy of Stephen Deban (online)

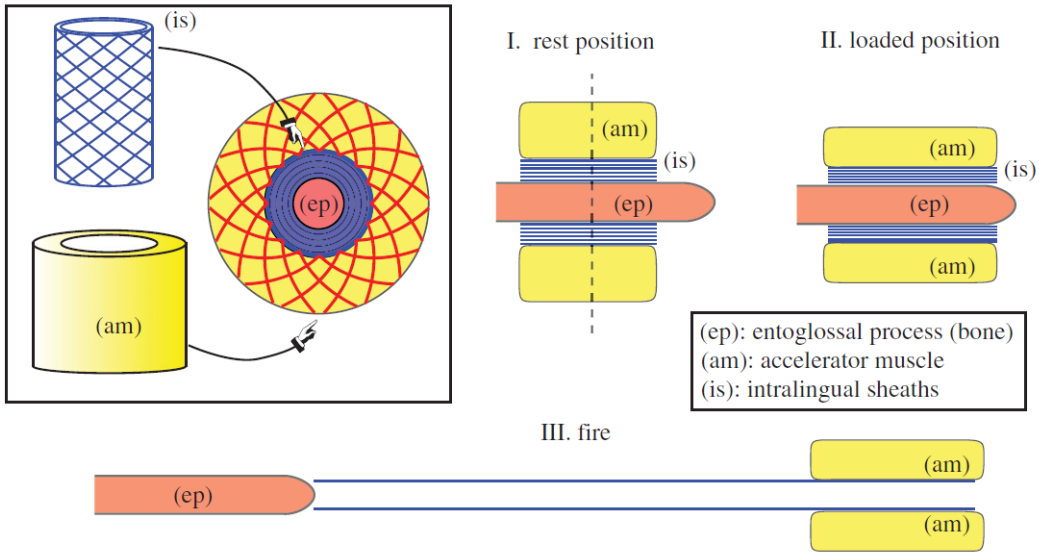


Figure 1.10: Chameleon tongue model. Inset: cross section of the tongue complex composed of the entoglossal process (bone), intralingual sheaths (with collagen fibres) and accelerator muscle (with helical muscular fibres). Images courtesy D.E. Moulton ([25])

consists in a nylon cap and a magnet. The shot is done by means of a high pressure source and an elastic cantilever as constraint. The cantilever stores the

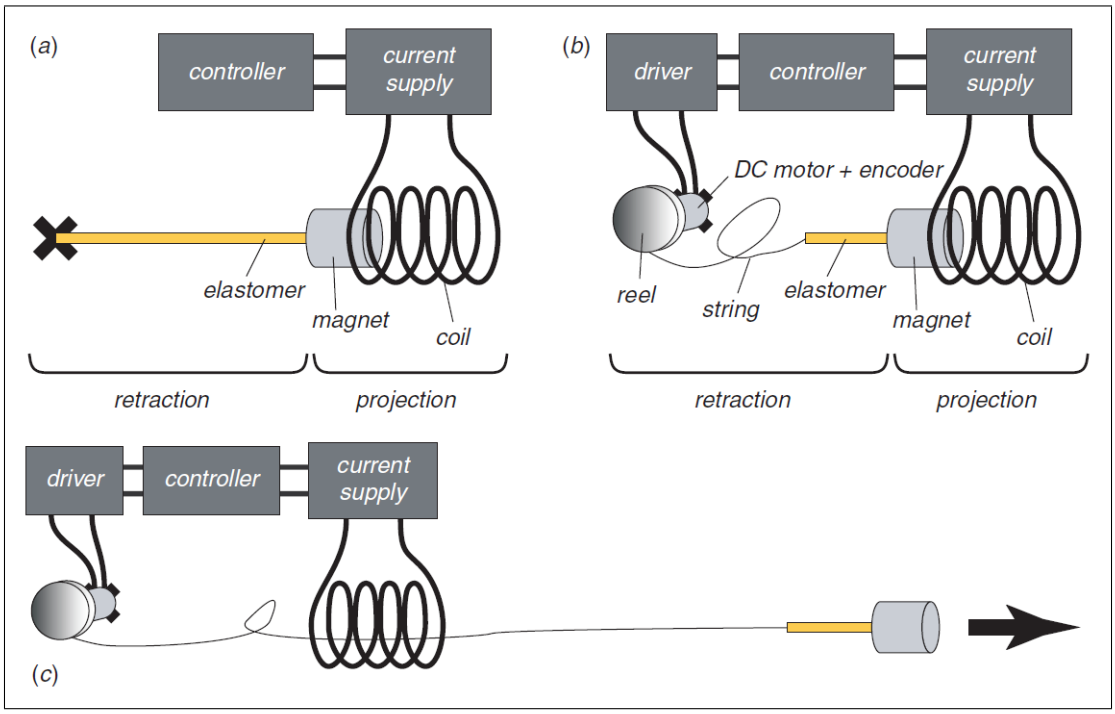


Figure 1.11: Courtesy of [26]. (a) Scheme of the coil-gun–elastomer system at rest, (b) scheme of the coil-gun–dc–elastomer system at rest and (c) scheme of the coilgun–dc–elastomer system during the projection phase.

kinetic energy and releases it when pulling back the end effector. The second manipulator is able not only to reach a target in linear motion, but has also the intent of reaching blind spots placed besides obstacles. In this case the constraint is an inertial wheel that rotates passively, as can be seen in the picture of the prototype built for the experiments in Fig.1.12.

To study the motion of the end effector, a 2D model is derived (and the motion is divided into multiple phases: uniform forward motion, before the string reaches its maximum length, in this phase the tension of the string applied on the end effector is zero; elastically constrained forward motion; resting at target position and fall due to gravity. The simulations show also robustness of the system with respect to changes of the mass of the end effector up to 10%.

In [28] the manipulator consists in an impulsive force generator that catapunts the end effector towards a distant target. The objective of the study is to test the manipulator for shooting distances higher than the ones in experiments

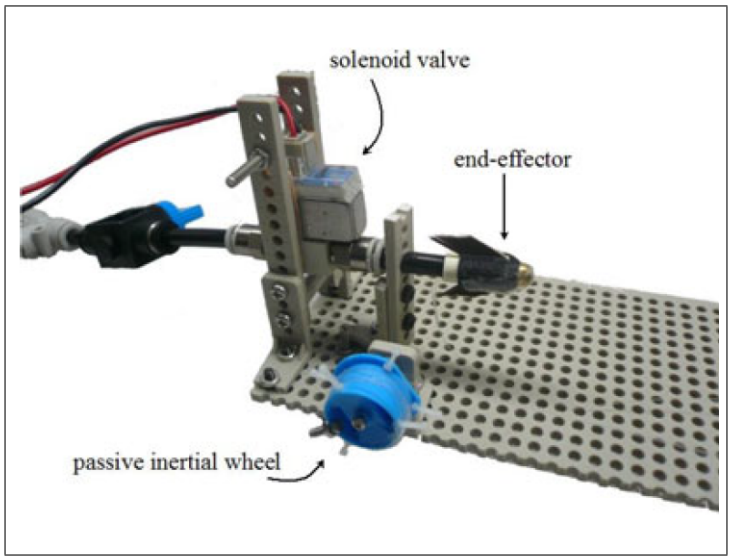


Figure 1.12: Picture of the prototype built in the work of [27].

performed in past studies for similar systems. The pull-back motion is realized thanks to a piece of light-weight viscoelastic string connected to the end effector. A 3D mathematical model is developed for a manipulator that is not connected to a fixed point, but only to an inertial object. The impulse generator used in the experiments is an air gun; the end effector has plastic wings and a carbon passive propeller and is able to reach very high speeds, over 20 m/s.
The motion measured during the experiment was almost linear and with uniform velocity. The prototype has been tested with and without the passive propeller, showing how this element largely increases the reaching accuracy and allows to increase the shooting distance, at the same time it introduces variability in the orientation of the end effector near the target point.

In [29] the presented casting manipulator is composed by rigid links, a string whose length can be modified, and a gripper. The system is modelled through a two-link planar manipulator with an actuator in the first joint and a second passive joint connected by a wire. The casting motion constists in a first phase of stable periodic swinging, then the gripper is thrown to the target. The focus of the study is to control the motion of the gripper after throwing it, in order to expand the workspace of the manipulator: this is possible thanks to the use of an impulsive braking force transmitted by the string to the gripper, Fig.1.14b. This kind of control is actuated through a solenoid that activates the braking at the desired time determined by the position of the target. The technique also allows to reach hidden positions in presence of obstacles, by applying multiple

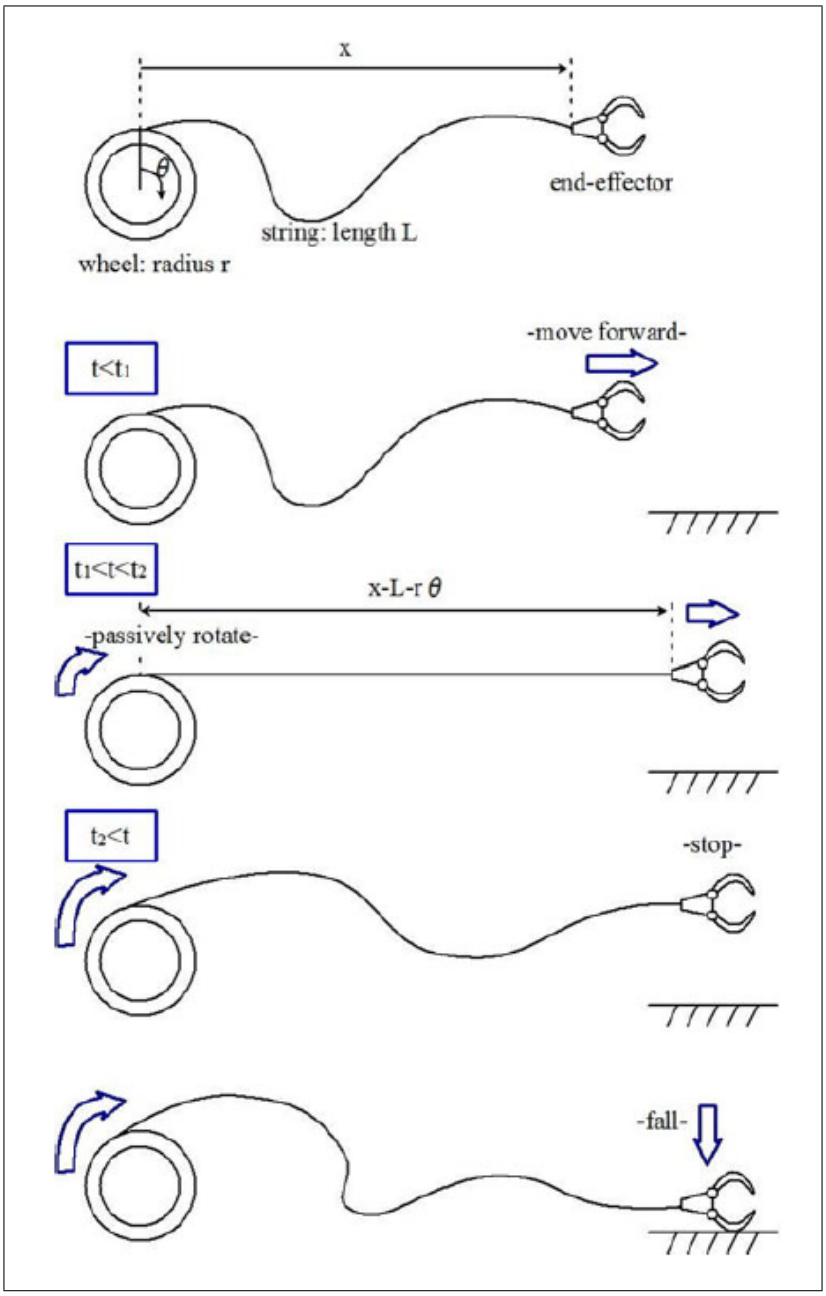
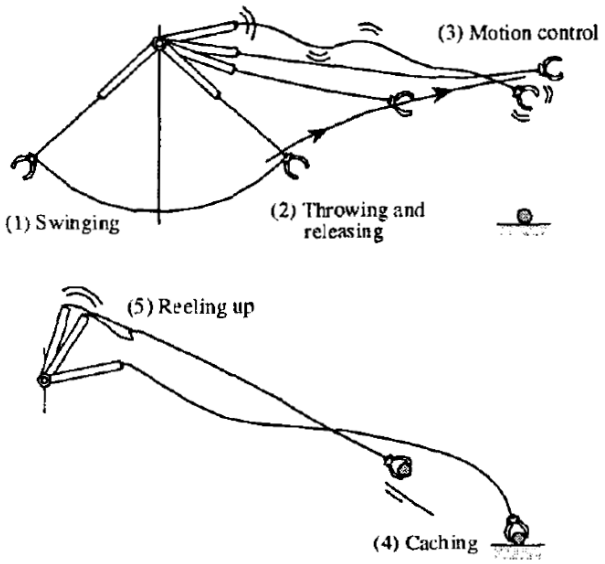


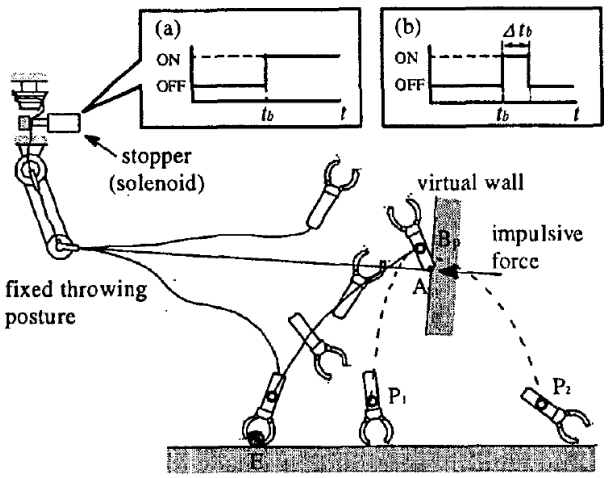
Figure 1.13: 2D model of the shooting manipulator. Courtesy of [27].

braking control. The control law is then optimized by the minimization of a cost function representing the distance between the final state and the specified state.

A general review on soft robotics can be found in [30]. Soft robotics in



(a) Phases of casting manipulation.



(b) Midair gripper motion control by braking the string.

Figure 1.14: Images courtesy of [29]. Representation and control phases of a casting manipulator.

industrial environments is useful to break the separation of human and robotic workspaces, since the soft nature of the materials guarantees safety in the event of collision. Aside of this advantage, soft robots also show a great adaptation ability, a large number of degrees of freedom and continuum deformations. The

"soft" term refers to the materials composing the body of the robot, that are characterized by a Young's Modulus that is in the same order of the one of soft biological materials such as skin and muscle tissue, that is $10^4 - 10^9 Pa$.
Actuation is usually pneumatic or consisting in variable length tendons. Biology is the inspiration of many soft robots and it influences the actuation systems that have frequently an agonist-antagonist bi-directional arrangement. The challenge of pneumatically actuated robots is the limit of the high strain that leads to low actuation rates to avoid rupture failures.

The study in [31] aims to find a model for soft robotics manipulators that is geometrically exact, since the assumptions of the known basic linear models neglect important aspects that lead to large position prediction errors if applied to larger deformations. Pneumatic artificial muscles consist of flexible rubber tubes encased in a braided mesh sleeve that are pressurized to actuate the manipulator. The model is based on the work-energy principle and takes into account also the nonlinearity of the membrane and the mesh angle change, whilw validation of the results is done on the robot Octarm V.

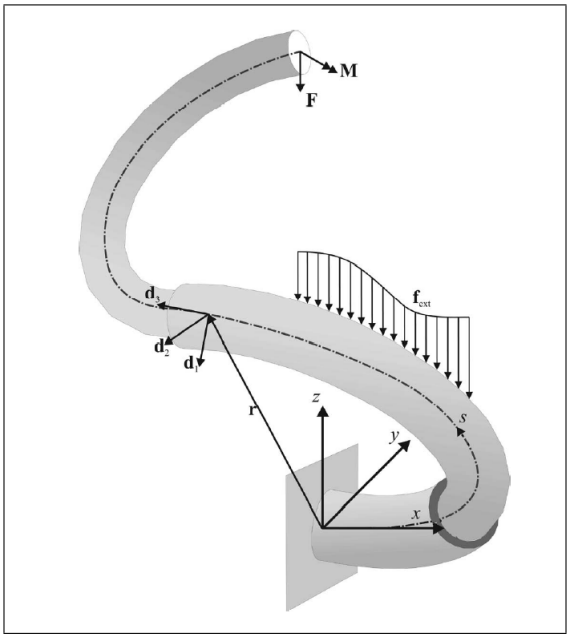


Figure 1.15: Image courtesy of [31]. Soft robotic manipulator model.

A soft flexible manipulator is presented in [32], it was prototyped with the objective of easily controlling a robot through an unstructured environment. The manipulator is a continuum robot composed of two cylindrical balloons and an extensible sheat used to constrain the two balloons to be close together.

By inflating the balloons, the robot extends and a force is applied to the tip, while the balloon has an inflated first section and a slack part. The difference of air volume in the two balloons results in the bending of the structure, making it easily controllable in presence of obstacles. The prototype has been tested for multiple inflation and deflation cycles showing that the maximum elongation does not change much after the first cycle, but the last stages of deflation present a stability problem resulting in two inflated sections with a slack part in the middle.

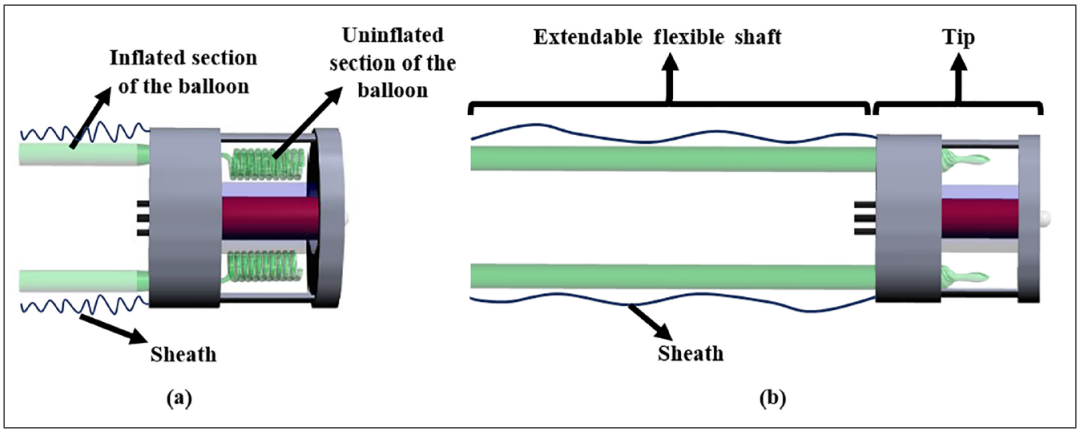


Figure 1.16: Courtesy of [32]. Schematic representation of the robot and the inflation process. (a) Uninflated robot in the initial condition, (b) inflated robot.

In this work, also flexible elastic manipulators will be taken into account for fullfilling the pick and place task. Such systems are generally modelled throug the beam theory [45]: each link of the robot is modelled as a beam or a beam-like element and the beam dynamics is used to analyze the dynamic behavior of the whole flexible system. A representation and force analysis of a beam can be observed in Fig.1.17.

The modelling of flexible robots is more complex than the rigid robot case because of the presence of an additional number of degrees of freedom due to deformation. These degreed of freedom depend on the model used to analyze the beam, Euler-Bernoulli is the easiest, but more advanced methods such as the finite element methos introduce a very high number of total degrees of freedom and relative vibrational modes. Luckily, many of these degrees of freedom can be removed by imposing constraints on the direction of deformation of the beams and many modes can be considered neglectable. Only a limited number of mode shapes can be considered in the dynamic analysis, an the choice of which modes to analyze has a big impact on the final model.

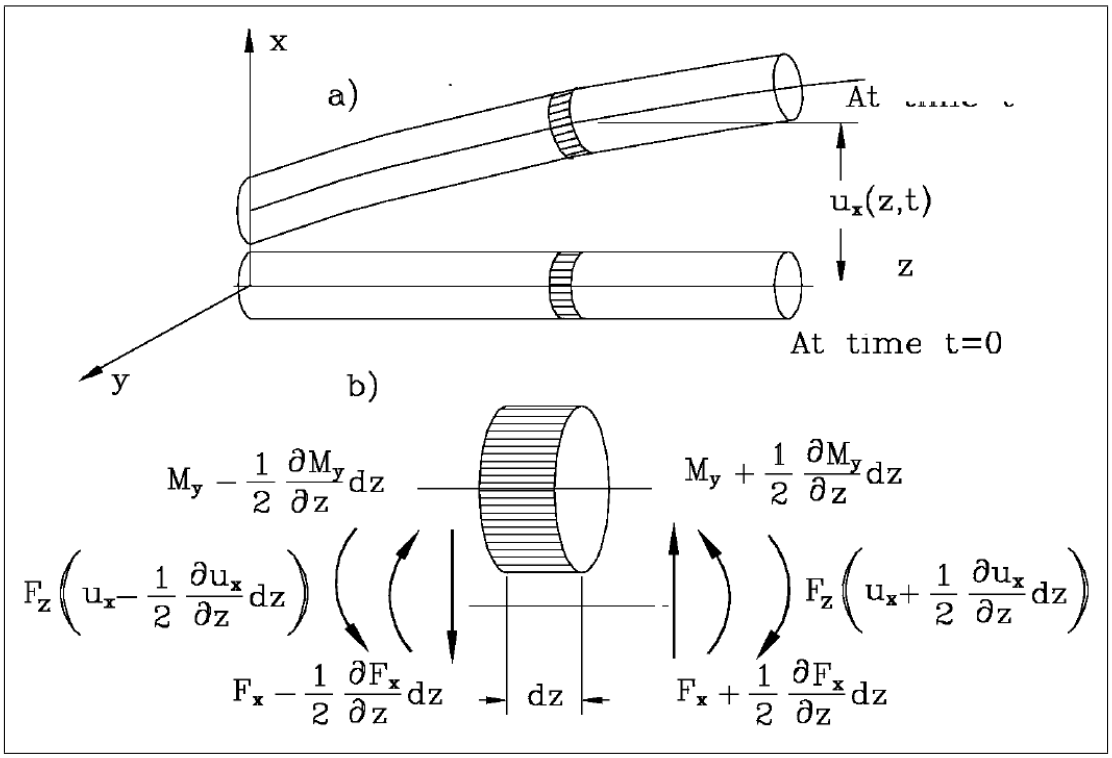


Figure 1.17: Courtesy of [44]. Beam element section forces and moment analysis for bending behavior.

Another key element when the control has to be applied to a real mechanical structure is system identification. As a matter of fact, the models are only approximations of the real system and sometimes these two entities behave in different ways, therefore the model has to be compared to experimental data to prove its effectiveness. System identification can be helpful to refine the model or even more when the real system contains phenomena that cannot be modelled with the theoretical knowledge on the matter. Moreover, the robot actuators have their own dynamics that adds a level of complexity to the total plan to be controlled.
Dynamics are of crucial importance in the control theory of flexible systems because of the vibrations that affect the transient conditions. These vibrations must be reduced and compensated with an appropriate controller.

A review on general multilink flexible manipulators modelling and control can be found in [36], where different methods are compared to deal with the oscillating behavior of such structures, for example Proportional-Derivative control, Input-Output linearization, active damping, adaptive control, neural networks, Lead-Lag control, output redefinition.

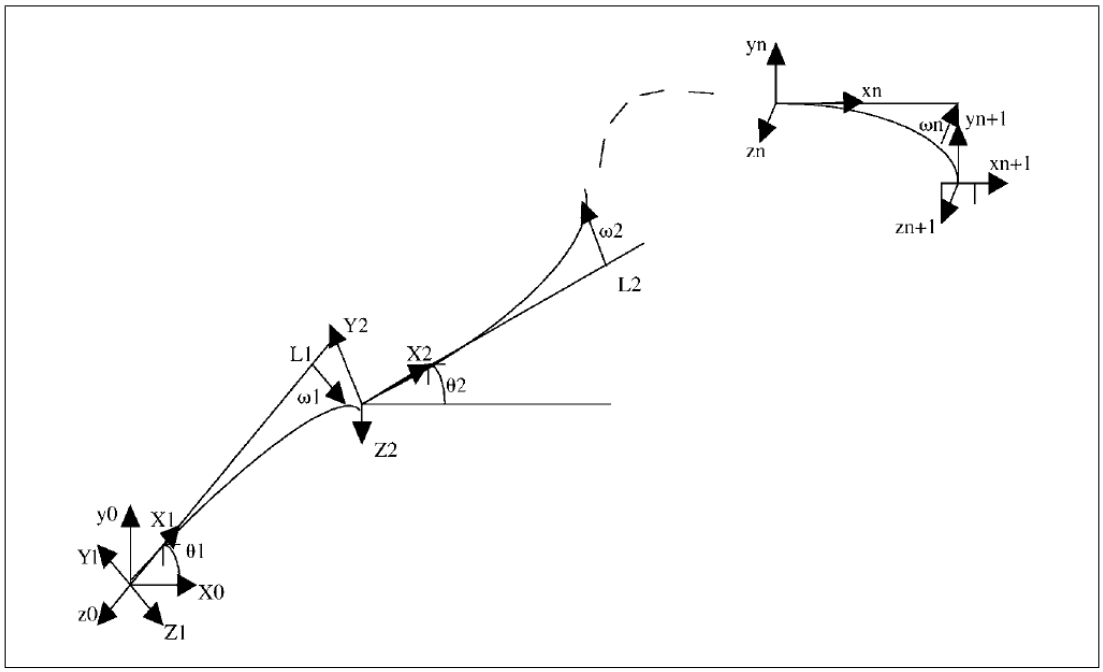


Figure 1.18: Courtesy of [36]. A general representation of multi-link flexible manipulators.

The general model considered for the system is in the lagrangian form, i.e. the form in equation Eq.1.1 expressed in the most general form. In several works the PD control has been proven to work well if applied to collocated actuators and also without taking into account the damping term in the dynamic equation. q is the vector of generalized coordinates, u the input vector, h_c the vector of centrifugal and coriolis effects, M is the inertia or mass matrix, which is symmetric and positive definite, K is the stiffness matrix, D the damping matrix, W is the input weights matrix and $g(q)$ the gravity vector.

$$M(q)\ddot{q} + h_c(q, \dot{q}) + Kq + D(q)\dot{q} + g(q) = Wu \tag{1.1}$$

The computed torque method is another control option. It is achieved with state feedback IO linearization and is strongly dependent on co-location of actuators and sensors. If the two are not collocated, stability cannot be assured. Therefore this scheme is most suitable for joint trajectory tracking because the sensed output is the joint position and is collocated with the actuatior input, that is joint torque.
Active damping is a method that allows to deal with tip oscillations when the actuators are placed in the joints and subsequently cannot provide an effective

compensation to the problem. The general control law is modified by adding linear terms proportional to velocity.

One of the techniques for vibration control and reduction is called input shaping, it is an open-loop control technique and it is described extensively in [45]. Input shaping filters are used to generate a sequence of impulses appropriately spaced and fed to the plant to cancel its vibration. The impulse frequency depends on the frequency of vibration of the system. The input command that is being fed to the system that makes it oscillate in the first place is convolved with the impulse command and sent as input to control and at the same time stabilize the system. The new convolved input is called the shaped command. It is a good technique for the cases where the closed loop control parameters cannot be modified, because it depends directly on the frequency at which the system is vibrating in a certain moment.

An example of the use of input shaping techniques is [42] where the autors control a two-link flexible-joint manipulator by combining input shaping and feedback control based on Adaptive-Parameter Auto Disturbance Rejection Controller (APADRC). This choice was driven by the fact that input shaping is not enough to cancel residual vibrations and vibrations due to external disturbances since its ability is only to cancel vibrations due to the control command.

Command smoothing is another technique used to suppress vibrations. An example of its application can be found in [43]. Here, the original command is smoothed basing on the system dynamic parameters, that are natural frequency and damping ratio. The method was tested experimentally on planar single and double pendulum bridge cranes and its performance was compared to a shaping technique, in particular a two mode zero vibration double derivative shaper. The robustness of the two methods was compared, i.e. their sensitivity to model parameters variation, and the result was that the smoother was more sentitive at higher frequencies while the shaper was more sensitive at lowe frequencies.

Chapter 2

Trajectory Optimization

2.1 Present technologies of the case study

The tests currently pursued at the Advanced Industrial Automation Laboratory of IIT make use of an industrial robotic arm with six revolute joints. The robot is ABB IRB 1600, Fig. 2.2., whose technical data can be found at [33] .
The kinematic model of the robot was represented in Matlab as a "Rigid-BodyTree" object by means of the Robotic Systems Toolbox. The method used to define the kinematic chain was the Denavit-Hartenberg convention, Tab.2.1. The graphical representation can be seen in Fig. 2.1.

a	α	d	θ
0.15	-$\pi/2$	0.4865	0
0.475	0	0	0
0	$-\pi/2$	0	0
0	$\pi/2$	0	0
0	0	0.065	0

Table 2.1: Denavit Hartenberg parameters of the robot kinematic chain

This model is needed to exploit the many functions of the toolbox, including the inverse kinematics and inverse dynamics algorithms. The most suitable trajectory planning method for the present task is planning in the joint space, this choice will help to create the smoothest trajectories and exploit the most the joints in terms of speed, allowing to reach the minimum task time posiible.

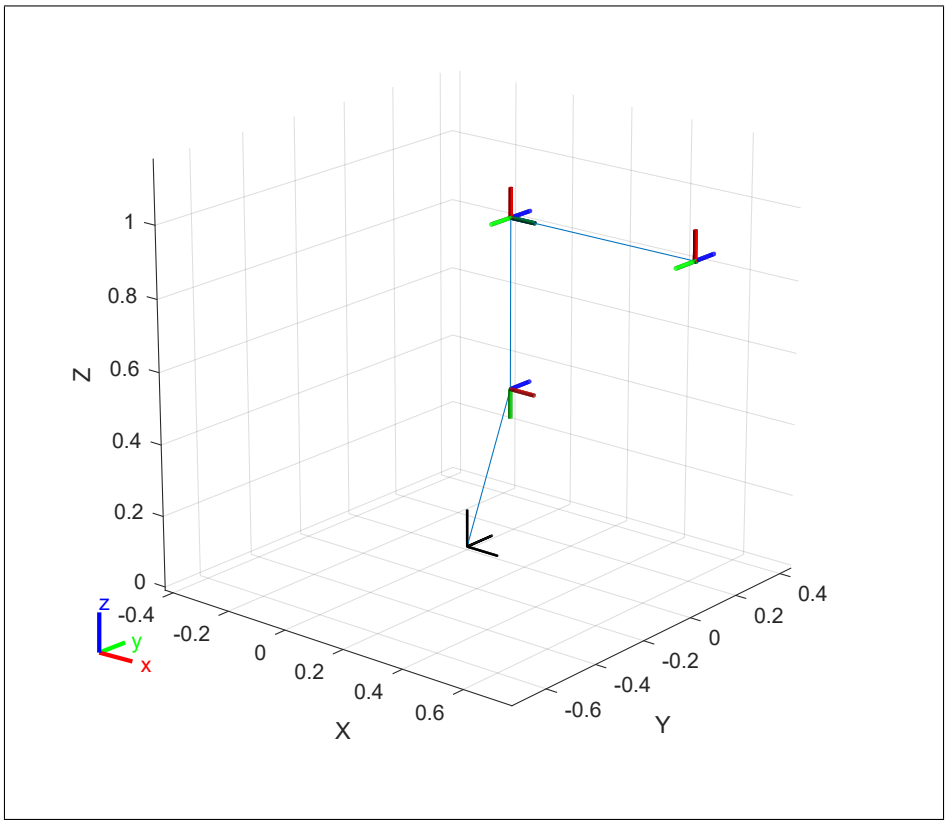


Figure 2.1: Representation of the robot links and joints given by Matlab Robotic Systems Toolbox

2.2 Path planning

The path planning procedure starts with the TCP positioned on the first conveyor belt.
Firstly the path is defined, then a certain number of samples are taken to compute the robot configuration in each of them through inverse kinematics.
The number of path samples varies depending on which kind of motion is considered: constant velocity motion or point to point motion. Constant velocity motion is a task that requires full control of the TCP velocity vector in each instant and theoretically the best way to deal with it is planning its trajectory directly in the task space.
In the present application the task does not need such precision and a good approximation is obtained by planning the whole trajectory in the joint space and taking a high number of samples during the constant TCP velocity phases.

Four different phases must be considered for a complete cycle of the task, as illustrated in Fig. 2.3:

Figure 2.2: Picture of the industrial robotic arm model ABB IRB 1600

- First phase: the robot TCP follows the target moving on the conveyor belt in a path geometrically linear, this is necessary to make the TCP velocity equal to the velocity of the target and eliminate the relative velocity between them. At the end of this phase the sucker on the gripper is actuated and picks the object.

- Second phase: the robot transfers the object to a point above the second conveyor belt. To do so, it should bring up the TCP to operate to a safe distance from the belts and ground, and the following path in the simplest case can be modelled as a point-to-point path. Nevertheless, to improve the smoothness of the motion, two more waypoint samples have been added, belonging to an elliptical path whose ends are initial and final position, for a total of four configurations for this phase; smoothness of the trajectory is a very important quality in in high-speed robotic operations, since small

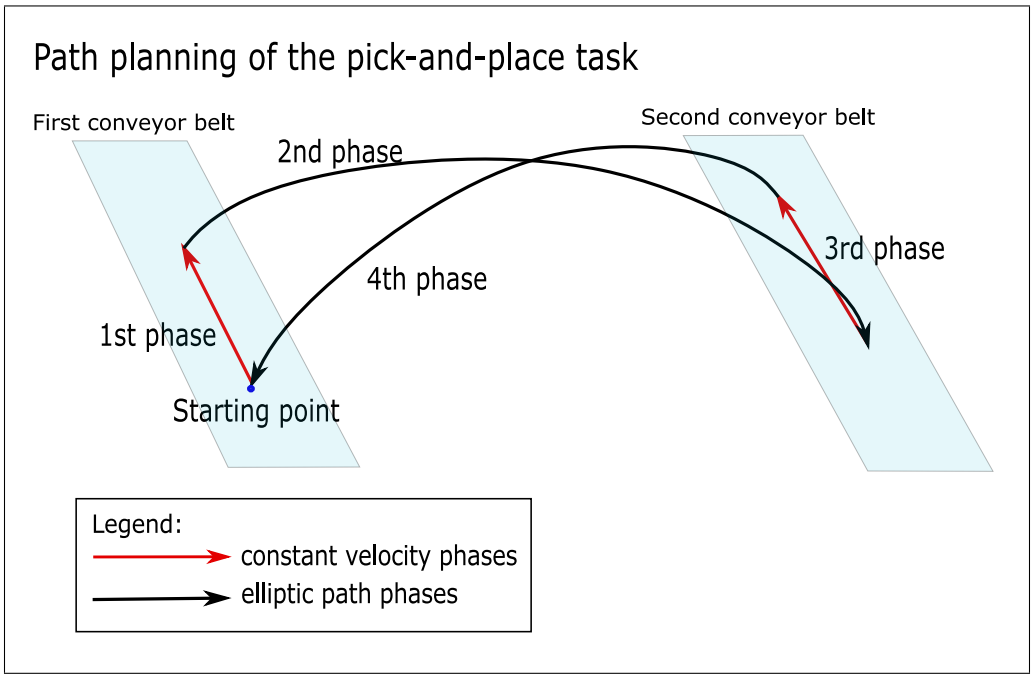


Figure 2.3: Different phases of the TCP path during the pick-and-place task

sudden variations in the velocity and acceleration profiles can cause abrupt variations in the torque profile, which is dangerous for actuators and can lead to vibrations and jerk.

- Third phase: the robot keeps holding the object while moving along the second conveyor belt to nullify the relative velocity between the belt and the object before dropping it. This phase is similar to the first but the motion is in the opposite direction.
- Fourth phase: the robot goes back to the beginning of the path to pick the following object. This part consists of another elliptic section with the motion in opposite direction with respect to the second phase.

Between each linear motion phase and elliptic phase, in the path definition there is a transition section where no samples are taken in order to to compute the corresponding trajectory parts directly in the joint space, by interpolating initial and final points of the two sections. This allows to obtain a smooth transition from the linear path to the lifting and transfer of the object without having to fully specify the path in these sections.

The first consideration about this method of planning is that it belongs to the field of tracking techniques, i.e the methods whose aim is to minimize the

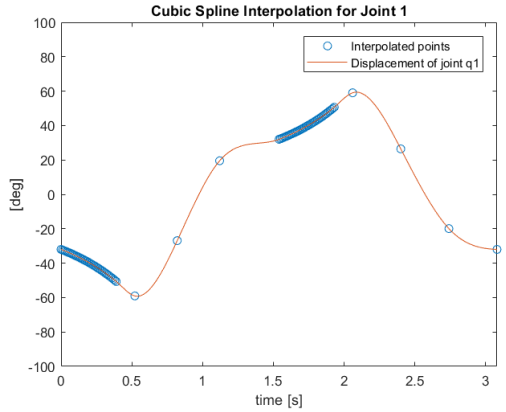
difference of velocity and position between target object and TCP over the control period [48]. The rendezvous point in this situation is not known until the interception instant, in opposition to the technique seen in [46] called APPE, Active Prediction Planning Execution, whose key feature is the possibility to select a desired rendezvous point. The drawback of APPE is the necessity of an accurate prediction of the position of the object to be grabbed, in order to compute the rendezvous point, therefore a more complex set of sensors is needed.
In the approach chosen here, i.e. the tracking technique, the available time to activate the gripping device, for example a suction gripper, is spread over the constant velocity phase when the robot TCP and the object are travelling at the same linear velocity. The exact interception point does not have to be known a priori, or predicted.

2.3 Trajectory planning

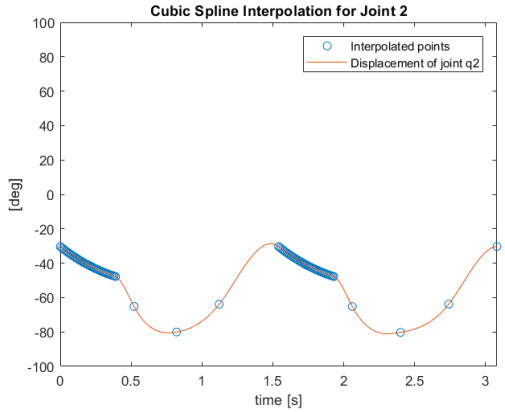
Trajectory planning in this specific task is pursued in the joint space. Therefore, for each sample of the path previously defined, inverse kinematics must be computed to obtain the robot joint configuration corresponding to the sample cartesian coordinates.
An inverse kinematics algorithm is part of the Matlab Robotic Systems Toolbox and it's generated for the specific "RigidBodyTree" object by the function *robotics.InverseKinematics*. The algorithm is solved by the function *step* which starts its iteration from the cartesian coordinated of the target point and an "initial guess". The initial guess is a joint configuration representing the first solution of the inverse kinematics problem, that has still to be updated by the algorithm during each iteration in order to reach feasibility. Generally, when no additional information is available about the joint configuration of the desired position, the initial guess is set as the home configuration of the robot (*robot.homeConfiguration*).
As demonstrated in [3], the best spline method for interpolation with strict kinematic constraints are cubic splines, for this reason they were chosen here to interpolate the path samples in the joint space.
Matlab provides the function *spline* for cubic spline interpolation, allowing also to set initial and final slope of the curve. This feature will be useful in this case to fix the initial and final velocity of the profile, expecially when starting with the constant velocity phase.

The geometric Jacobian is used to link the geometric velocities to the joint velocities [1]. In the present case it can be exploited to compute the joint velocities corresponding to the fixed TCP velocity at the beginning of the linear

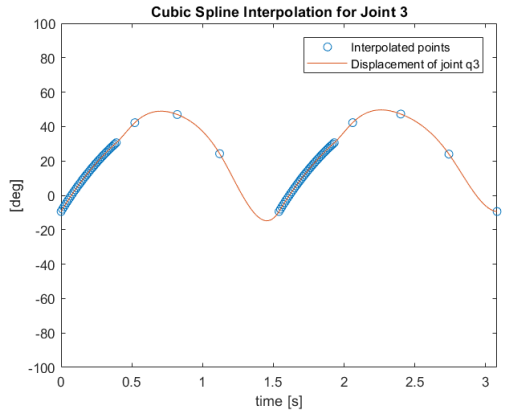
path, in order to obtain a motion with no abrupt change between the cycles. To get this result, it is necessary that initial and final velocity for each cycle are the same. After computing the wanted starting velocity, it can be fed to the *spline* function together with the joint variables samples and their respective time vectors to get as output the interpolated curve. At the same time it is not necessary to compute the initial and final velocities of the wrist, it will have the time to settle during the linear part, reaching the wanted velocity.



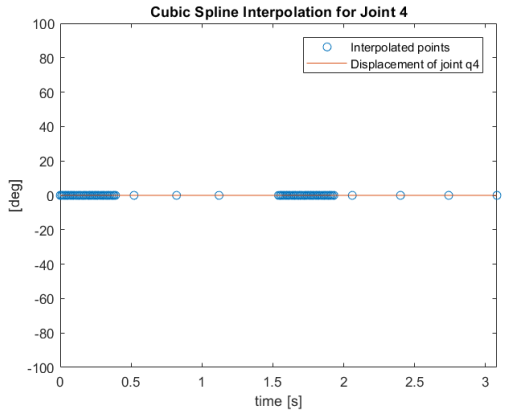
(a) 1-st joint variables



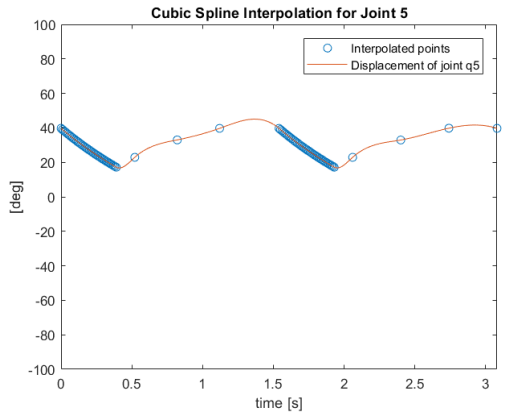
(b) 2-nd joint variables



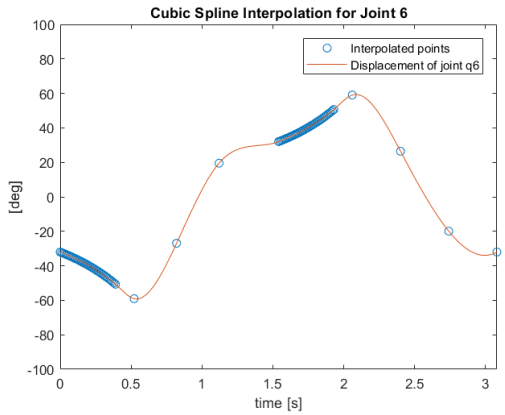
(c) 3-rd joint variables



(d) 4-th joint variables



(e) 5-th joint variables



(f) 6-th joint variables

Figure 2.4: Path samples interpolated with cubic splines in the joint space: the samples are more packed in the section of the linear path and less packed in the elliptic path

Chapter 3

Developement of novel solutions

In the studied case, the velocities of the two conveyor belts are considered to be equal and the screw conveyor rotates with a speed such that the linear velocity of the carried items is synchronous with the veocity of the second belt, since the items are actually laying on the second belt. If v_1 and v_2 are the velocities of the two conveyors, as represented in Fig.1.1, p is the pitch of the screw and n is the speed of the screw in revolutions per second, we have:

$$v_1 = v_2 = v \tag{3.1}$$

$$n = \frac{v_2}{p} = \frac{v}{p} \tag{3.2}$$

Therefore the speed of rotation of the screw is fixed, given the pitch.

3.1 Choice of the manipulators

The general set up common to all the systems that have been designed includes one or more manipulators whose aim is to extend, grab the bottles and then retract.
This kind of solution can solve many of the initial problems concerting the present layout: the need of high speed of the cycle (grabbing and positioning) and the physical encumbrance of the screw conveyor, which makes the standard robotic solution inapplicable for the presence of the bulky gripper.

3.1.1 Shooting manipulator

One of the analyzed possible systems recalls the ones studied in [27] and [28].

When these systems are applied to the considered practical scenario, one of the first problems that arise is the gripper choice: the use of a pneumatic sucker is challenging to implement in this case, due to the presence of the air tube necessary to actuate the sucker. Indeed this kind of air tubes are too stiff and thick to be connected to the end effector in the phase of shooting without affecting largely its motion, and most of all they cannot be rewinded during the pull of the bottle towards the second conveyor.
The end effector must be connected to a wire with no other forces acting on it. A solution might be the use of a gripper that grabs the bottle by closing authomatically at the moment of contact. This kind of gripper is obviously applicable only in the case of small bottles or, more in general, small objects, because its volume and weight must remain restrained in order to limit its influence on the shooting trajectory.
The second problem is the perturbation of the position of the bottles, which may require the end effector to change its shot angle. The shooting system should be equipped with two rotating joints for pitch and yaw control. These joints will respond to a control system that elaborates the visual information received by a robotic camera, compute the target position at the rendez-vous point and orientate the shooting direction consecutively. This part represents the main control part and will require a model of the shooting manipultor that will change in each of the configurations that will be presented in the next sections.
After the object catching and the subsequent rewind phase, the release instant should be computed depending on the screw teeth position, to ensure that the object is dropped in the right place. To this phase follows the final rewind to reposition the gripper in the starting point for next shot. To study the behavior of the manipulator, it has been modelled in 2-D in [3], as represented in Fig.3.1. The shot is done by means of a high pressure source and an air flow applied at time t_0, whose impulsive force is expressed by I_{imp} and the shooting angle is ϕ. L and k are the length and spring constant of the string. The constraint is an inertial wheel that rotates passively, with inertia I, radius r and rotation angle θ. T is the tension of the string and m is the mass of the ned effector.

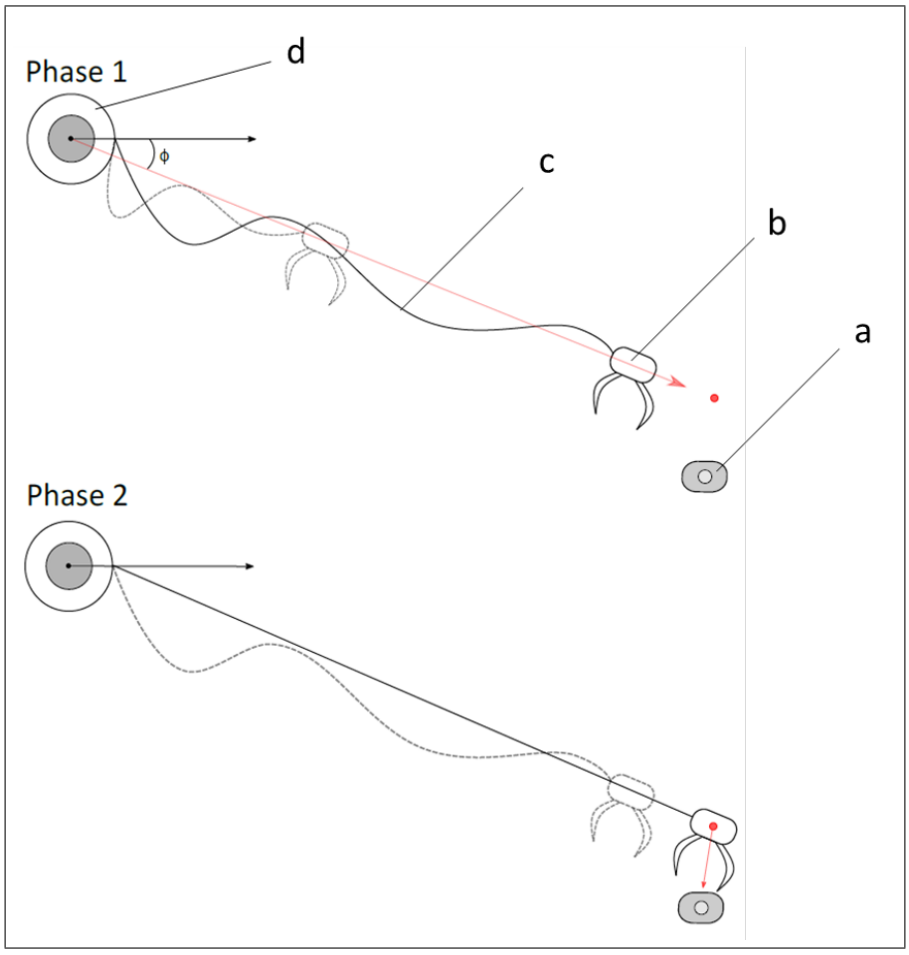


Figure 3.1: 2-D representation of the shooting manipulator

The equation of motion is:

$$m\ddot{x} = -T \tag{3.3}$$

$$m\ddot{y} = -mg \tag{3.4}$$

$$I\ddot{\theta} = rT \tag{3.5}$$

$$x(t_0) = y(t_0) = \theta(t_0) = \theta(\dot{t}_0) = 0 \tag{3.6}$$

$$\dot{x}(t_0) = cos\phi\frac{I_{imp}}{m} \tag{3.7}$$

$$\dot{y}(t_0) = -sin\phi\frac{I_{imp}}{m} \tag{3.8}$$

$$T = \begin{cases} k\Delta l, & \text{if } \Delta l \geq 0 \\ 0, & \text{if } \Delta l < 0 \end{cases} \tag{3.9}$$

$$\Delta l := (x - L) - r\theta \tag{3.10}$$

In the uniform forward motion subsequent to the shot, the tension T of the string is null, therefore the time law is:

$$x(t) = cos\phi \frac{I_{imp}}{m} t \tag{3.11}$$

$$y(t) = -sin\phi \frac{I_{imp}}{m} t - \frac{1}{2} g t^2 \tag{3.12}$$

$$\theta = 0 \tag{3.13}$$

After the string starts to be in tension,supposing that ϕ is small, the equation of motion is:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & 0 & \frac{rk}{m} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{rk}{I} & 0 & -\frac{r^2 k}{I} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{3.14}$$

$$\ddot{y}(t) = -mg \tag{3.15}$$

By imposing the boundary conditions, the time $t_1$can be computed.

$$x(t_1) = L \tag{3.16}$$

$$\dot{x}(t_1) = cos\phi \frac{I_{imp}}{m} \tag{3.17}$$

$$y(t_1) = -sin\phi \frac{I_{imp}}{m} t_1 - \frac{1}{2} g t_1^2 \tag{3.18}$$

$$\dot{y}(t_1) = -sin\phi \frac{I_{imp}}{m} - mgt_1 \tag{3.19}$$

$$\theta(t_1) = \dot{\theta}(t_1) = 0 \tag{3.20}$$

The following phase consists in resting in the target position and then fall: in this moment the gripper will grab the object, then the rewind phase takes place through a motor connected to the inertial wheel.

3.1.2 Soft robotics

The idea is to use a more precise and controllable manipulator consisting in four inflatable balloons placed inside a sheat, at the tip of the balloons a disk is attached with the gripper fixed on the disk. The structure is similar to the one studied in [30], with some differences. In our case the actual system should

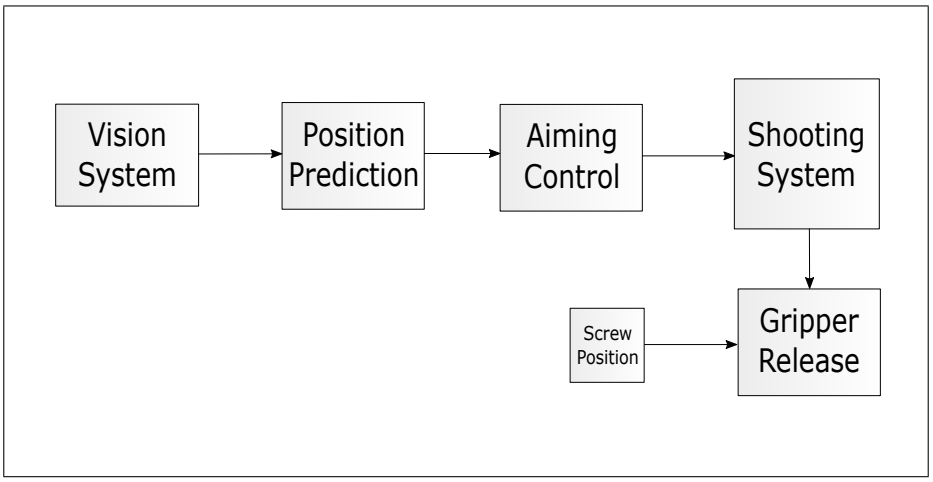


Figure 3.2: Block diagram of the control system

have four inflatable balloons to fulfill the need of one more degree of freedom. The problem of adapting this system to the presented task is the constraint on speed: since in today's state of the art the materials used can bear a relatively low maximum pressure for rupture avoidance, the inflation phase can be performed at a limited speed and this fact affects the total time of execution of the task. The system is worth to be studied anyways, with a view to the possible improvement of the mechanical flexible structure in further studies.
The model considered here is the simplified model with just two inflatable balloons inside the sheath, where thw balloons are modelled as a cylindrical membrane. For a single cylindrical element, the relation between hoop stress and axial stress is:

$$\sigma_\theta = 2\sigma_L \tag{3.21}$$

The relation between axial stretch $\lambda = L/L_0$ and hoop stretch $\mu = \rho/\rho_0$ and between pressure and hoop stretch is:

$$\lambda(\mu) = \sqrt{\frac{1}{2}\frac{K\mu^2 - \frac{1}{\mu^2}}{2K+\mu^2} \sqrt{\left(\frac{1}{2}\frac{K\mu^2 - \frac{1}{\mu^2}}{2K+\mu^2}\right)^2 + \frac{K\mu^{-2}+2}{2K+\mu^2}}} \tag{3.22}$$

$$\frac{p(\mu)}{2|s_-|d_0/\rho_0} = \frac{1}{\lambda\mu^2}\left(\lambda^2 - \frac{1}{\lambda^2\mu^2}\right)\left(K+\mu^2\right) \tag{3.23}$$

For the purpose of this study, the soft manipulator solution has been considered as the least suitable due to the present state of technology that does

not allow a significant improvement in the time reduction of the pick and place cycle. Due to this problem, although the system's high controllability, the simulation and control chapters of this thesis will be entirely dedicated to the other manipulator models presented in this chapter.

3.1.3 Telescopic manipulator

A more rigid manipulator can be considered instead of the two previous ones, that is a controllable cabled telescopic arm, as it can be seen in Fig.3.3. Considering a system with a disk on the tip of an extensible arm, four wires are connected to the perimeter of the disk and are in tension. With the force of the wires it is possible to control the bending of the telescopic arm. On the disk there is the fixed gripper that may be of many kinds.
The system is pneumatically actuated for the extension of the telescopic arm, while the bending due to the cables is controlled by pulling them with electric motors. The more the system is extended, the more it can bend and therefore expand its reachable workspace.
The idea to use an arm controlled this way is very useful referring to vibrational behavior in flexible structures. In such structures, the tip vibration is difficult to control when joints and actuators are not collocated, i.e. command input and measured output are applied on the same point. In this case, the physical system allows co-location since the final output, that is the TCP position, and the command input, that is the bending moment applied by the wires, are both placed on the tip of the arm. This condition allows much more controllability in terms of vibrations.

The mathematical model of the telescopic manipulator is simplified as a beam element with anular cross section, whose tip is subjected to normal forces that are applied by the controlled pulling of the wires. In the following static model, we suppose that the wires apply forces only on one axis, either the x- or y-axis.

Considering the statics of the system, the deflection of the bending beam is given by ([35]):

$$y = \frac{Pl^3}{3E_sI} \tag{3.24}$$

$$I_{x,y} = \pi r^3 t \tag{3.25}$$

where P is the static load placed on the tip of the beam, I is the area moment of inertia of the beam, l is the length of the arm and E_s is the static modulus of elasticity, as explained in Fig. 3.4.

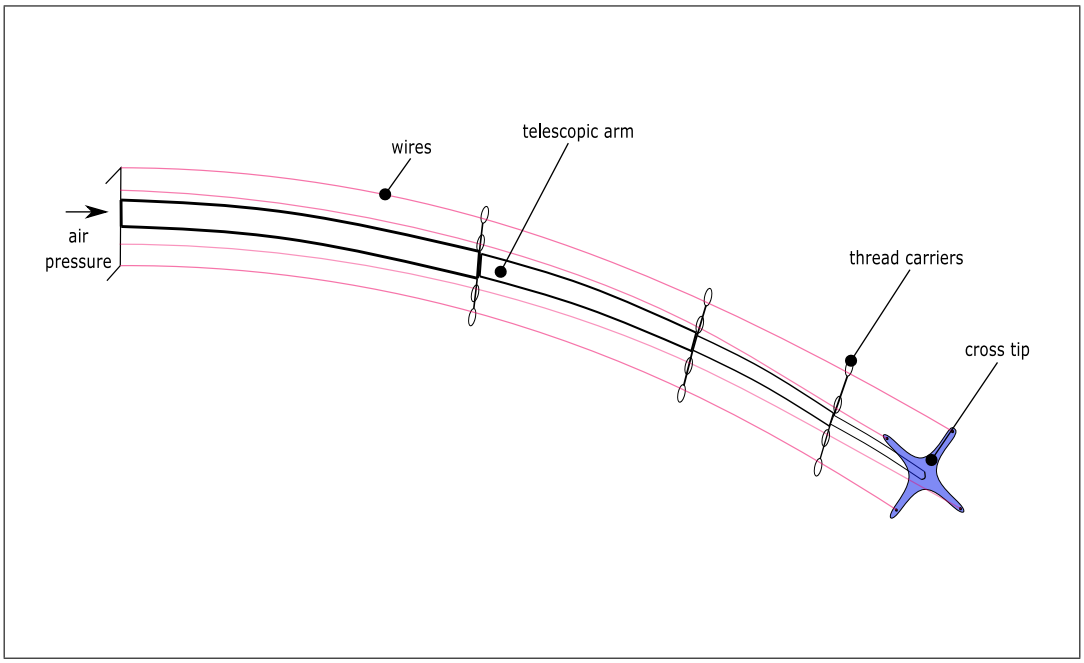


Figure 3.3: Representation of the telescopic manipulator controlled by wires

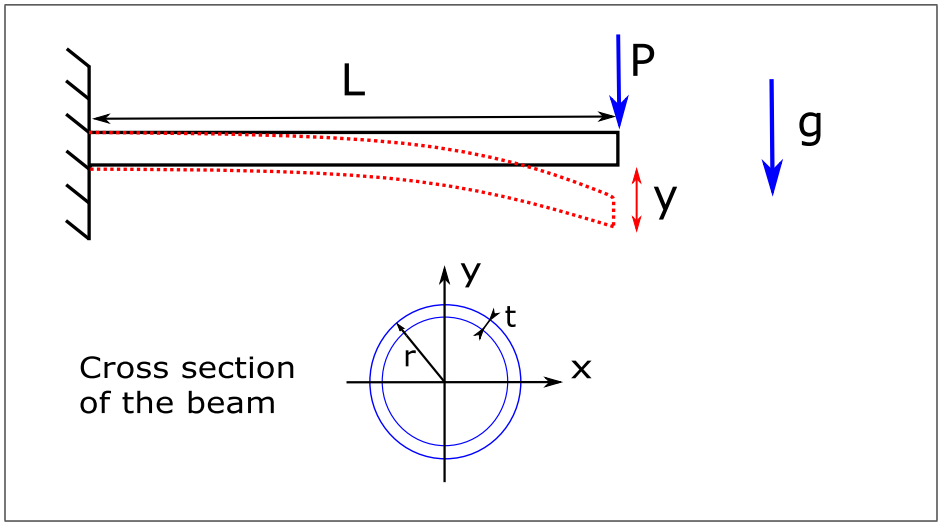


Figure 3.4: Deflection of the simplified beam model of the telescopic manipulator and second moment of area

This model ignores the vibrational behavior of the system, therefore it is necessary to model the system from a dynamic point of view. The mass and

stiffness matrices are considered to be the ones of a straight Euler-Bernoulli beam element with translational inertia, where the rotary inertia is neglectable [37].
A beam is a solid with elastic properties and its main characteristic is that one dimension of its geometry is prevalent with respect to the others. Beams have some frequent properties, such as constant cross section, for which they are called prismatic, homogeneity and straightness [44].
Each beam has six degrees of freedom: three translational and three rotational. In the case of the telescopic arm, the cross section is not constant, therefore the beam is not prismatic, but its length can be divided in a discrete number of constant section segments. Moreover, we can consider a reduced number of degrees of freedom by neglecting the tortional behavior around the principal axis of the beam and considering only the bending behavior in the directions of the pulling wires.
The most simple beam model is the Euler-Bernoulli one, which neglects shear deformations cosidering the cross section always perpendicular to the principal axis of inertia when subjected to transverse loads. This approximpation is applicable in most cases and it's the one that will be used in this study. Two other assumptions are that the consecutive elements of the arm are rigidly connected with no friction phenomena, and that the deformation of the arm is not axial, but only a bending deformation.
In Eq.3.26 we considered for simplicity a single element of the telescopic arm, i.e. a single beam element with constant circular cross section and four d.o.f.:

$$M\ddot{x} + C\dot{x} + Kx = f \tag{3.26}$$

$$x = \begin{bmatrix} x_R \\ \alpha_R \\ x_T \\ \alpha_T \end{bmatrix} \tag{3.27}$$

$$f = \begin{bmatrix} f_R \\ m_R \\ f_T \\ m_T \end{bmatrix} \tag{3.28}$$

$$M = \frac{\rho A l}{420} \begin{bmatrix} 156 & 22l & 54 & -13L \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \tag{3.29}$$

$$K = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6L \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \tag{3.30}$$

$$C = a_M M + a_K K \tag{3.31}$$

Eq.3.26 is the lagrangian formulation of the model, and the most important hypothesis for its validity is that the beam is subjected to small elastic displacements. This assumption can be considered acceptable in the present case since the control problem has the objective of small precision adjustments.
The variables of the equations are: x_R and α_R are deflection and bending angle at the root and x_T and α_T are the analogous quantities for the tip of the beam. M is the mass matrix, K is the stiffness matrix, C is the damping matrix. While mass and stiffness matrices can be easily computed from the geometry and material properties, the damping matrix introduces complex issue because it is affected by many phenomena and can be modelled in many forms that may vary greatly between each other. Damping can be considered as purely viscous, structural or internal, that is strictly dependent on the microscopic properties of the material [49]. Damping is also influenced by manufacturing processes, interactions between parts of the structure, possible presence of fluids, etc.
In the equations above, the matrix C is expressed in terms of Rayleigh damping model, that defines the damping matrix as the linear combination of the mass and stiffness matrices, and it's a representation of the viscous type of damping. Most numerical methods use the viscous damping model because it's computationally convenient and in particular the Rayleigh representation has the big

advantage of being a linear representation, therefore it is suitable for modal analysis and modal uncoupling of the system [50].
The coefficients a_M and a_K can be computed by setting the value of the desired damping ratio ζ and a range of workig frequencies where the modes are relevant in working condition, expressed as $\hat{\omega} \div R\hat{\omega}$. Then the coefficients are determined as follows.

$$\Delta = \frac{1 + R - 2\sqrt{R}}{1 + R + 2\sqrt{R}} \tag{3.32}$$

$$a_M = 2\zeta\hat{\omega}\frac{2R}{1 + R + 2\sqrt{R}} \tag{3.33}$$

$$a_K = 2\zeta\frac{1}{\hat{\omega}}\frac{2}{1 + R + 2\sqrt{R}} \tag{3.34}$$

It is possible to consider a further reduced d.o.f. model if the boundary conditions are imposed, in this case the fact that the beam is clamped at the root fixes the root variables x_R and α_R to zero. Such reduced model is represented in equation 3.35.

$$M_{red}\ddot{x} + C_{Red}\dot{x} + K_{red}x = f \tag{3.35}$$

$$x = \begin{bmatrix} x_T \\ \alpha_T \end{bmatrix} \tag{3.36}$$

$$M_{red} = \frac{\rho A l}{420}\begin{bmatrix} 156 & -22l \\ -22l & 4l^2 \end{bmatrix} \tag{3.37}$$

$$K_{red} = \frac{EI}{l^3}\begin{bmatrix} 12 & -6l \\ -6l & 4l^2 \end{bmatrix} \tag{3.38}$$

$$C_{red} = a_M M_{red} + a_K K_{red} \tag{3.39}$$

To obtain a more efficient model of the telescopic arm it's possible to model the section of the beam as a variable radius anular section, the parameter r decreasing with the increasing length of the manipulator. The dynamic model of the manipulator can be considered the one of many beam elements as in Fig:3.5.
Considering the example with two sliding elements, we can compute a new dynamic equation with a total of six degrees of freedom as follows in Eq. 3.40. Note that we consider each element of the same length $l_1 = l_2 = l$.

$$M\ddot{x} + C\dot{x} + Kx = f \tag{3.40}$$

$$x = \begin{bmatrix} x_1 \\ \alpha_1 \\ x_2 \\ \alpha_2 \\ x_3 \\ \alpha_3 \end{bmatrix} \tag{3.41}$$

$$f = \begin{bmatrix} f_1 \\ m_1 \\ f_2 \\ m_2 \\ f_3 \\ m_3 \end{bmatrix} \tag{3.42}$$

$$M = \frac{\rho l}{420} \begin{bmatrix} 156A_1 & 22lA_1 & 54A_1 & -13lA_1 & 0 & 0 \\ 22lA_1 & 4l^2A_1 & 13lA_1 & -3l^2A_1 & 0 & 0 \\ 54A_1 & 13lA_1 & 156A_1 + 156A_2 & -22lA_1 + 22lA_2 & 54A_2 & -13lA_2 \\ -13lA_1 & -3l^2A_1 & -22lA_1 + 22lA_2 & 4l^2A_1 + 4l^2A_2 & 13lA_2 & -3l^2A_2 \\ 0 & 0 & 54A_2 & 13lA_2 & 156A_2 & -22lA_2 \\ 0 & 0 & -13lA_2 & -3l^2A_2 & -22lA_2 & 4l^2A_2 \end{bmatrix} \tag{3.43}$$

$$K = \frac{E}{l^3} \begin{bmatrix} 12I_1 & 6lI_1 & -12I_1 & 6lI_1 & 0 & 0 \\ 6lI_1 & 4l^2I_1 & -6lI_1 & 2l^2I_1 & 0 & 0 \\ -12I_1 & -6lI_1 & 12I_1 + 12I_2 & -6lI_1 + 6lI_2 & -12I_2 & 6lI_2 \\ 6lI_1 & 2l^2I_1 & -6lI_1 + 6lI_2 & 4l^2I_1 + 4l^2I_2 & -6lI_2 & 2l^2I_2 \\ 0 & 0 & -12I_2 & -6lI_2 & 12I_2 & -6lI_2 \\ 0 & 0 & 6lI_2 & 2l^2I_2 & -6lI_2 & 4l^2I_2 \end{bmatrix} \tag{3.44}$$

where A_1 and A_2 are the different areas of the cross sections of the two modules.
This system can be reduced as well, by imposing the boundary condition that is x_1, $\alpha_1 = 0$ (beam clamped at end 1 in Fig.3.5), and obtaining the system in Eq.3.45. The applied force is only a bending moment on the tip, represented by point 3 in the figure, therefore $f_2 = f_3 = m_2 = 0$, while m_3 is the control input determined by the torque applied by the cable. This torque can assume positive or negative value, depending on which wire is being pulled, so the actuation system must be provided of two symmetrical wires able to pull the tip in opposite directions.

$$M_{red}\ddot{x} + C_{red}\dot{x} + K_{red}x = f \tag{3.45}$$

$$x = \begin{bmatrix} x_2 \\ \alpha_2 \\ x_3 \\ \alpha_3 \end{bmatrix} \tag{3.46}$$

$$f = \begin{bmatrix} f_2 \\ m_2 \\ f_3 \\ m_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_3 \end{bmatrix} \tag{3.47}$$

$$M_{red} = \frac{\rho l}{420} \begin{bmatrix} 156A_1 + 156A_2 & -22lA_1 + 22lA_2 & 54A_2 & -13lA_2 \\ 22lA_1 + 22lA_2 & 4l^2A_1 + 4l^2A_2 & 13lA_2 & -3l^2A_2 \\ 54A_2 & 13lA_2 & 156A_2 & -22lA_2 \\ -13lA_2 & -3l^2A_2 & -22lA_2 & 4l^2A_2 \end{bmatrix} \tag{3.48}$$

$$K_{red} = \frac{E}{l^3} \begin{bmatrix} 12I_1 + 12I_2 & -6lI_1 + 6lI_2 & -12I_2 & 6lI_2 \\ -6lI_1 + 6lI_2 & 4l^2I_1 + 4l^2I_2 & -6lI_2 & 2l^2I_2 \\ -12I_2 & -6lI_2 & 12I_2 & -6lI_2 \\ 6lI_2 & 2l^2I_2 & -6lI_2 & 4l^2I_2 \end{bmatrix} \tag{3.49}$$

$$C_{red} = a_M M_{red} + a_K K_{red} \tag{3.50}$$

In a more realistic model of the telescopic arm, it may have five or more modules, as in Fig.3.5. In this case the mass, stiffness and damping matrices are assembled as in the previous simplified example of two modules and the final model will have $2(n+1)$ d.o.f., where n is the number of modules. Among these $2(n+1)$ variables, two are fixed because they are relative to point 1, that is the clamped root of the arm. These two variables can be removed from the total number of d.o.f., obtaining $2n$ variables of the reduced dynamic equation.
The reduced matrices of the final 5-element telescopic arm will have dimension 10×10.

3.2 Solution 1

In this solution, the screw conveyor hosts the shooting system inside it. The tube of the screw is equipped with one or more holes that allow the passage of the wire to which the end effector is connected. The idea is to shoot the gripper at the bottle and then retract it, positioning the bottle inside the spot on the screw conveyor. The final position of the bottle will be vertical, it will continue to move at the same speed on the second conveyor belt.

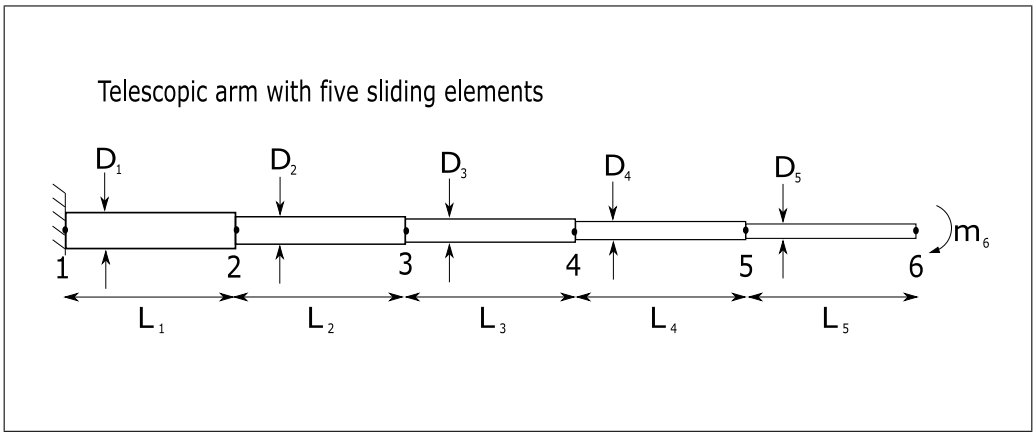


Figure 3.5: Modelling of the telescopic arm with five modules represented as five beam elements of same length

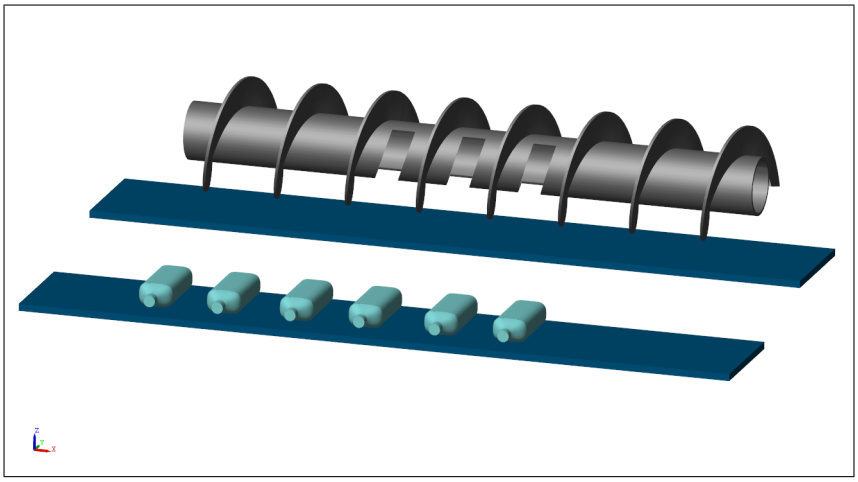


Figure 3.6: Solution 1. Screw conveyor with shooting system positioned inside the screw.

The system is as compact as possible and the cost is minimized, but the main drawback is that the procedure of shooting, grasping and pulling back the target must be performed in a very short time. Since every spot in the screw must be filled with one bottle, this time is determined by the screw rotating speed and the dimension of the gap. For example, if the gap has an elongated shape that covers half of the perimeter of the internal cylinder (Fig.3.6), the available time to complete the task is the time it takes the screw to do half a revolution. This observation suggests that such solution is more suitable for the applications that require speeds close to the best case of 0.4 m/s.

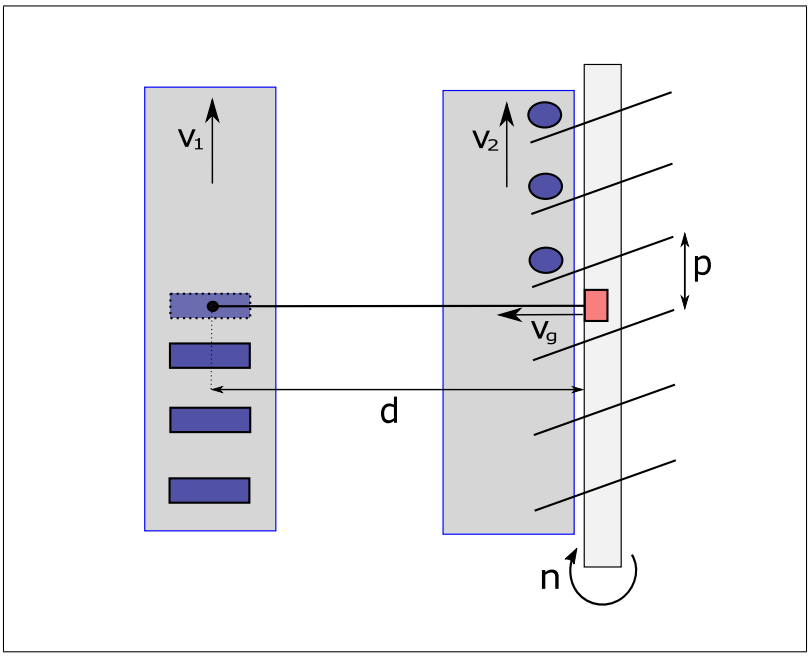


Figure 3.7: Solution 1. 2D scheme in the X-Y plane

Referring to Fig.3.7, we consider as h the fraction of the screw tube circumference that is taken by the hole, and as v_g the average gripper speed norm during the whole task, the operation time is limited as follows:

$$t_o = 2\frac{d}{v_g} \leq \frac{h}{n} \tag{3.51}$$

$$v_{gmin} = \frac{2dn}{h} \tag{3.52}$$

3.3 Solution 2

In this system, the shooting equipment is placed inside a separate structure positioned above the screw conveyor. The holes from which the end effector is shot lay on a sloped plane, from where the shot must aim at the bottle. The bottle is pulled back above the screw conveyor and then dropped vertically in its spots.
In this case the problem of speed is not so pressing as in the previous solution thanks to the possibility to grab many bottles at the same time with many shooting manipulators, possibly covering the whole length of the screw conveyor (Fig.3.8). If the system grabs N bottles at the same time and places them in N

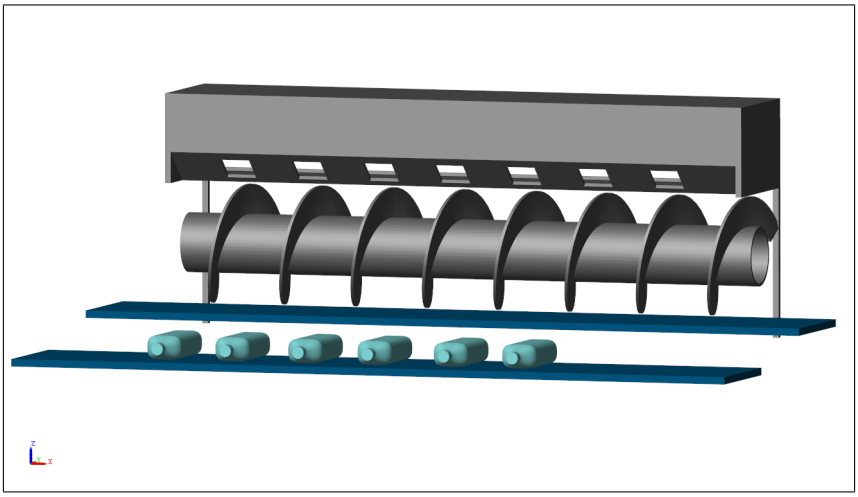


Figure 3.8: Solution 2. Multiple shooting systems fixed above the screw

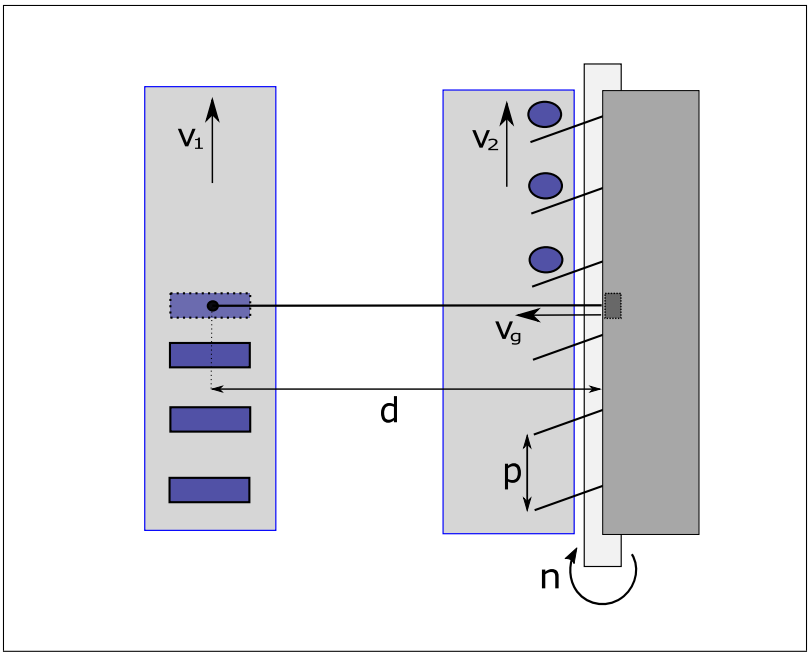


Figure 3.9: Solution 2. 2D scheme in the X-Y plane

spots of the screw conveyor, the time available to reposition the grippers, perform the next shot and pull is the time employed by the second conveyor/screw to drain away the first N bottles and leave N spots free for the next bottles, i. e. N revolutions of the screw. The available time to perform the whole process is multiplied by a factor N with respect to the previous solution.

Considering Fig.3.9 and defining as N the number of manipulators acting at the same time, we have:

$$t_o = 2\frac{d}{v_g} \leq \frac{N}{n} \tag{3.53}$$

$$v_{gmin} = \frac{2dn}{N} \tag{3.54}$$

Note that the minimum average velocity of the task is considerably lower than in the previous solution since it's always true that $N \geq 1$.

3.4 Solution 3

The idea at the base of the third solution is to let the shooting devices slide on a guide above the conveyors, moving from one conveyor to the other and shooting and dropping the bottles. The main drawback of this configuration is the necessity of actuation for the sliding motion of the system, which can be seen as a prismatic joint. This problem leads to strict constraints on the maximum speed of the process and is more complex than the previous ones, but has the advantage seen also for the second solution, that is the possibility to transfer many bottles at the same time.
The operation time can be determined similarly to the previous case and by looking at the scheme in Fig.**??**c, consider as t_g the time that in this case is reserved to the drop of the gripper and grabbing the object and release of the object, during which the sliding part is still above the conveyor. We want to compute how fast the sliding joint must behave in order to transfer the bottles flow rate (minimum v_t).

$$t_o = 2\frac{d}{v_g} \leq \frac{N}{n} \tag{3.55}$$

$$v_{gmin} = \frac{2dn}{N} \tag{3.56}$$

$$v_{tmin} = \frac{2d}{t_o - t_g} \tag{3.57}$$

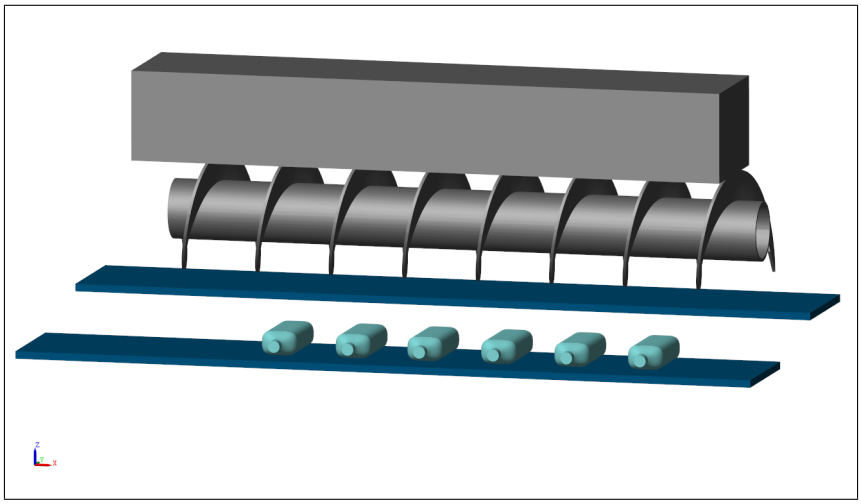


Figure 3.10: Solution 3. Multiple shooting systems placed in a moving body above the screw

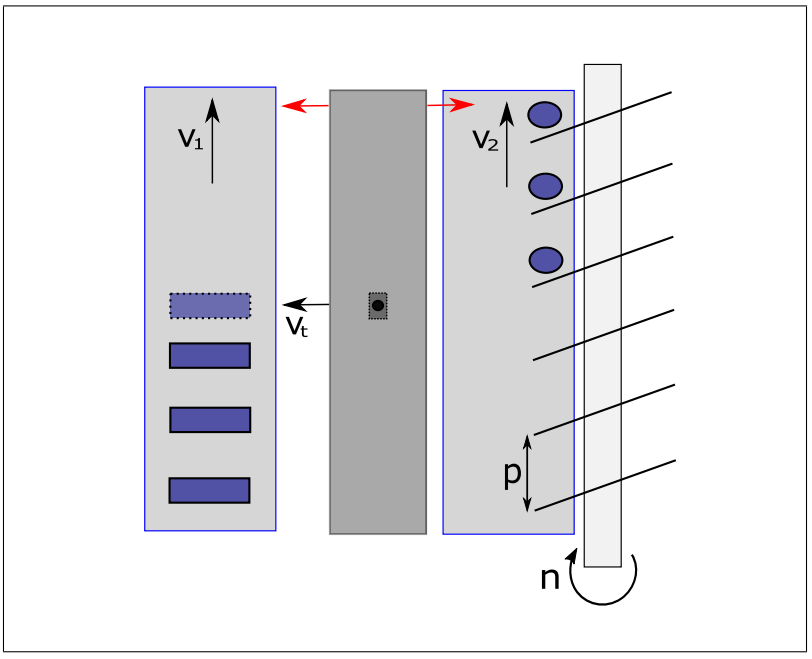


Figure 3.11: Solution 3. 2D scheme in the X-Y plane

Chapter 4

Simulation and discussion

4.1 Robot trajectory simulation

A model of the robot was built with the Simscape Multibody tool of Simulink, represented in Fig.2.1, in order to have an intuitive graphical representation of the execution of the task. The Simscape Multibody "Revolute Joint" blocks were exploited not only to connect and actuate the links of the robot, but also for sensing: the input simulation parameter was the trajectory and the sensed output parameters were angular speed, acceleration and actuation torque of the joint. Each joint is connected to the consecutive link, which is represented by a "Solid" block that makes use of a .step file containing the CAD drawing of each specific link. For torque sensing, the obtained result is indicative and approximated by an estimated distribution of the mass of the robot on each link, starting from the data available about its total weight and knowing that the configuration of the robot is vertical, i.e. the gravity vector pointing towards the $-z$ axis.
The linear constant velocity sections have been fixed to a length of $0.4m$ and divided in 40 samples for the computation of the inverse kinematics, while only four samples have been considered in the elliptic path parts, in order to leave to the interpolation method the task to output the trade-off between the fastest and smoothest trajectory.

The simulations have been carried on with a trial-and-error approach aiming to find the minimum time that can be imposed to complete each section of the path without exceeding the maximum joint speed limits. The speed limits of the robot can be found on the datasheet [33], Fig.4.3 and have been plotted in Fig. **??** together.
In the final simulation, the linear parts of the path have a fixed execution time imposed by the task, that is the time needed to complete the considered section

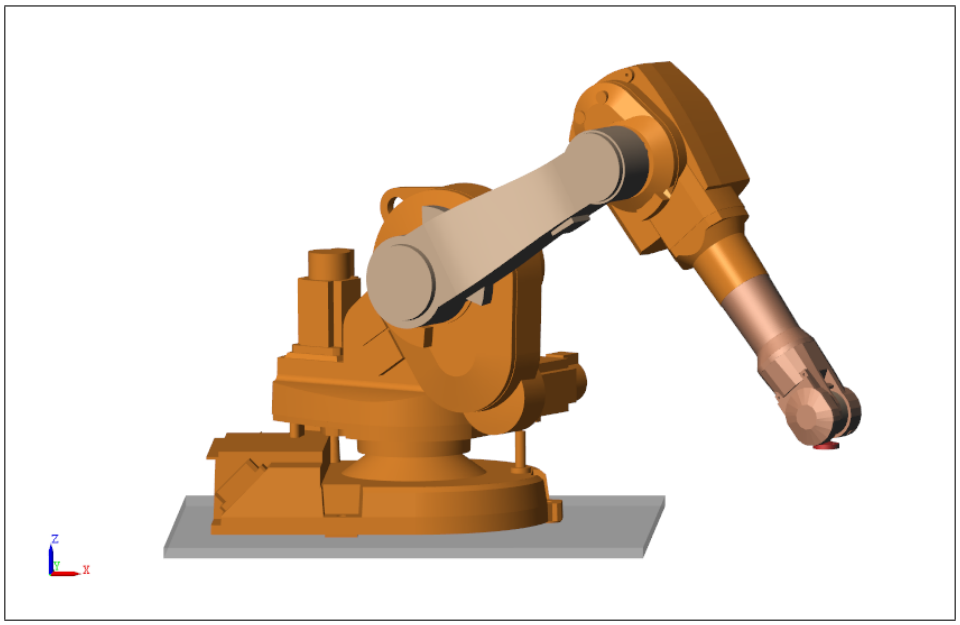


Figure 4.1: Representation of the robot in the Simscape Multibody environment

with constant velocity. Here a linear path of 0.4 m is considered in both the conveyor belts, therefore since the target velocity is considered fixed to 1 m/s, the execution time of the linear sections is set to 00.4 s and cannot be modified. The sections that allow a modification on the execution time are the elliptic path; on the results it can be seen that the maximum velocity in the first elliptic phase is close to but below the limit, this result has been obtained by setting to 0.9 s the execution time for each elliptic section.

As stated above, between each path section a transition section has been left (0.1 m in the $x - axis$ and 0.1 m in the $z - axis$), whose execution time has been fixed to 0.1 s. The total simulation time of the final trajectory for the complete cycle of four phases is 3.08 s. This is the minimum time that allows to keep the velocity of every joint of the robot under the maximum allowed speed.

4.2 Manupulators simulations

4.2.1 Shooting manipulator control

The shooting system described and modelled in the previous sections is clearly not controllable after the instant of the shot. Subsequently, the control scheme associated to the task is an open loop scheme where the selection of a rendezvous

point and the prediction of the interception between the target object and the gripper is crucial [46]. The technique is identified in classic industrial robotics as APPE, Active Prediction Planning Execution, but it can be exploited in the present soft robotics situation with the limitation of the open-loop control scheme that must be used here, and furthermore assuming the total impossibility of control after the instant of the shot. In particular, the interesting part of the method for the present application is the prediction part, that must be followed by an estimation of the trajectory of the gripper after the shot.
While in a closed loop controlled robot, such as a general rigid robotic arm, the trajectory can be planned and executed reliably, in the shooting system control this phase is not present and must be compensated with a correct estimation of the shot gripper trajectory. This issue clearly introduces uncertainty in the overall control of the task.
Moreover, the APPE approach uses the real-time updates on the motion of the target object to replan the optimal trajectory of the TCP and feed it to the control of the robot. In the present application this approach is not feasible but in the phase before the shot, which is a grat limitation as well. Predicting the interception point means having reliable information on the trajectory of the object, which is affected by uncertainty as well. A method can be a tuned Kalman filter, as shown in [47].

After the the shooting instant has been computed, the two control inputs available in the considered model are to be chosen: the impulse of the shot and the shooting angle. The technique presented here is an iterative algorithm developed in matlab that runs the simulation of the plant a fixed number of times to find the best combination of impulse and angle.
The algorithm is divided into three phases: firstly the best shooting angle is estimated considering the target point position as the angle between the horizontal axis and a value corresponding to the z coordinate of the target corrected of a certain amount of height to allow the shooting manipulator to fully elongate and drop vertically on the target; then the simulation of the plant is run multiple times, each time changing the value of the impulse of the shot, which will vary in a limited range of allowed values.
In the same for loop, the output of the simulations is compared to the target coordinates in each time instant to check if that specific tragectory meets the target or not. This comparison checks if the simulated trajectory passes through a spheric portion of the space defined by a tolerance set by the user, and whose center is the target point.
If the interception happens, the value of the impulse used for that simulation in particular is stored in a vector containing all the feasible impulses.
In the third phase, the final control variables values have to be chosen and the

technique may vary depending on the goal of the optimization problem. In this case, the minimum energy consumption objective has been considered, therefore the algorithm chooses the input that is the minimum impulse force among the ones that allow the interception of the object.
The final control inputs can be fed to the real plant.

4.2.2 Shooting manipulator simulation

The simulation of the plant has been carried on by considering the mathematical model depicted above and reducing the state equations to two main scenarios, depending on the value assumed by the parameter Δl.
The state vector contains the three physical variables x, y, θ and their first derivatives.

$$\Delta l := (x - L) - r\theta \tag{4.1}$$

1. $\Delta l > 0$
   In this case there is a non-null tension T applied on the string:

   $$T = k\Delta l \tag{4.2}$$

   The state equation in matrix form is the following:

   $$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k}{m} & 0 & 0 & 0 & \frac{rk}{m} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{rk}{I} & 0 & 0 & 0 & -\frac{r^2k}{I} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{kL}{m} \\ 0 \\ -g \\ 0 \\ -\frac{rkL}{I} \end{bmatrix} \tag{4.3}$$

2. $\Delta l \leq 0$
   In this case the string is loose and the tension applied on it is null:

   $$T = 0 \tag{4.4}$$

   The state equation in matrix form is the following:

   $$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g \\ 0 \\ 0 \end{bmatrix} \tag{4.5}$$

The result of the simulation led to the behavior of Fig. 4.7 in the X-Y plane where it is possible ro recognize the phases described in the study [27], that are: ballistic motion, contribution of the elastic effect when the string reaches its maximum length, resting in place of the gripper and then dropping down.

4.2.3 Telescopic arm control

The control system of the telescopic manipulator has been based on its dynamic model previously described, but the allowed controllable deflection directions are now two: x and z axes. To implement in simulink such model, the dynamic equation has been transformed into state-space representation in Eq.4.6. The state vector z contains displacements x and angles α, that are vectors with generic dimension depending on the number of beam elements in which the model is divided.

$$\dot{z} = Az + Bu \tag{4.6}$$

$$z = \begin{bmatrix} \dot{x} \\ \dot{\alpha} \\ x \\ \alpha \end{bmatrix} \tag{4.7}$$

$$u = \begin{bmatrix} 0 \\ m_T \end{bmatrix} \tag{4.8}$$

$$A = \begin{bmatrix} -M^{-1}C & -M^{-1}K \\ I & 0 \end{bmatrix} \tag{4.9}$$

$$B = \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix} \tag{4.10}$$

$$\tag{4.11}$$

There may be many different control objectives to which one must aim to choose the most suitable control scheme, these are resumed in [36] as:

- End effector regulation: the arm must reach a desired point with the minimum oscillations of the tip;
- End effector to rest motion: the tip must reach a desired point in a defined time and then rest, i.e. there must be a complete elimination of residual osillations;
- Joint trajectory tracking: the joint angles must follow the desired angular trajectories;

- End effector trajecory trajectory tracking: the cartesian position of the end effector must follow the desired trajectory, this is the most problematic condition since the dynamics of the system is non-minimum phase;

The control objective presented here is to follow a planned trajectory in the space. It is important to point out that the developed control scheme is a superficial level control that transcends the choice of specific actuators, taking as control input the torque applied on the tip and ignoring the drives of the actual motors that would be used to apply such torque.
The control scheme is composed of a logic block subsystem whose role is to handle the switching of the states of the system: the arm elongating, the arm at its maximum length and then the retracting phase.
After this block the trajectory is generated for each phase to start the deflection for the desired position during the varying length phase, while the arm is extending; the aim of this block is to generate a reference for the tip of the arm, that we consider the tool center point.
The actual control consists in a PID tuned to reduce the vibrations of the system at high speed in the x-axis The control input to the plant is the moment on the tip m_T which is proportional to the pulling force of the wires and the arm of this force.

The implemented PID controller features also a filter and its law in the Laplace transform domain is represented in Eq.4.12, where P is the proportional coefficient, D is the derivative coefficient, I is the integral coefficient and N is the filter coefficient.

$$P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \tag{4.12}$$

4.2.4 Telescopic arm simulation

The dynamic model presented above has been used to build two simulations, in the first one only one cartesian variable is controlled, in the second one a second force has been added to control both x and z axes. Firstly the system has been simulated for an arm with just two modules with anular cross section of diameter $4cm$ and $3.4cm$ respectively. The Young's modulus has been set to $E = 200 \cdot 10^9 Pa$ and the material density to $\rho = 8 \cdot 10^3 \frac{kg}{m^3}$.
The basic assumptions are that the arm extends at constant velocity $v = 2\frac{m}{s}$ and the elements of the arm are extending all simultaneously. The arm geometric parameters are the total length $l = 1.2m$, of which $0.2m$ is the rest length of the retracted arm.The thickness of the two modules is $3mm$ and $2mm$ respectively. The parameters for the computation of the damping coefficients are: damping

ratio $\zeta = 0.7, \hat{\omega} = 10^3 \frac{rad}{sec}$ and $R = 4$, the damping is limited between $\zeta - \Delta(R, \hat{\omega})$ and $\zeta + \Delta(R, \hat{\omega})$, the Rayleigh coefficients obtained are $a_M = 124.44$ and $a_K = 3.11 \cdot 10^{-5}$.
The bode diagram of the plant and the open loop is represented in Fig.4.10.

The parameters for the tuned controller are the following.

$$P = 12.6 \cdot 10^4 \tag{4.13}$$
$$I = 60.3 \cdot 10^6 \tag{4.14}$$
$$D = 45.2 \tag{4.15}$$
$$N = 10.6 \cdot 10^3 \tag{4.16}$$

The controlled system has been tested in a simulation of $T = 1sec$ for different kinds of input: step, sinusoidal and a composite trajectory signal computed to simulate the possible trajectory of the pick-and-place task.
The simulated system presents a marked oscillatory behaviour in the step response, but has a better response in the case of piecewise linear input and sinusoidal waveform input. Some of the simulation outputs can be seen in Fig.4.11. The vibrations take place in particular at time $\hat{t} = 0.5sec$ where the area of the telescopic arm cross section changes abruptly because of the sliding of the second module.

The simulation of the task trajectory was pursued considering as controlled variables the x and z-axes moments. The reference and output of the simulation can be seen in Fig: 4.12. This trajectory was generated through the cubic spline interpolation method in an external matlab script and it is function of the position of the bottle sensed by a vision system. The bottle is moving at $v = 1\frac{m}{s}$ in the x direction, the TCP of the manipulator follows the direction of the target linearly for a few centimeters during which the gripping takes place. During this linear phase the z variable remains constant at the target coordinate, then the grabbed object is lift and the x variable returns to zero to face the retracting phase of the manipulator.

It is interesting to observe how the results of the simulation change after modifying the geometric parameters of the plant. For example, a new simulation shows how reducing the diameters and the thickness of the beam tubes, the system shows a much better step response. Such results (Fig.4.13) have been obtained by leaving the same PID controller gains P, D, I and only changing the above geometric quantities in the plant equation in the single d.o.f. control system. The new diameters of the two modules are $d_1 = 3cm$ and $d_2 = 2cm$, and the respective thickness is $1.5mm$ and $1mm$.
This analysis has not the goal of finding the best geometric construction of the telescopic arm, which is left to future works, but just underlining how important

the choice of such parameters is for the controllability and control performance of the whole system.

A third simulation on the same system shows how the new geometric parameters react with a different controller with respect to the previous PID, tuned specifically on the new plant model. The new controller is a simple integrator with integral gain $I = 12.9 \cdot 10^3$ and a filter coefficient $N = 100$. This is a much slower controller, with a response time $t_r = 0.2188s$ and tuned with the Matlab PID Tuner App to give a robust transient response. This aspect is crucial from the actuators point of view, since actuators cannot not provide sudden variations in the input and it is well known that introducing delays in the control of dynamic systems can easily lead to instablility [44].
As consequence, a slower controller means that the change rate of the control input is more feasible in a real application.
It can be observed in Fig.4.15 that the controlled output is not free of oscillations, because of the relatively low response time, but it is bounded and not diverging. This was not true when trying to apply a similar controller, integrator with low response time, to the previous system with larger and thicker beam elements; in fact, the geometrical quantities of the beams affect their mass and stiffness matrices, changing the resonant frequencies of the system and therefore its response to external solicitations.

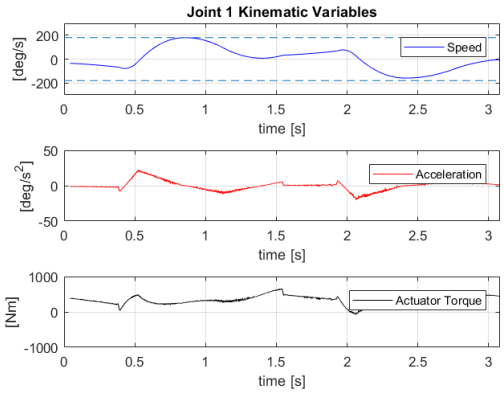
Simulation	States	d_1	d_2	p_1	p_2	Figure
		mm	mm	mm	mm	
1 a	x	40	34	3	2	4.11
1 b	x, z	40	34	3	2	4.12
2 a	x	30	20	1.5	1	4.13
2 b	x, z	30	20	1.5	1	4.14
3 a	x	30	20	1.5	1	4.15
3 b	x,z	30	20	1.5	1	4.17

Table 4.1: Geometric parameters of the telescopic arm in each simulation

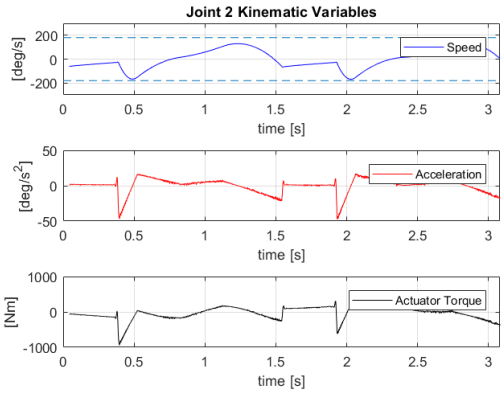
The conclusion to which these last two simulation lead is the necessity to study in depth the mechanical design of the telescopic arm, which has to be pursued in parallel to the dimensioning of the side components of the whole system, such as actuators. The importance of this process lays in the total equilibrium of the controlled system involving many elements and whose main difficulty is the presence of oscillating behavior and dangerous vibrating resonance.

Simulation	States	P	I	D	N	Figure
1 a	x	$12.6 \cdot 10^4$	$60.3 \cdot 10^6$	45.2	$10.6 \cdot 10^3$	4.11
1 b	x, z	$12.6 \cdot 10^4$	$60.3 \cdot 10^6$	45.2	$10.6 \cdot 10^3$	4.12
2 a	x	$12.6 \cdot 10^4$	$60.3 \cdot 10^6$	45.2	$10.6 \cdot 10^3$	4.13
2 b	x, z	$12.6 \cdot 10^4$	$60.3 \cdot 10^6$	45.2	$10.6 \cdot 10^3$	4.14
3 a	x	0	$12.9 \cdot 10^3$	0	100	4.15
3 b	x,z	0	$12.9 \cdot 10^3$	0	100	4.17

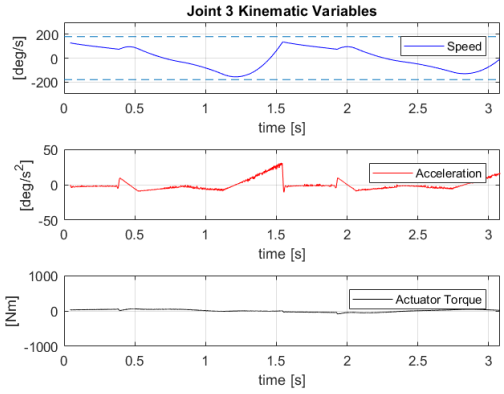
Table 4.2: Gain parameters of the PID controller in each simulation



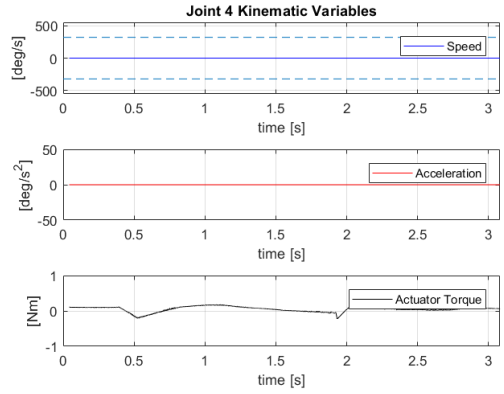
(a) 1-st joint variables



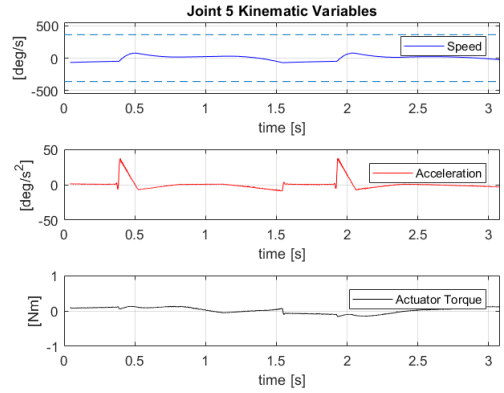
(b) 2-nd joint variables



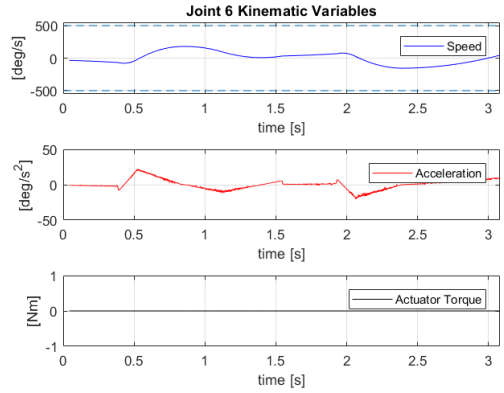
(c) 3-rd joint variables



(d) 4-th joint variables



(e) 5-th joint variables



(f) 6-th joint variables

Figure 4.2: Simulation results: Speed, Acceleration and Torque plot of a complete task execution

Alimentazione trifase

N. asse	IRB 1600-6/1.2 IRB 1600-6/1.45	IRB 1600-10/1.2 IRB 1600-10/1.45	IRB 1660ID-6/1.55 IRB 1660ID-4/1.55
1	150°/s	180°/s	180°/s
2	160°/s	180°/s	180°/s
3	170°/s	185°/s	180°/s
4	320°/s	385°/s	320°/s
5	400°/s	400°/s	360°/s
6	460°/s	460°/s	500°/s

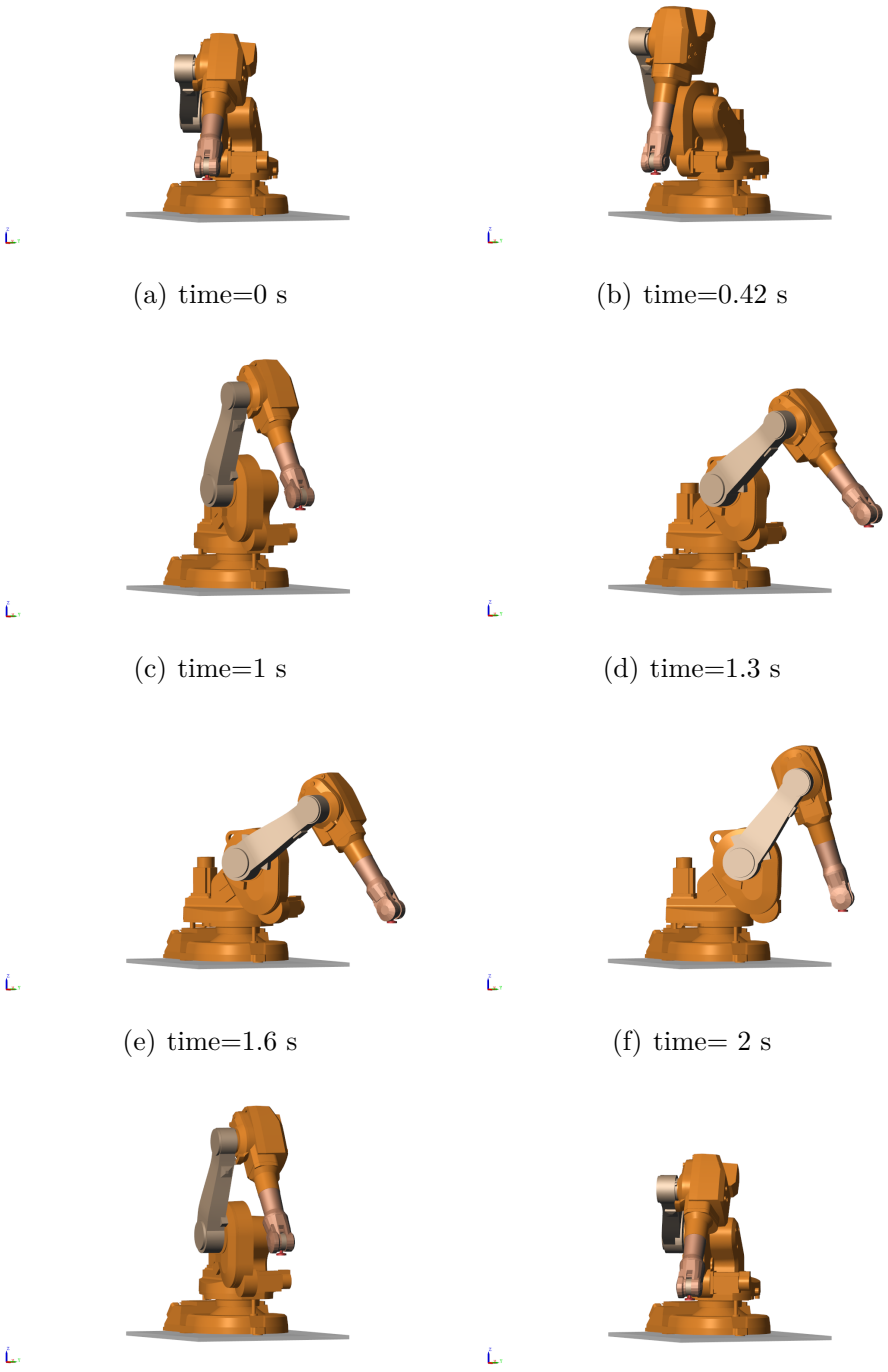
Alimentazione monofase

Quando il robot utilizza un'alimentazione monofase, come nel caso del Compact Controller, le prestazioni riguardanti la velocità massima dell'asse sono ridotte; vedere la tavola sottostante. La velocità massima ridotta può essere incrementata se la tensione minima dell'alimentazione è più elevata del valore d'impostazione predefinito, ovvero di 187 V (220 V x 0,85). Per informazioni dettagliate, vedere "Mains tolerance min" nel Manuale tecnico di riferimento - Parametri di sistema, "Come ottimizzare i parametri del sistema di trasmissione".

Da notare che sull'accelerazione del robot non influisce l'alimentazione monofase. Per questo motivo, può darsi che il tempo di ciclo non vari per niente. Per il test del tempo di ciclo si può utilizzare RobotStudio. Anche RobotStudio consente la modifica di "Mains tolerance min".

N. asse	IRB 1600-6/1.2 IRB 1600-6/1.45	IRB 1600-10/1.2 IRB 1600-10/1.45	IRB 1660ID-6/1.55	IRB 1660ID-4/1.55
1	144°/s	144°/s	142°/s	142°/s
2	130°/s	139°/s	141°/s	141°/s
3	153°/s	163°/s	157°/s	157°/s
4	320°/s	376°/s	320°/s	320°/s
5	364°/s	354°/s	329°/s	329°/s
6	460°/s	460°/s	368°/s	371°/s

Figure 4.3: Speed limits for the robot joints in two different cases: three-phase power and single phase power [33].



(a) time=0 s (b) time=0.42 s

(c) time=1 s (d) time=1.3 s

(e) time=1.6 s (f) time= 2 s

(g) time=2.6 s (h) time=3.08 s

Figure 4.4: Simulation in Simscape Multibody

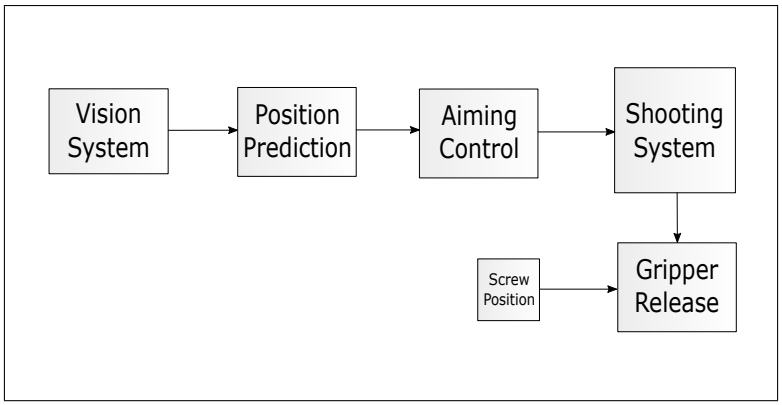


Figure 4.5: Block diagram of the control system

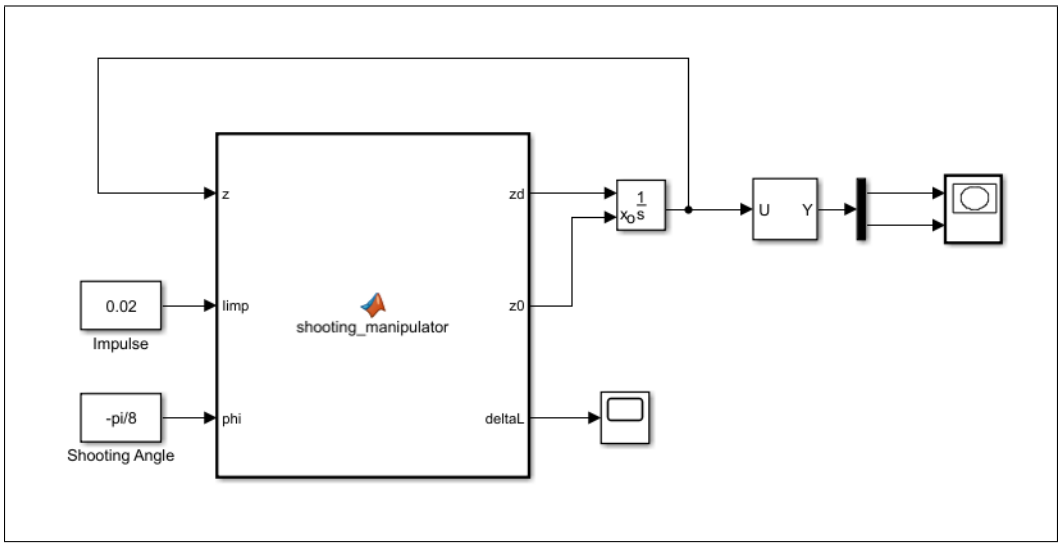


Figure 4.6: Simulation and control of the shooting system

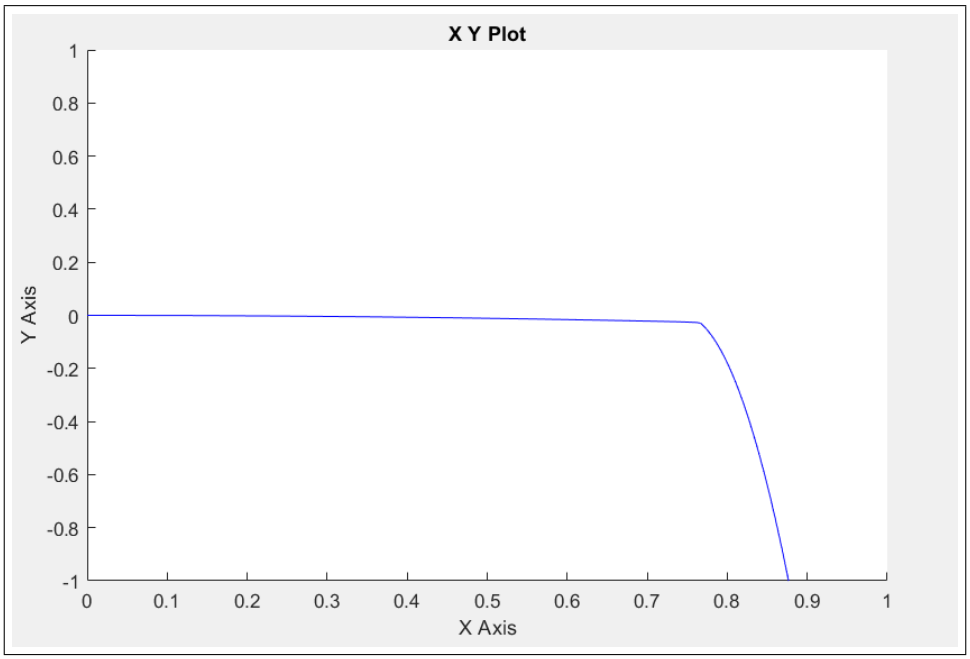


Figure 4.7: X-Y plot of the motion of the gripper after the shot

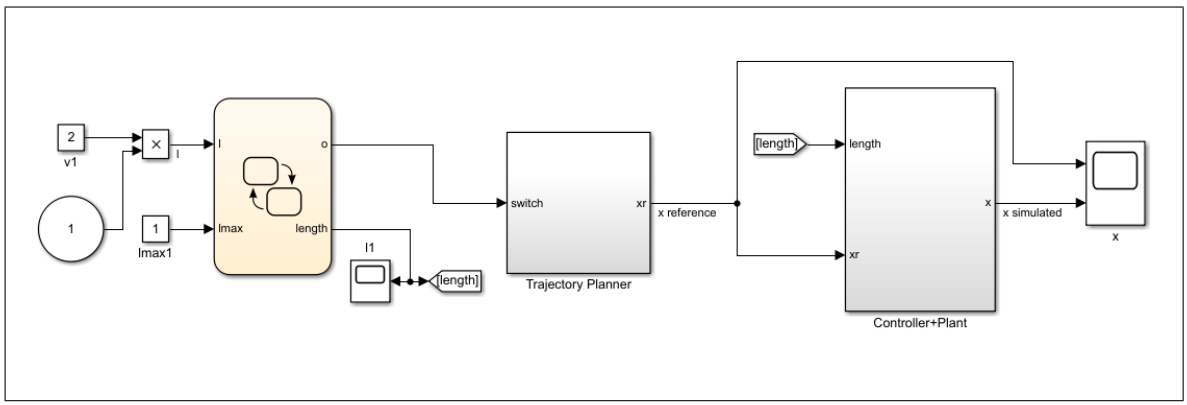


Figure 4.8: Simulation and control of the telescopic arm system with 1 d.o.f.: only the deflection in the x direction is controlled

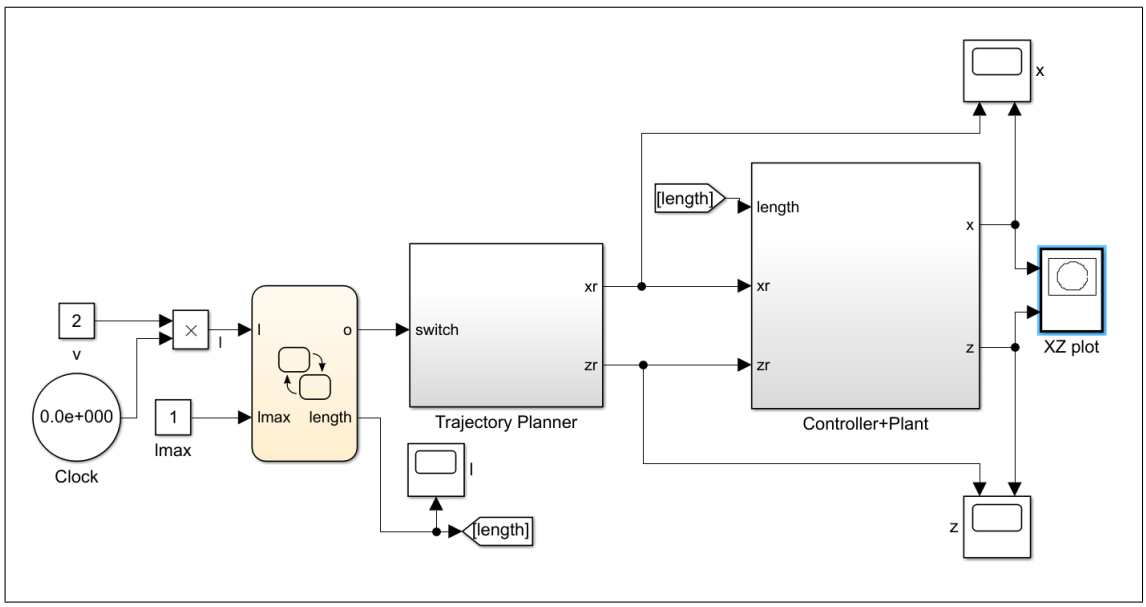
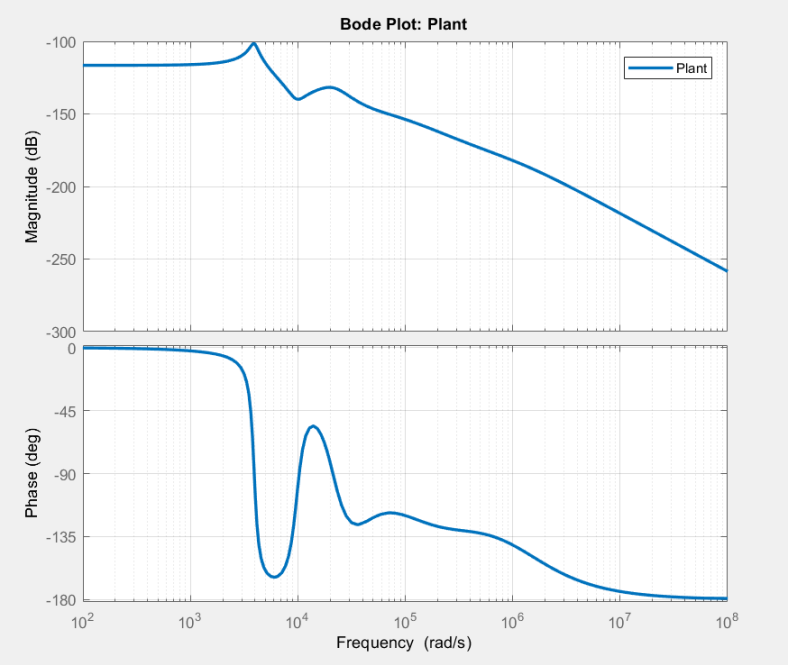
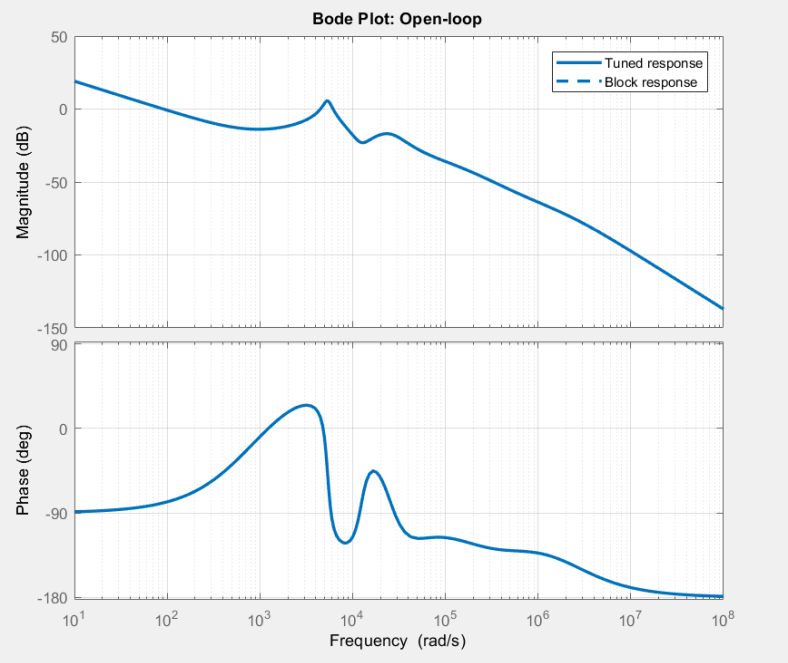


Figure 4.9: Simulation and control of the telescopic arm system with 2 d.o.f.: deflection in the x and z directions is controlled

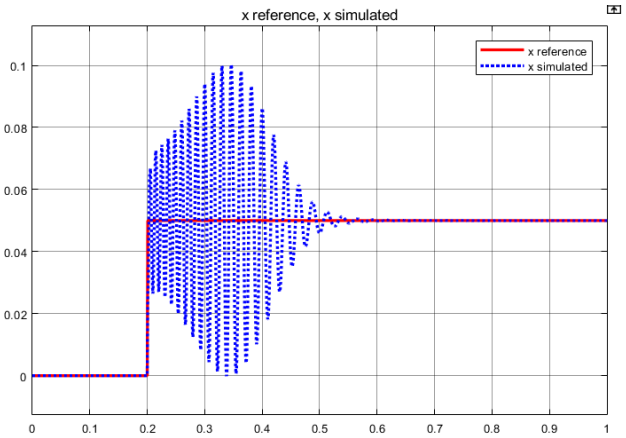


(a) Plant

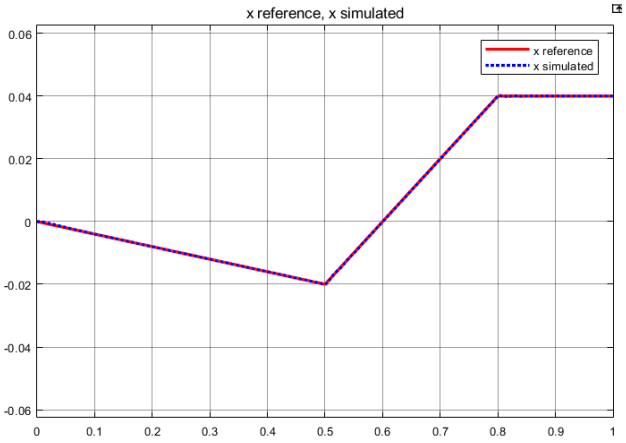


(b) Open-loop

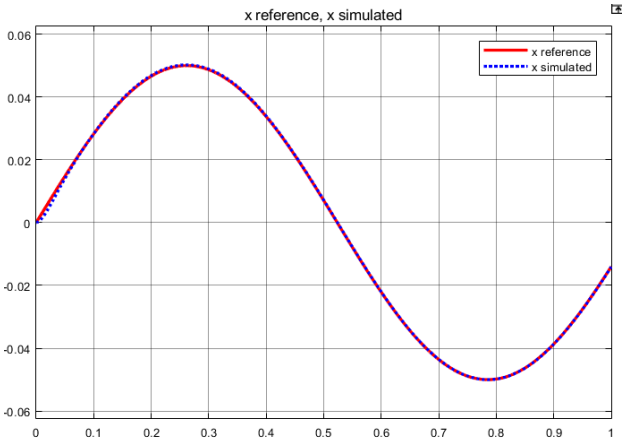
Figure 4.10: Bode plot of the plant alone and the open-loop chain



(a) Step response

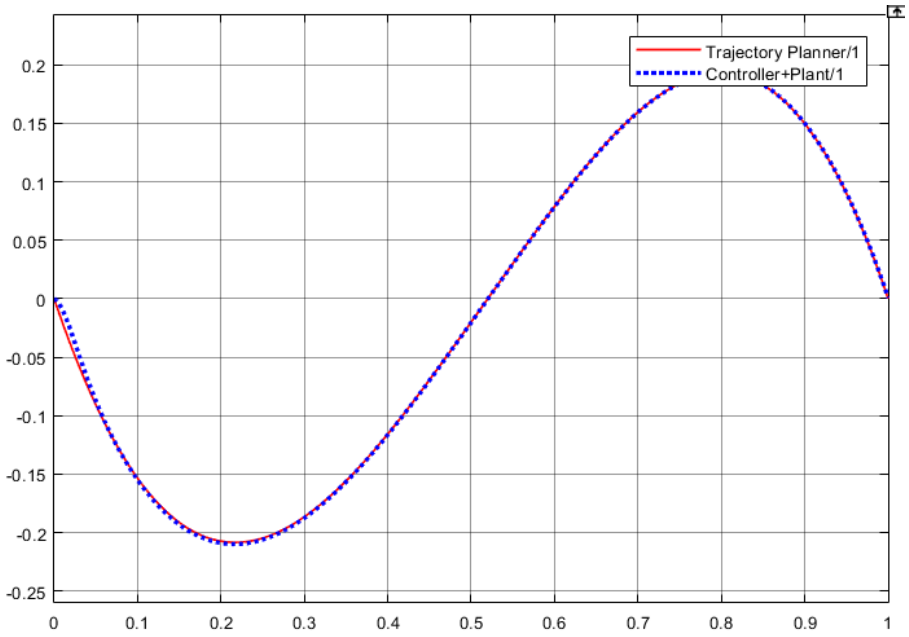


(b) Ramp response

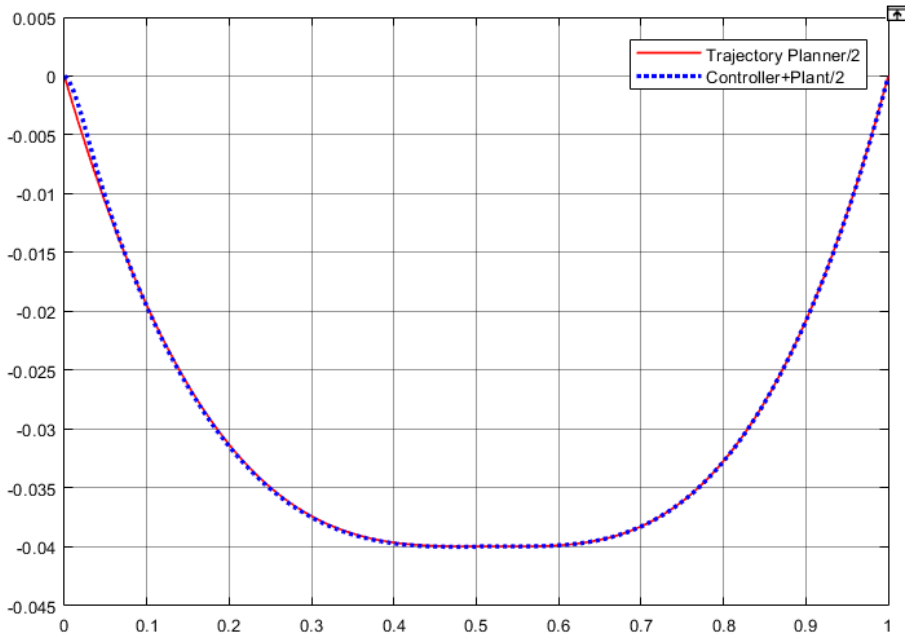


(c) Sinusoidal response

Figure 4.11: Examples of the responses of the controlled system for different reference signals

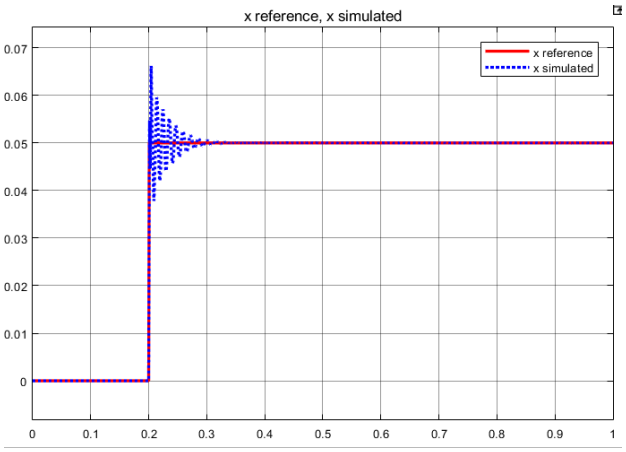


(a) x variable trajectory tracking

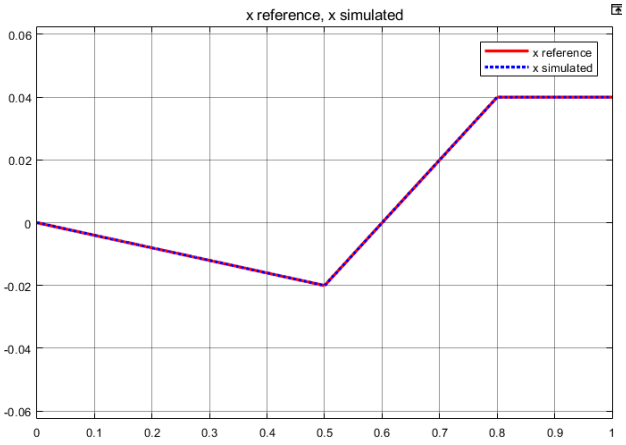


(b) z variable trajectory tracking

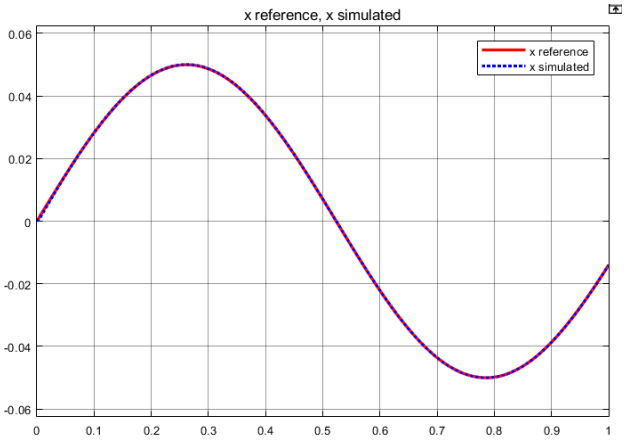
Figure 4.12: Planned trajectory and controlled system response for the pick-and-place task



(a) Step response

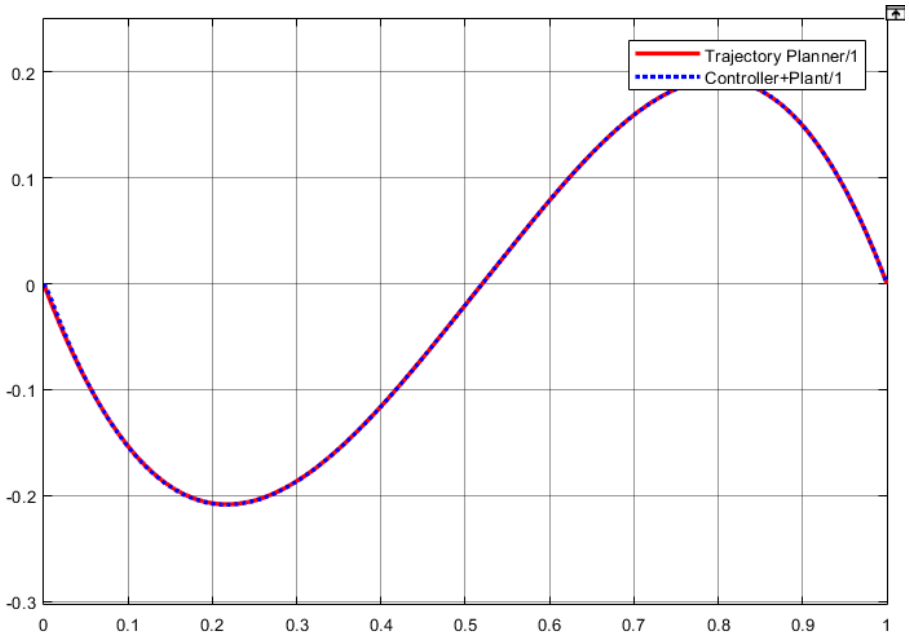


(b) Ramp response

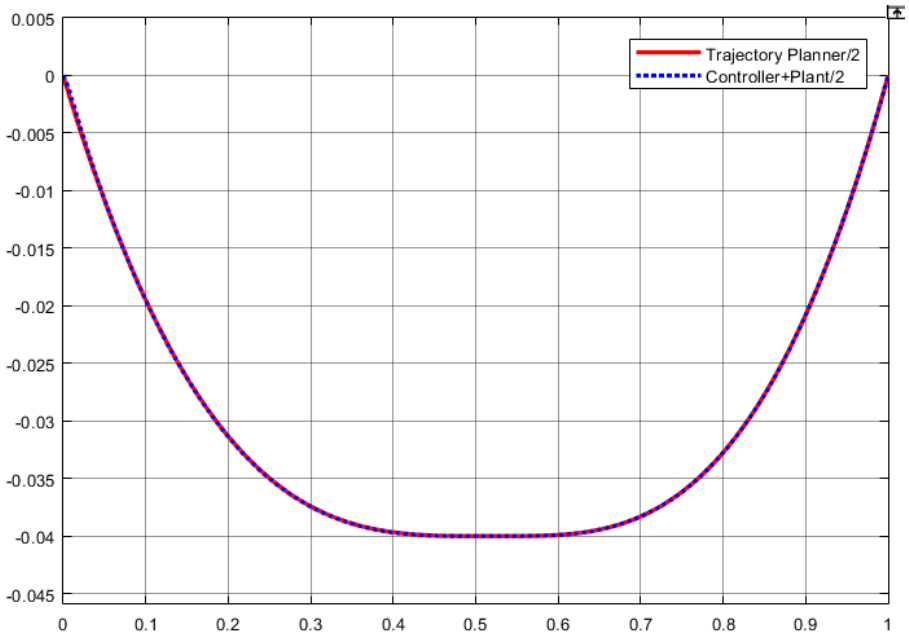


(c) Sinusoidal response

Figure 4.13: Examples of the responses of the controlled system after changing the plant geometric properties

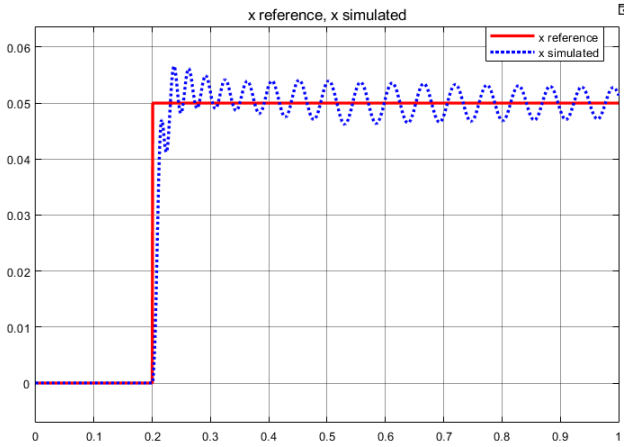


(a) x variable trajectory tracking

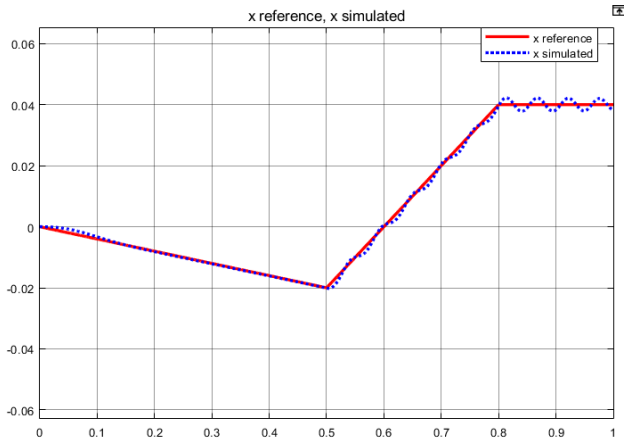


(b) z variable trajectory tracking

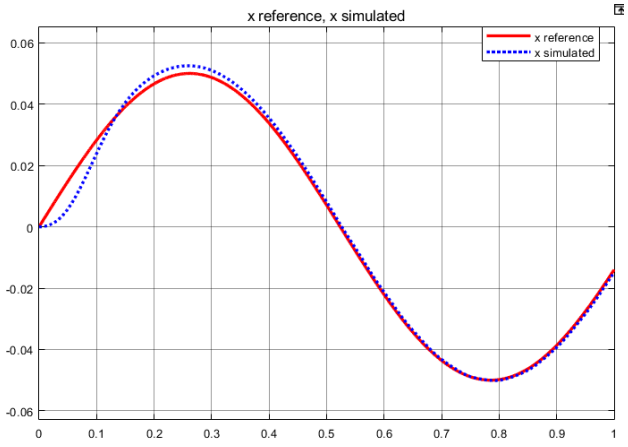
Figure 4.14: New geometry. Planned trajectory and controlled system response for the pick-and-place task.



(a) Step response

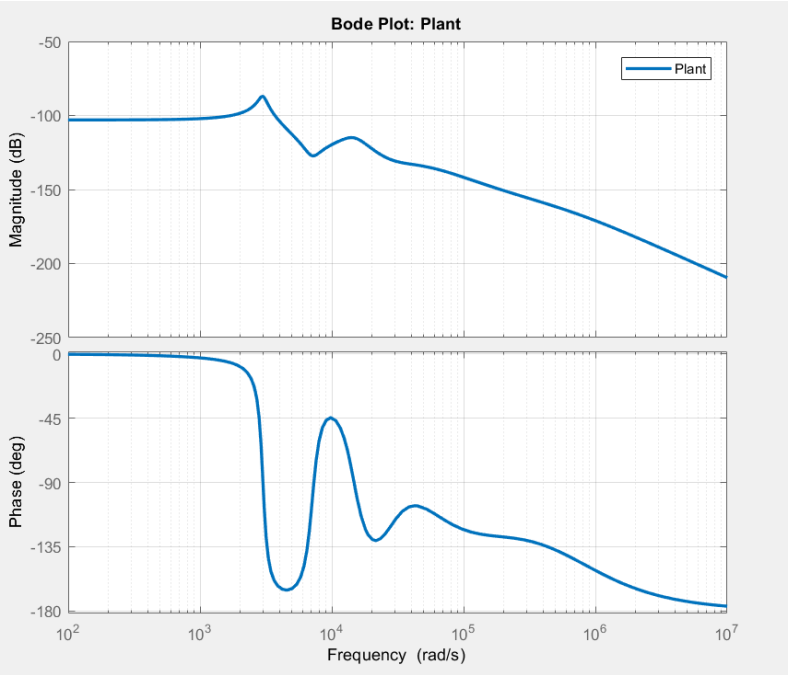


(b) Ramp response

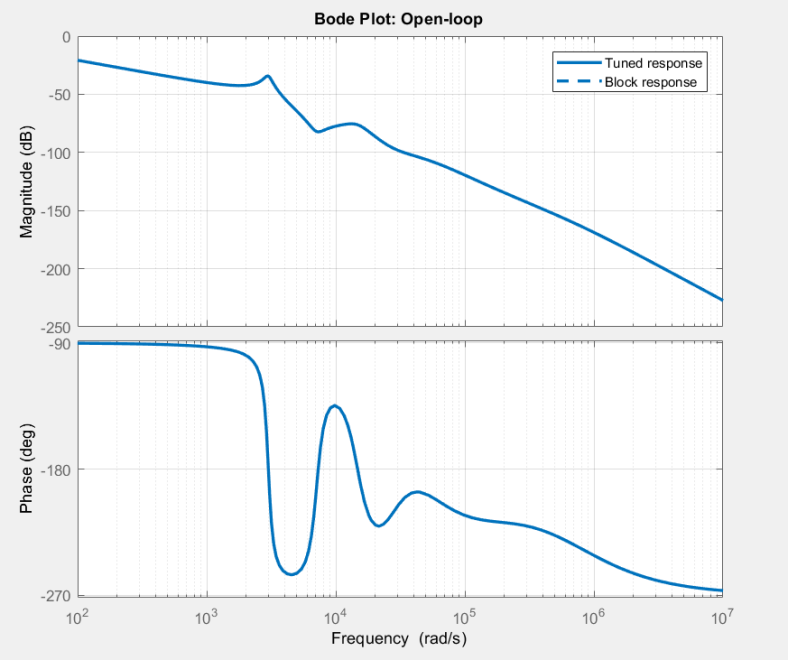


(c) Sinusoidal response

Figure 4.15: Examples of the responses of the controlled system after changing the plant geometric properties and the controller gains.

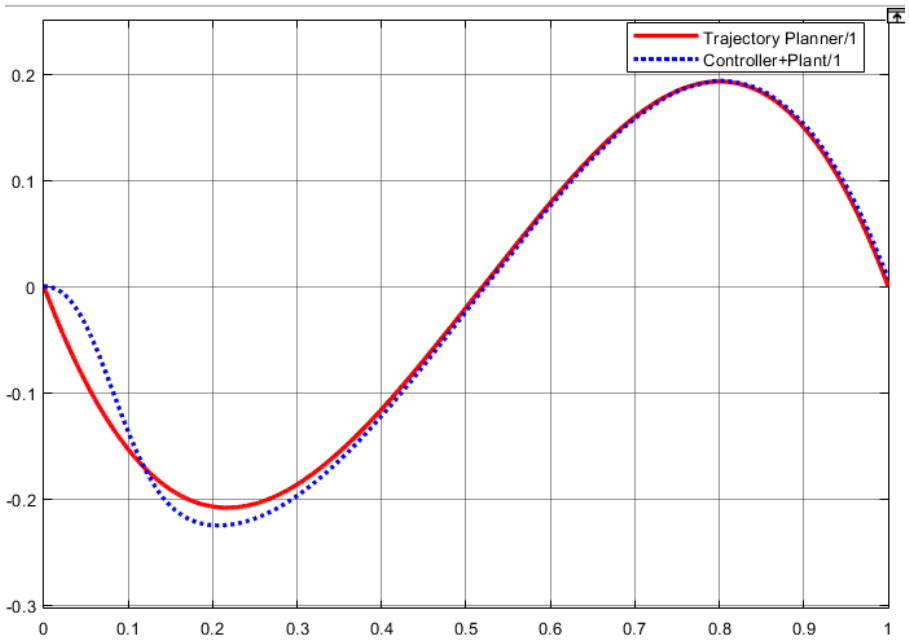


(a) Plant

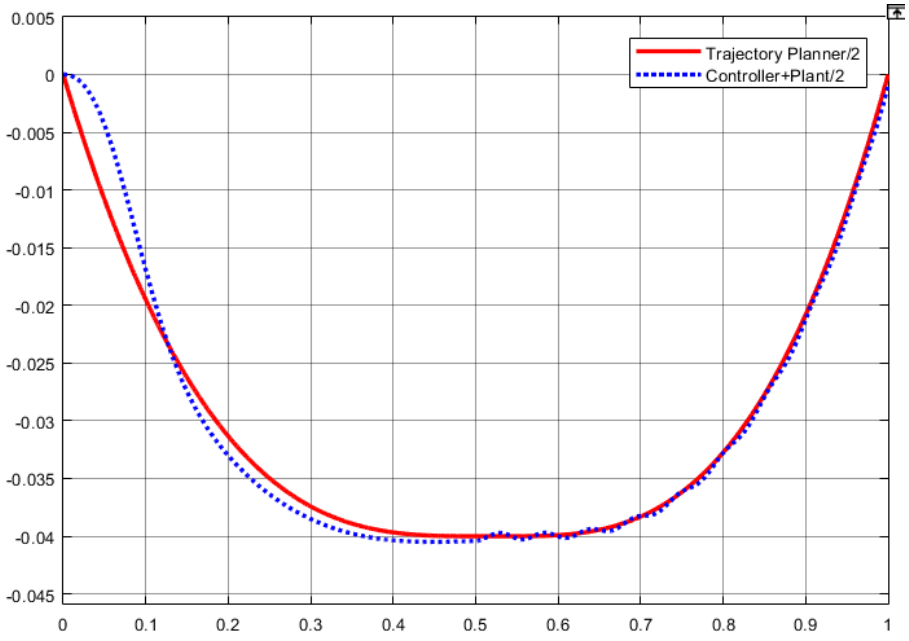


(b) Open-loop

Figure 4.16: Bode plot of the plant alone with the new geometric features and the open-loop chain.



(a) x variable trajectory tracking



(b) z variable trajectory tracking

Figure 4.17: New geometry and new controller. Planned trajectory and controlled system response for the pick-and-place task.

Chapter 5

Conclusions

In this thesis, the issue of high speed manipulation of objects has been studied from many points of views.

The first part of the study demonstrated that the most common technology nowadays, i e. industrial robotic arms, is not efficient in some given industrial environments. The problem was its physical encumbrance, as it is a voluminous rigid body, and the necessity of higher speed to reduce the total completion time of the pick-and-place task. The trajectory of the robot has been optimized by choosing the most studied planning techniques for off-line joint-space trajectory planning. The path profile followed the linear-elliptical-linear phases for each pick-and-place cycle, and the corresponding joint coordinates were interpolated via cubic spline curves in Matlab to generate a time-optimal low-jerk time law for each joint.
Then the simulated speed and acceleration profiles for each joint have been compared for different cycle completion times to find the minimum time for which the specified robot model guarantees the velocity profile to be under the maximum constructive limits of the robot. The optimal total task completion time was found to be 3.08 seconds, which is way above the time required to satisfy the bottle flow on the conveyor belt, and leads to the result that more than one robot is necessary to transfer all the bottles to their final position.

The second part of this thesis aimed to find innovative and unprecedented solutions to problem of high-speed tasks and physical volume of rigid robots. The proposed solutions deal with new soft and flexible manipulator models embedded in the final industrial chain layout consisting of two conveyor belts and a screw conveyor.
A consistent part of this section has the objective to understand what kind of manipulator is suitable for the pick-and-place task. This discussion has been pursued through the results of the simulations of the physical models of the

manipulators in order to compare their estimated performances.
The conclusion to which the simulations led is that the best two innovative manipulators most suitable to the task are the shooting manipulator and the controlled telescopic arm, while the least suitable manipulator kind is the soft inflated robot, for its low elongation speed.
The two simulated manipulators have different pros and cons: the shooting manipulator is very fast, cheap and definitely the smallest system that can complete the task, but on the other hand it is not fully controllable because of the impossibility of changing the gripper trajectory after the shot. Furthermore, the gripper choice for the shooting manipulator is not trivial because it must be a light-weight gripper and its gripping mechanism must not interfer with the manipulator wire. On the contrary, the telescopic arm manipulator allows the gripper trajectory to be controlled within certain limits, but is is a more complex structure with a nonlinear model, and worst of all it is affected by vibrations.
The oscillating behavior of the telescopic arm has been studied and observed in the literature and in our specific model with the results of multiple simulations, leading to the conclusion that the application of such system is possible in the present study case, but its geometrical dimensioning in the construction phase assumes crucial importance for its controllability.

Chapter 6

Appendix A - Source code listings

In the following lines, the Matlab source code for the robot declaration is reported.

```
clear all,close all, clc

dhParam=[0.15 -pi/2 0.4865 0
         0.475 0 0 0
         0 -pi/2 0 0
         0 pi/2 0.6 0
         0 -pi/2 0 0
         0 0 0.065 0];

robot=robotics.RigidBodyTree();

base1=robotics.RigidBody('base1');
jnt1=robotics.Joint('jnt1','revolute');
jnt1.PositionLimits=deg2rad([-180,180]);
setFixedTransform(jnt1,dhParam(1,:),'dh');
base1.Joint = jnt1;
addBody(robot,base1,'base')


link2 = robotics.RigidBody('link2');
jnt2 = robotics.Joint('jnt2','revolute');
link3 = robotics.RigidBody('link3');
jnt3 = robotics.Joint('jnt3','revolute');
```

```
link4 = robotics.RigidBody('link4');
jnt4 = robotics.Joint('jnt4','revolute');
link5 = robotics.RigidBody('link5');
jnt5 = robotics.Joint('jnt5','revolute');
link6 = robotics.RigidBody('link6');
jnt6 = robotics.Joint('jnt6','revolute');

jnt2.PositionLimits=deg2rad([-153,46]);
jnt3.PositionLimits=deg2rad([-235,55]);
jnt4.PositionLimits=deg2rad([-200,200]);
jnt5.PositionLimits=deg2rad([-115,115]);
jnt6.PositionLimits=deg2rad([-400,400]);

setFixedTransform(jnt2,dhParam(2,:),'dh');
jnt2.HomePosition=-pi/2;
setFixedTransform(jnt3,dhParam(3,:),'dh');
setFixedTransform(jnt4,dhParam(4,:),'dh');
setFixedTransform(jnt5,dhParam(5,:),'dh');
setFixedTransform(jnt6,dhParam(6,:),'dh');

link2.Joint = jnt2;
link3.Joint = jnt3;
link4.Joint = jnt4;
link5.Joint = jnt5;
link6.Joint = jnt6;

addBody(robot,link2,'base1')
addBody(robot,link3,'link2')
addBody(robot,link4,'link3')
addBody(robot,link5,'link4')
addBody(robot,link6,'link5')
```

Path planning and trajectory generation codes are reported as follows.

```
%% Path planning
% Constant velocity path
cvl=0.4; % constant velocity part length
xstart=0.8;
xl=xstart:-0.01:xstart-cvl;
yl=-0.5;
zl=0;
```

```
xlf=xstart:-0.01:xstart-cvl;
ylf=0.5;
zlf=0;

% Elliptic path

a=0.5; % horizontal radius
b=0.1; % vertical radius
y0=0; % x0,y0 ellipse centre coordinates
z0=0;
rad=0:0.01:pi;
y=y0+a*cos(rad);
z=z0+b*sin(rad);
xtrans=0.1;
ztrans=0.1;
x=linspace(xstart-cvl-xtrans,xstart+xtrans,length(y));
%figure
%plot(y,z)
%axis equal

%% Extracting samples from data

%elliptic path

ns=4;  % number of samples
c=ceil(length(y)/(ns-1));  % magnitude of intervals
 in which the path is divided

ys=zeros(ns,1); %sampled path initialization
zs=zeros(ns,1);

ys(1)=-y(1);    %initial and final points are assigned
ys(ns)=-y(end);
zs(1)=z(1);
zs(ns)=z(end);
xs(1)=x(1);
xs(ns)=x(end);

%computation of via points
for i=1:ns-2
```

```
ys(i+1)=-y(c*i);
zs(i+1)=z(c*i);
xs(i+1)=x(c*i);
end
% figure
% plot(ys,zs,'o'),legend('Path samples');
% axis equal

%% Trajectory planning from path
close all
gs=0.2; %gripper length

ik=robotics.InverseKinematics('RigidBodyTree', robot);
 %inverse kinematics definition
weights=[1 1 1 1 1 1];
ik.SolverParameters.MaxIterations=10000;
ikInitialGuess=robot.homeConfiguration;

%computation of inverse kinematics for each point

Qxlsol=ikInitialGuess;
%linear path 1
for i=1:length(xl)
xpl{i}=eul2tform([0 0 pi]);
xpl{i}(1:3,4)=[xl(i) yl zl+gs];
[Qxlsol,~]=step(ik,'link6',xpl{i}, weights, Qxlsol);
% figure
% show(robot,Qxlsol);
qlSol(:,i)=[Qxlsol.JointPosition]';
end

%linear path 2

Qxlfsol=Qxlsol;
for i=1:length(xl)
xplf{i}=eul2tform([0 0 pi]);
xplf{i}(1:3,4)=[xlf(i) ylf zlf+gs];
[Qxlfsol,~]=step(ik,'link6',xplf{i}, weights, Qxlsol);
% figure
% show(robot,Qxlsol);
```

```
qlfSol(:,i)=[Qxlfsol.JointPosition]';
end

%elliptic path
for i=1:ns
xp{i}=eul2tform([0 0 pi]);
xp{i}(1:3,4)=[xs(i) ys(i) zs(i)+ztrans+gs];
[Qxsol,~]=step(ik,'link6',xp{i}, weights, Qxlfsol);
%figure
%show(robot,Qxsol);
qSol(:,i)=[Qxsol.JointPosition]';
end

%repositioning elliptic path
for i=1:ns
%xsrep=flip(xs);
ysrep=flip(ys);
xprep{i}=eul2tform([0 0 pi]);
xprep{i}(1:3,4)=[xs(i) ysrep(i) zs(i)+ztrans+gs];
[Qxrepsol,~]=step(ik,'link6',xprep{i}, weights, Qxlsol);
%figure
%show(robot,Qxrepsol);
qrepSol(:,i)=[Qxrepsol.JointPosition]';
end

%% Jacobian and inverse dinamics
Drobot=robot;
Drobot.DataFormat = 'row';
Drobot.Gravity = [0 0 -9.81];

for i= 1:length(xl)
% nqlSol=qlSol;
% nqlSol(2,i)=qlSol(2,i)+pi/2;
Jl{i}=geometricJacobian(robot,qlSol(:,i)','link6');
qdot(:,i)=Jl{i}\[1 0 0 0 0 0]';
Tl(:,i) = inverseDynamics(Drobot,qlSol(:,i)',qdot(:,i)');

end

for i= 1:length(xl)
```

```
% nqlSol=qlSol;
% nqlSol(2,i)=qlSol(2,i)+pi/2;
Jlf{i}=geometricJacobian(robot,qlfSol(:,i)','link6');
qfdot(:,i)=Jlf{i}\[-1 0 0 0 0 0]';
Tlf(:,i) = inverseDynamics(Drobot,qlfSol(:,i)',qfdot(:,i)');

end

for i=1:ns
% nqSol=qSol;
% nqSol(2,i)=qSol(2,i)+pi/2;
T(:,i)=inverseDynamics(Drobot,qSol(:,i)');
end

%% Spline interpolation Joint Position

nj=6; %number of joints

trfin=0.12; %motion time of the transition path in seconds
tlfin=cvl; %motion time for linear path set to cvl, imposed
                %by the speed of the bottles
tfin=0.9; % elliptic motion time set to 0.8 sec
trepfin=0.9; %repositioning time

tr1=linspace(tlfin,tlfin+trfin,ns);
tr2=linspace(tlfin+trfin+tfin,
tlfin+trfin+tfin+trfin,ns-1);
ttr=0:0.001:trfin;

tl=linspace(0,tlfin,length(xl));
tl2=linspace(tlfin+trfin+tfin+trfin,
tlfin+trfin+tfin+trfin+tlfin,
length(xl));
ttl=0:0.001:tlfin;

te=linspace(tlfin+trfin+0.001,tlfin+trfin+tfin,ns);
tte=0:0.001:tfin;
trep=linspace(tlfin+trfin+tfin+trfin+tlfin+trfin,
tlfin+trfin
+tfin+trfin+tlfin+trfin+trepfin+trfin,ns);
```

```
tt=0:0.001:tfin+2*tlfin+4*trfin+trepfin;
t=[tl(1:end-1) te(1:end-1) tl2(1:end-1) trep];

Tsim=tfin+2*tlfin+4*trfin+trepfin;
for i=1:3
qtot(i,:)=[qlSol(i,1:length(xl)-1) qSol(i,1:ns-1)
qlfSol(i,1:length(xl)-1) qrepSol(i,1:ns-1) qlSol(i,1)];
stot(i,:)=spline(t,[qdot(i,1) qtot(i,:) qdot(i,1)],tt);
figure
plot(t,rad2deg(qtot(i,:)),'o',tt,rad2deg(stot(i,:))),
legend('Interpolated points',
['Displacement of joint q' num2str(i)]),
title(['Cubic Spline Interpolation for Joint ' num2str(i)]) ;
axis([0 Tsim -100 100])
xlabel('time [s]'),ylabel('[deg]');
end

%For wrist joints velocities are not imposed

for i=4:6
qtot(i,:)=[qlSol(i,1:length(xl)-1) qSol(i,1:ns-1)
qlfSol(i,1:length(xl)-1) qrepSol(i,1:ns-1) qlSol(i,1)];
stot(i,:)=spline(t,qtot(i,:),tt);
figure
plot(t,rad2deg(qtot(i,:)),'o',tt,rad2deg(stot(i,:))),
legend('Interpolated points',
['Displacement of joint q' num2str(i)]),
title(['Cubic Spline Interpolation for Joint ' num2str(i)]) ;
axis([0 Tsim -100 100])
xlabel('time [s]'),ylabel('[deg]');
end
stotdeg=180*stot/pi;

Tsim=tfin+2*tlfin+4*trfin+trepfin;
Tspace=linspace(0,Tsim,length(tt));

%Translation of the vectors into a signal for Simulink

s1=timeseries(stotdeg(1,:),Tspace);
```

```
s2=timeseries(stotdeg(2,:)+90,Tspace);
s3=timeseries(stotdeg(3,:),Tspace);
s4=timeseries(stotdeg(4,:),Tspace);
s5=timeseries(stotdeg(5,:),Tspace);
s6=timeseries(stotdeg(6,:),Tspace);
```

In the following lines, the code for the interpretation of the results of the simulation is reported.

```
%% Data analysis

tic
sim('Simulation_Displacement');
toc
close all

jointsim{1}=get(joint1data,'Data');
jointsim{2}=get(joint2data,'Data');
jointsim{3}=get(joint3data,'Data');
jointsim{4}=get(joint4data,'Data');
jointsim{5}=get(joint5data,'Data');
jointsim{6}=get(joint6data,'Data');

time=get(joint1data,'Time');
ini=40;

vmax=[180 180 180 320 360 500]';

for j=1:3
figure

subplot(3,1,1)
ppp=plot(time(ini:end),rad2deg(jointsim{j}(ini:end,1)),'b');
title(['Joint ' num2str(j) ' Kinematic Variables' ]);
line1=line([0 time(end)],[vmax(j) vmax(j)]);
line1.LineStyle='--';
line2=line([0 time(end)],[-vmax(j) -vmax(j)]);
line2.LineStyle='--';
ylabel('[deg/s]'),xlabel('time [s]');
legend(ppp,'Speed');
axis([0 Tsim -300 300])
```

```
grid on
subplot(3,1,2)
plot(time(ini:end),jointsim{j}(ini:end,2),'r'),
ylabel('[deg/s^{2}]'),xlabel('time [s]'),
legend('Acceleration');
axis([0 Tsim -50 50])
grid on
subplot(3,1,3)
plot(time(ini:end),jointsim{j}(ini:end,3),'k'),
ylabel('[Nm]'),xlabel('time [s]'),
legend('Actuator Torque');
axis([0 Tsim -1000 1000])
grid on
end

for j=4:6
figure

subplot(3,1,1)
ppp=plot(time(ini:end),rad2deg(jointsim{j}(ini:end,1)),'b');
title(['Joint ' num2str(j) ' Kinematic Variables' ]);
line1=line([0 time(end)],[vmax(j) vmax(j)]);
line1.LineStyle='--';
line2=line([0 time(end)],[-vmax(j) -vmax(j)]);
line2.LineStyle='--';
ylabel('[deg/s]'),xlabel('time [s]');
legend(ppp,'Speed');
axis([0 Tsim -550 550])
grid on
subplot(3,1,2)
plot(time(ini:end),jointsim{j}(ini:end,2),'r'),
ylabel('[deg/s^{2}]'),xlabel('time [s]'),
legend('Acceleration');
axis([0 Tsim -50 50])
grid on
subplot(3,1,3)
plot(time(ini:end),jointsim{j}(ini:end,3),'k'),
ylabel('[Nm]'),xlabel('time [s]'),
legend('Actuator Torque');
axis([0 Tsim -1 1])
```

```
grid on
end
```

Simulated dynamic model of the shooting manipulator code.

```
function [zd,z0,deltaL] = shooting_manipulator(z,Iimp,phi)
%% z=[x,xd,y,yd,theta,thetad]'
k=83;
m=1.9e-03;
r=1e-02;
I=1.8e-07;
g=9.81;
L0=0.71;
z0=[0 cos(phi)*Iimp/m 0 sin(phi)*Iimp/m 0 0]';
%% Matrices for cases a) and b)

A=[0 1 0 0 0 0;
    -k/m 0 0 0 r*k/m 0;
    0 0 0 1 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 1;
    r*k/I 0 0 0 -r^2*k/I 0];

a=[0 k*L0/m 0 -g 0 -r*k*L0/I]';

B=[0 1 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 1 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 1;
    0 0 0 0 0 0];

b=[0 0 0 -g 0 0]';

x=z(1);
theta=z(5);
deltaL=(x-L0)-r*theta;

if deltaL>=0
    zd=A*z+a;
else
```

```
    zd=B*z+b;
end

end
```

Simulation of the controlled shooting manipulator system.

```
close all, clear all, clc

xt=0;
yt=0.6;
zt=-0.3;
eps=0.01;
tfin=0.7;
niter=50;

for i=0:niter
  Iimp=0.0002*i;
  phi=-atan((abs(zt)-0.2)/(yt));
    sim('ShootingControlSimulink');
    ysim=y.data;
    zsim=z.data;

    for j=1:length(ysim)
        if abs(ysim(j))<=abs(yt)+eps && abs(ysim(j))>=abs(yt)-eps
         && abs(zsim(j))<=abs(zt)+eps && abs(zsim(j))>=abs(zt)-eps
            Ifin(j,i+1)=Iimp;
           %phifin(j,i+1)=phi;
        else
            Ifin(j,i+1)=100;
           %phifin(j,i+1)=100;
        end
    end
end
%% Results analysis and best combination choice
I=10;
for i=0:niter
    for j=1:length(ysim)
        if Ifin(j,i+1)<100 && Ifin(j,i+1)>0
            I(end+1)=Ifin(j,i+1);
        end
```

```
    end
end
If=min(I)

%% Final simulation with the desired value
Iimp=If;

    sim('ShootingControlSimulink');
    yfin=y.data;
    zfin=z.data;
    plot(yfin,zfin);
```

Simulated dynamic model for the telescopic manipulator code. Control for only direction x.

```
function zxd = telescopic_arm(zx,mtx,L)
% zx=[xd; x];
E=200e09;
r1=0.02;
r2=0.017;
p1=0.003;
p2=0.002;
A1=pi*r1^2;
A2=pi*r2^2;
ro=8000;
I1=pi*r1^3*p1;
I2=pi*r1^3*p2; % Second moment of area valid
        % both for x and y direction
am=124.4444;
ak=3.1111e-05;

Mred=(ro*L/420)*[A1*156+A2*156 -A1*22*L+A2*22*L A2*54 -A2*13*L
    -A1*22*L+A2*22*L A1*4*L^2+A2*4*L^2 A2*13*L -A2*3*L^2
    A2*54 A2*13*L A2*156 -A2*22*L
    -A2*13*L -A2*3*L^2 -A2*22*L A2*4*L^2];


Kred=(E/(L^3))*[I1*12+I2*12 -I1*6*L+I2*6*L -I2*12 I2*6*L
    -I1*6*L+I2*6*L I1*4*L^2+I2*4*L^2 -I2*6*L I2*2*L^2
    -I2*12 -I2*6*L I2*12 -I2*6*L
    I2*6*L I2*2*L^2 -I2*6*L I2*4*L^2];
```

```
Cred=am*Mred+ak*Kred;

A=[-Mred\Cred -Mred\Kred; eye(4) zeros(4)];
B=[inv(Mred) zeros(4)]';
zxd=A*zx+B*[0 0 0 mtx]';

end
```

Trajectory planning for the telescopic manipulator picking of the object.

```
clear all, close all, clc
xt=0; %x coord. position of the bottle
zt=-0.04;%z coord. position of the bottle

c=-0.02:0.001:xt+0.02;
tt=0:0.001:1;
t=[0,0.5:0.001:0.5+0.001*(length(c)-1),1]';
x=[0,-0.02:0.001:xt+0.02,0]';
z=[0,zt*ones(1,length(c)),0]';
xs=spline(t,x,tt);
xr=timeseries(xs,tt);
zs=spline(t,z,tt);
zr=timeseries(zs,tt);
plot(xr);
figure
plot(zr);
```

Bibliography

[1] B. Siciliano, L Sciavicco, L. Villani, G. Oriolo, Robotics Modelling, Planning and Control, 631, Advanced Textbooks in Control and Signal Processing (2009)

[2] A. Gasparetto, V. Zanotto, A new method for smooth trajectory planning of robot manipulators, Mechanism and Machine Theory, 42, 455-471 (2006)

[3] A. Gasparetto, V. Zanotto, Optimal trajectory planning for industrial robots, Advances in Engineering Software, 41, 548-556 (2009)

[4] A. Valente, S. Baraldo, E. Carpanzano, Smooth trajectory generation for industrial robots performing high precision assembly processes, CIRP Annals-Manufacturing Technology, 66, 17-20 (2017)

[5] C.G. Lo Bianco, F. Ghilardelli, A scaling algorithm for the generation of jerk-limited trajectories in the operational space, Robotics and Computer-Integrated Manufacturing, 44, 284-295 (2017)

[6] R. J. Moreno Masey, J. O. Gray, T.J. Dodd, D. G. Caldwell, Elliptical point to point trajectory planning using electronic cam motion profiles for high speed industrial pick and place robots, ETFA 2009. IEEE Conference on Emerging Technologies & Factory Automation, (2009)

[7] T. Borangiu, Visual conveyor tracking for "pick-on-the-fly" robot motion control, Conference: Advanced Motion Control, (2002)

[8] T.Huang, P.F.Wang, J.P.Mei, X.M.Zhao, D.G.Chetwynd, Time-minimum trajectory planning of a 2-DOF translational parallel robot for pick-and-place operations, CIRP Annals, 56, 365-368 (2007)

[9] S. Kucuk , Optimal trajectory generation algorithm for serial and parallel manipulators, Robotics and Computer-Integrated Manufacturing, 48, 219-232 (2017)

[10] T.Chettibi, H.E.Lehtihet, M.Haddad, S.Hanchi, Minimum cost trajectory planning for industrial robots, European Journal of Mechanics - A/Solids, 23, 703-715 (2004)

[11] F. Bourbonnais, P. Bigras, I.A. Bonev, Minimum-time trajectory planning and control of a pick-and-place five-bar parallel robot, IEEE/ASME Transactions on Mechatronics, 20, 740-749 (2014)

[12] R.Mattone, G.Campagiorni, F.Galati, Sorting of items on a moving conveyor belt. Part 1: a technique for detecting and classifying objects, Robotics and Computer-Integrated Manufacturing, 16, 73-80 (2000)

[13] R.Mattone, M.Divona, A.Wolf, Sorting of items on a movin conveyor belt. Part 2: performance evaluation and optimization of pick-and-place operations, Robotics and Computer-Integrated Manufacturing, 16, 81-90 (2000)

[14] H. IşılBozma, M.E.Kalalıoğlu, Multirobot coordination in pick-and-place tasks on a moving conveyor, Robotics and Computer-Integrated Manufacturing, 28, 530-538, (2012)

[15] L. Comba, G. Belforte, P. Gay, Plant layout and pick-and-place strategies for improving performances in secondary packaging plants of food products, Packaging technology and science, 26, 339-354 (2012)

[16] Y. Huang, R. Chiba, T.Arai, T. Ueyama, J. Ota, Robust multi-robot coordination in pick-and-place tasks based on part-dispatching rules, Robotics and Autonomous Systems, 64, 70-83 (2015)

[17] S. Daoud, H. Chehade, F. Yalaoui, L. Amodeo, Efficient metaheuristics for pick and place robotic systems optimization, Journal of Intelligent Manufacturing, 25, 27-41

[18] N. Houshangi , Control of a robotic manipulator to grasp a moving target using vision, Proceedings., IEEE International Conference on Robotics and Automation, (1990)

[19] A. Šabanović, K. Jezernik, K. Wada, Chattering free sliding modes in robotic manipulators control, 14, 17-29, (2009)

[20] D.B. Zhang, L. Van Goo, A. Oosterlinck, Stochastic predictive control of robot tracking systems with dynamic visual feedback, Proceedings., IEEE International Conference on Robotics and Automation (1990)

[21] R. Menasri, A. Nakib, B. Daachi, H. Oulhadj, P. Siarry, A trajectory planning of redundant manipulators based on bilevel optimization, Applied Mathematics and Computation, 250, 934-947 (2015)

[22] H. Liu, X. Lai, W. Wu, Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints, Robotics and Computer-Integrated Manufacturing, 29, 309-317 (2013)

[23] M. Marcos, J.A. Tenreiro Machado, T.-P. Azevedo-Perdicoúlis, A multi-objective approach for the motion planning of redundant manipulators, Applied Soft Computing, 12, 589-599 (2012)

[24] A. Piazzi, A. Visioli, Global minimum-jerk trajectory planning of robot manipulators, IEEE Transactions on Industrial Electronics, 47, 140-149 (2000)

[25] D. E. Moulton, T. Lessinnes, S. O'Keeffe, L. Dorfmann, A. Goriely, Elastic secrets of the chameleon tongue, Proceedings Royal Society A, 472 (2016)

[26] A. Debray, Manipulators inspired by the tongue of the chameleon, Bioinspiration and Biomimetics, 6 (2011)

[27] T. Hatakeyama, H.Mochiyama, Shooting manipulation inspired by chameleon, IEEE/ASME Transactions on Mechatronics 18, 2 (2013)0

[28] H.Mochiyama, H. Nakajima, T. Hatakeyama, Chameleon-like shooting manipulator for accurate 10-meter reaching, 2015 10th Asian Control Conference (ASCC) (2015)

[29] H. Arisumi, K. Komoriya, Catching motion of casting manipulation, Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, (2000)

[30] D. Rus, M. T. Tolley, Design, fabrication and control of soft robots, Nature, 521, 467-475 (2015)

[31] D. Trivedi, A. Lotfi, C.D. Rahn, Geometrically Exact Models for Soft Robotic Manipulators, IEEE Transactions on Robotics, 24, 773-780 (2008)

[32] E. Y. Yarbasi, E. Samur, Design and evaluation of a continuum robot with extendable balloons, Mechanocal Sciences, 9, 51-60 (2018)

[33] [online] https://search-ext.abb.com/library/Download.aspx?DocumentID=3HAC023604-007&LanguageCode=it&DocumentPartId=&Action=Launch

[34] [online] https://new.abb.com/products/3HAC021770-001/irb-1600

[35] J.F. Hunt, H. Zhang, Z. Guo, F. Fu, Cantilever Beam Static and Dynamic Response Comparison with Mid-Point Bending for Thin MDF Composite Panels, BioResources, 8 (2013)

[36] M. Benosman, G. Le Vey, Control of flexible manipulators: A survey, Robotica, 22, 533–545 (2004)

[37] Buntara S. Gan, An Isogeometric Approach to Beam Structures, 233. Springer International Publishing, (2018)

[38] E. Bayo, A finite-element approach to control the end-point motion of a single-link flexible robot, Journal of field robotics, 4, 63-75 (1987)

[39] A. De Luca, L. Lanari, G. Ulivi , End-Effector Trajectory Tracking in Flexible Arms: Comparison of Approaches Based on Regulation Theory, Advanced Robot Control, 162, 190-206 (1991)

[40] A. De Luca, L. Lanari, P. Lucibello, S. Panzieri, G. Ulivi, Control experiments on a two-link robot with a flexible forearm, 29th IEEE Conference on Decision and Control, (1990)

[41] M. Balas, Feedback control of flexible systems, IEEE Transactions on Automatic Control, 23, 673-679 (1978)

[42] W.P.Li, B.Luo, H.Huang, Active vibration control of Flexible Joint Manipulator using Input Shaping and Adaptive Parameter Auto Disturbance Rejection Controller, Journal of Sound and Vibration, 363, 97-125 (2016)

[43] X. Xie, J. Huang, Z. Liang, Vibration reduction for flexible systems by command smoothing, Mechanical Systems and Signal Processing, 39, 461-470 (2013)
[44] G. Genta, Vibration Dynamics and Control, 856. Springer US, Boston (2009)
[45] R.D. Robinett III, J. Feddema, G.R. Eisler, C. Dohrmann, G.G. Parker, D.G. Wilson, D. Stokes, Flexible Robot Dynamics and Controls, IFSR International Series in Systems Science and Systems Engineering, 339. Springer US, (2002)
[46] E. A. Croft, R. G. Fenton, B.Benhabib, Optimal Rendezvous-Point Selection for Robotic interception of Moving Objects, IEEE Transactions onSystems, Man and Cybernetics- Part B: Cybernetics, 28, 192-204 (1998)
[47] D. Hujic, An active prediction, planning and execution system for moving object interception, M.A.Sc thesis, Dept. Mechanical Engineering, Toronto University, Toronto, Canada (1995)
[48] P. K. Allen, Member, A. Timcenko, B. Yoshimi, P. Michelman, Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System, IEEE Transactions on Robotics and Automation, 9, 152-165 (1993)
[49] H.Mevadaa,D. Patelb, Experimental determination of structural damping of different materials, 12th International Conference on Vibration Problems, 2015
[50] J. F. Hall, Problems encountered from the use (or misuse) of Rayleigh damping, Earthquake Engineering and Structural Dynamics, 35, 525–545 (2006)