POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering

ADAPTIVE AND LESS CONSERVATIVE GENERATION OF HUMAN SWEPT VOLUMES FOR SAFE COOPERATION WITH ROBOTS

Supervisor: Prof. Paolo Rocco

Co-supervisors: Andrea Casalino

Dr. Andrea Maria Zanchettin

M.Sc. Thesis by: Alberto Brameri

Student ID 852384

Academic Year 2017 - 2018

Abstract

Nowadays collaboration between human operators and robots is a growing field. To ensure the safety of the worker, while preserving robot's productivity, several control strategies have been developed.

One effective approach is based upon the construction of the "swept volumes", that are the volumes containing human predicted motion for next instants of time. To obtain this prediction, the operator is tracked with depth sensors and then measurements are exploited by a Linear Kalman Filter, that estimates human speed, acceleration and jerk.

In past works, swept volumes were generated in a conservative way, because the algorithm, that computes the reachable set of the human operator, utilized only the velocity estimate, while it assumed that acceleration was always set to its maximum or minimum value, to strengthen the safety constraint.

The goal of this thesis is to improve the computation of swept volumes, with the aim of obtaining reduced volumes and consequently enhancing robot's productivity. To achieve this result, first, bounded accelerations have been replaced with estimated ones, then, bounded jerk has been introduced in the algorithm.

Another result, achieved by this thesis, was the realization of a better estimate: an Adaptive Kalman Filter has been implemented; the peculiarity of this new filter is that it adapts the process noise covariance matrix.

To validate the new algorithm and the Adaptive Kalman filter, first computer simulations have been done and then some experiments, simulating real Human-Robot Collaboration (HRC) scenarios, using ABB YuMi robot, have been performed.

Sommario

Oggi il campo della collaborazione tra operatori umani e robot è in forte crescita. Per garantire la sicurezza del lavoratore, pur preservando il rendimento del robot, sono state sviluppate diverse strategie di controllo.

Uno di queste si basa sulla costruzione degli "swept volumes", cioè i volumi che contengono la predizione del moto dell'umano per i prossimi istanti di tempo. Per ottenere questa previsione, l'operatore viene monitorato con sensori di profondità e le misure ottenute vengono inviate a un filtro lineare di Kalman, che stima la velocità, l'accelerazione e il jerk dell'operatore.

Negli studi precedenti, gli "swept volumes" erano generati in modo conservativo, poiché l'algoritmo, che calcola l'insieme di punti raggiungibili per l'uomo, utilizzava solo la stima della velocità, mentre veniva ipotizzato che l'accelerazione fosse sempre impostata al valore massimo o minimo, per rafforzare il vincolo di sicurezza.

L'obiettivo di questa tesi è quello di migliorare l'algoritmo che calcola gli "swept volumes", per ottenere volumi ridotti, aumentando così la produttività del robot. Per ottenere questo risultato, innanzitutto, la stima dell'accelerazione è stata introdotta nell'algoritmo che calcola l'insieme raggiungibile, rimuovendo così l'ipotesi di movimento umano sempre ad accelerazione massima o minima. Poi il jerk, impostato al suo valore massimo o minimo, è stato aggiunto all'algoritmo.

Un altro aspetto su cui la tesi fornisce un contributo è il miglioramento della stima. È stato infatti implementato un filtro di Kalman adattivo: la peculiarità di questo nuovo filtro è quella di adattare la matrice di covarianza del rumore sul processo.

Per validare il nuovo algoritmo e il filtro adattivo, dapprima sono state eseguite simulazioni al computer e quindi sono stati condotti esperimenti di collaborazione uomo-robot, utilizzando il robot ABB YuMi.

Contents

List of Figures

List of Tables

Chapter 1

Thesis Outline

Industrial manipulators represent a well-established technology in several industry sectors, such as automotive, or for machines tending and parts movement. However, their diffusion in Small and Medium-sized Enterprises is hampered by an insufficient flexibility. Human-Robot Collaboration (HRC) represents a promising solution to such a problem, as cooperation between robots and workers could greatly increase robots flexibility, and, at the same time, increase the productivity, because robots perform repetitive operations faster than a human operator. Nevertheless, the deployment of industrial robots in HRC scenarios poses new challenges for robot manufacturers and system integrators: guaranteeing safety for human operators cooperating with robots, while maximizing productivity. Robots should avoid collisions and reduce the risk of consequent injuries. At the same time, the need for safety must not diminish robots productivity, nor should it disrupt the possibility of task completion or generate a risk of damages for the manipulator or the production setup. In the last years, many robots, specifically built for HRC, have been introduced onto the market; furthermore, control strategies, to ensure workers' safety, preserving robots' productivity, have been developed. One of these is based on the generation of human "swept volumes": they are the portion of the space, that is predicted to be occupied by the human in a given time interval, based on his/her current state of motion, perceived with some optical sensors.

The aim of this thesis is to improve swept volumes generation, further enhancing robot's productivity, while preserving safety. To achieve this result, an improvement of the algorithm, that computes swept volumes, is proposed and a technique, making swept volumes generation adaptive, is introduced.

1.1 Thesis Contributions

In this thesis, two main contributions are given:

1. A modification of the algorithm that computes the Reachable Set of the human, that dictates how large swept volumes will be. The modified algorithm is less conservative than the original one. In this way, swept volumes will be smaller, enhancing the productivity of the robot, while maintaining human worker's security.

2. A technique to adapt the Kalman filter process noise covariance matrix. The Adaptive Kalman Filter permits to have better estimates and, if jerk is considered as process noise, instead of being part of the state, to adapt jerk bounds, that are used to compute the reachable set of the human in the modified algorithm.

Both contributions have been experimentally validated in a realistic setup, involving collaboration between a human operator and an industrial robot.

1.2 Thesis Organization

The thesis is organised as follows:

In **Chapter 2** an overview of Human Robot Collaboration and the state of the art for Detecting and Predicting Human Motion techniques are introduced.

In **Chapter 3** a review of safety constrains, model of Human Motion, Constrained Linear Kalman Filter, Reachable Set and swept volumes algorithms and finally robot's velocity scaling is given.

In **Chapter 4** the modifications to the Reachable Set algorithm are presented: first a simple change is discussed and then the main modification is given.

In **Chapter 5** the Adaptive Kalman Filter is introduced: first a brief outline of different adaptation methods is given, then the chosen method and the changes to the Constrained Linear Kalman Filter are presented.

In **Chapter 6** some Matlab simulations, used to check the correctness and the efficiency of algorithms and techniques introduced in previous chapters, are carried out.

In **Chapter 7** the strategies proposed in previous chapters are validated by an experiment of HRC, in which a robot and a human operator cooperate to assemble an electronic board.

In **Chapter 8** conclusions are drawn and possible future developments are proposed.

Chapter 2

State of the Art

The industrial robots' market is growing year after year (see [40]); these robots are able to perform fast and accurate tasks of various kind, like welding, painting, packaging, palletizing, etc. Up to the past years, robots were placed in protected environments by means of physical infrastructures (fences or optical barriers) for safety reasons (see Fig. 2.1). But nowadays, the idea of a collaboration between human workers and robots (HRC) is rapidly spreading: in fact, there are too complex operations to be performed by the robot and vice versa repetitive ones in which the precision and reliability of the robots cannot be matched by human.

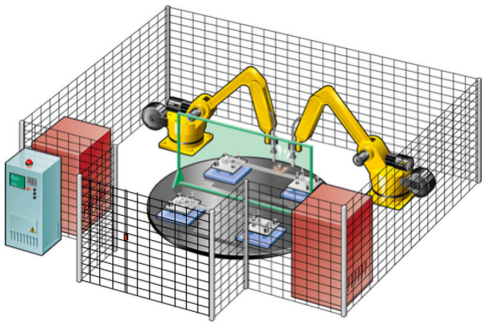


Figure 2.1: Traditional industrial robotic cell with safety barriers.

In this new scenario robots and humans work together, so physicals barriers must be removed (see Fig. 2.2). To ensure the safety of the human operator and to comply with European standards (see [2], inspired by American standard [1]), a new kind of robots has been developed. They are lighter, without edges, sometimes covered with paddings to damp the effects of any impacts. The robots are also more sensitized than in the past: they can recognize the presence of obstacles and interrupt the operation in case of collisions. Many of these robots are also equipped with kinematic redundancy, i.e. they have a number of joints greater than the strictly necessary one. This gives more naturalness to movements of the

arm and may allow to perform certain operations enhancing security. Sometimes they also have a two arms configuration(see, for example, YuMi, recently marketed by ABB).

Figure 2.2: HRC scenario where human workers and robot work directly in touch.

Furthermore many control strategies, to guarantee a safe HRC, have been developed. A technique is the Human motion Detection and Tracking (HDT) and human motion prediction, which is also known as Intention Estimation (IE), to generate safety constraints for collision avoidance. HDT consists in detecting the presence of one or more human beings inside the supervised environment and to track their movements on the basis of a series of consecutive "descriptions" of the scene provided by one or more sensors. On the other hand, IE consists in predicting the intentions of a human in a structured environment on the basis of the tracked positions.

Techniques to perform HDT in an industrial environment, using single camera or multiple cameras, are described in [11], while high-visibility industrial clothing detection strategies based on RGB and IR cameras have been proposed in [29]. Approaches based on pressure-sensitive sensors mounted on the floor have been proposed as well, like for instance [32] and [4], in which the concept of "smart floor" is introduced. Finally, examples of HDT relying on RGB-D sensor can be found in [30].

Regarding IE, in [19] techniques combining vision and physiological signal measurement for human motion estimation during HRC are presented, while [5] describes a system for predicting the probability of an accident in a HRC industrial scenario based on a dynamic stochastic model of human motion. Other approaches utilize Hidden Markov models to estimate and reshape human intention, as for instance [9]. A technique, for human walking IE, through the use of a Kalman Filter, is proposed in [46]. A method, that employs neural networks to estimate motion intention, can be found in [20]. Lastly, in recent works, algorithms to infer human

arms motion estimation are proposed: one is based on Multiple Model Filtering (see [39]), the other is an Expectation-Maximization Algorithm with online Model Learning (see [38]).

A technique, combining HDT and IE, is based on "Swept Volumes" generation: swept volumes are the portions of the environment, containing human predicted motion for next instants of time. To obtain these volumes, first the operator is tracked with depth sensors, then human motion is predicted. This method of safe HRC is developed in [36], that constitutes the main background of this thesis (see also [37], for a summary of the Ph.D. thesis). Other studies on swept volumes generation can be found in [42] and [23].

Other strategies for HRC are proposed in [12], in which a Depth Space approach to Collision Avoidance is introduced, and [22], where phase estimation makes it possible to classify human actions, and to generate a corresponding robot trajectory before the human finishes his/her movement.

In the last two years, many other control strategies for safe HRC have been developed. In [34], the focus is on prediction of arm occupancy, to implement a safe collision-avoidance strategy. [15] presents an approach to formulate a cost function, for motion planner, so that the motion is both safe and efficient. A real-time gesture-based HRC can be found in [10]; the human pose in this work is estimated using a self organizing map approach, and particular attention is paid to hand-finger pose estimation. An energy-based technique, to compute safety constraints, is introduced in [25]. In [33], a probabilistic approach, to generate safe robot trajectories, is proposed. [14] presents a method, based on distance calculation and discrete detection, for dynamic obstacle avoidance in a HRC scenario. In [28], a novel technique for active collision avoidance, driven by vision sensors, is given. Finally, a HRC framework, based on the understanding of human intention, can be found in [21].

Chapter 3

Background on Swept Volumes Generation

In this chapter, a review of the main concepts developed in [36] is given. While this is not a contribution of this thesis, it is essential to fully understand the algorithms presented in next chapters, and to make the present document self-contained.

3.1 Safety Constraints

If an industrial manipulator is performing a task in the presence of both static and dynamic obstacles (for instance, human workers moving inside the robot workspace), the problem of possible collisions between the robot itself and these obstacles could easily arise. In order to safely accomplish the task by avoiding such collisions, the robot control system must be able to detect the presence of the obstacles, to monitor the distance between them and the manipulator, and to adapt the robot velocity accordingly.

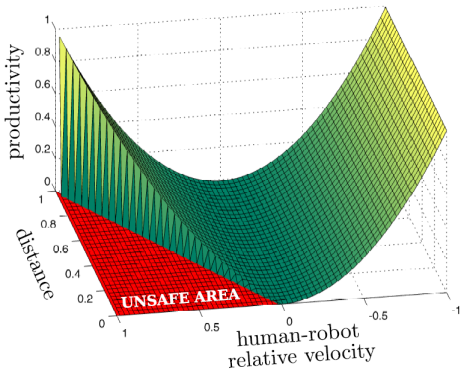


Figure 3.1: Trade-off between productivity and safety.

Figure 3.1 shows the relationship between production (in terms of robot veloc-

ity) and safety. As the distance between the robot and the human becomes smaller, the velocity of the robot should be reduced, thus decreasing the productivity of the robot. On the other hand, even in case of a reduced separation distance, the robot should continue its task if its velocity is oriented so that the distance with the human operator will increase.

To achieve a fruitful trade-off between safety and productivity it is possible to consider the first as a hard constraint, in which respect the latter could be somehow maximized. In other words, safety constraints must be defined in order to ensure that the entire kinematic chain of the manipulator performs a collision-free motion during task execution. These constraints must necessarily take into account the kinematic configuration of the manipulator, known obstacles positions and their geometry.

3.1.1 Safety constraints for a point obstacle

First, derivation of safety constraints for a point obstacle is reported, as introduced in [49, 51]. A generic robot link, whose endpoints are at positions $\boldsymbol{r_a}$ and $\boldsymbol{r_b}$ and a point obstacle $\boldsymbol{r_{obst}}$ are considered (see Fig. 3.2).

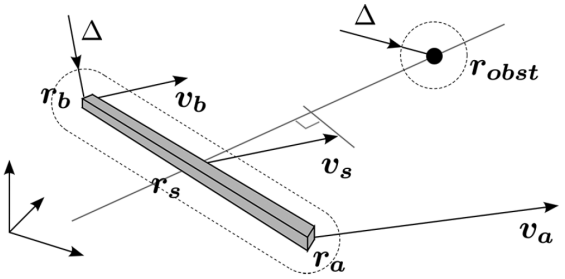


Figure 3.2: A rigid beam representing one link and a point-shaped obstacle.

At all time, the robot trajectory must obey the following safety requirement expressed as an inequality:

$$velocity \cdot T_s \leq \max\left(0, distance - \Delta\right) \tag{3.1}$$

where "velocity" represents the robot velocity in the direction of the obstacle, the worst-case braking time T_s possibly depends on the robot payload [1], "distance" is the distance between the robot and a generic obstacle and Δ allows to take into account both robot and obstacle dimensions, sensor uncertainties and ultimately an actual clearance.

For a generic point $\boldsymbol{r_s}$ on the robot link, with velocity $\boldsymbol{v_s}$:

$$\boldsymbol{v_s}^T \frac{\boldsymbol{r_{obst}} - \boldsymbol{r_s}}{\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|} T_s \leq \max\left(0, \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta\right) \tag{3.2}$$

where $\boldsymbol{v_s}^T \left(\left(\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right) / \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|\right)$ represents the projection of $\boldsymbol{v_s}$ onto the normalized segment connecting $\boldsymbol{r_s}$ to $\boldsymbol{r_{obst}}$. This constraint can be further arranged as

$$\boldsymbol{v_s}^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right) T_s \leq \max\left(0, \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 - \Delta\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|\right) \tag{3.3}$$

Assume now the following parametrization of the link in terms of position and velocity of its end points

$$\boldsymbol{r_s} = \boldsymbol{r_a} + s\left(\boldsymbol{r_b} - \boldsymbol{r_a}\right) \quad \boldsymbol{v_s} = \boldsymbol{r_b} + s\left(\boldsymbol{v_s} - \boldsymbol{v_a}\right) \tag{3.4}$$

In order to enforce the safety constraints,the inequality in (3.3) is required to be satisfied for all $s \in [0, 1]$. The left hand side becomes

$$\begin{aligned} \boldsymbol{v_s}^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_s}\right) = \boldsymbol{v_a}^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right) + s\left(\boldsymbol{v_b} - \boldsymbol{v_a}\right)^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right) \\ - s\boldsymbol{v_a}^T \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right) - s^2 \underbrace{\left(\boldsymbol{v_b} - \boldsymbol{v_a}\right)^T \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right)}_{=0} \end{aligned} \tag{3.5}$$

Moreover, notice that for the right hand side

$$\left[\max\left(0, \ \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta\right)\right]^2 \leq \max\left(0, \ \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 - \Delta\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|\right) \tag{3.6}$$

As a result, the set of inequalities describing the safety constraints can be written as follows

$$\alpha + \beta s \leq g\left(s\right), \forall s \in [0, 1] \tag{3.7}$$

where

$$\begin{aligned} \alpha &= T_s \boldsymbol{v_a}^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right) \\ \beta &= T_s \left(\boldsymbol{v_b} - \boldsymbol{v_a}\right)^T \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right) - T_s \boldsymbol{v_a}^T \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right) \\ g\left(s\right) &= \left[max\left(0, \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta\right)\right]^2 \end{aligned} \tag{3.8}$$

Since the left hand side is a linear function in s, it is possible to write the following sufficient condition for the safety constraint (3.7) to be satisfied

$$\max\left\{\alpha, \alpha + \beta\right\} \leq \min_s g\left(s\right) \tag{3.9}$$

In turn, within the right hand side term, it is possible to exchange the min and

max operators, obtaining

$$\min_s g(s) = \left[\max\left(0,\ \min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta\right)\right]^2 \tag{3.10}$$

where the term $\min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\| - \Delta$ represents, when it is positive, the distance between a sphere of radius Δ centred in $\boldsymbol{r_{obst}}$ and the segment from $\boldsymbol{r_a}$ to $\boldsymbol{r_b}$. Finally, the following pair of inequalities is obtained

$$\begin{aligned} \alpha &= T_s \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{v_a} \leq \min_s g(s) \\ \alpha + \beta &= T_s \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{v_b} - T_s \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right)^T \boldsymbol{v_a} \leq \min_s g(s) \end{aligned} \tag{3.11}$$

Summarizing, the minimum separation distance criterion can be written in matrix form as:

$$T_s \boldsymbol{E} \dot{\boldsymbol{q}} \leq \boldsymbol{f} \tag{3.12}$$

where

$$\begin{aligned} \boldsymbol{E} &= \begin{bmatrix} \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{J_a} \\ \left(\boldsymbol{r_{obst}} - \boldsymbol{r_a}\right)^T \boldsymbol{J_b} - \left(\boldsymbol{r_b} - \boldsymbol{r_a}\right)^T \boldsymbol{J_a} \end{bmatrix} \\ \boldsymbol{f} &= \min_s g(s) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned} \tag{3.13}$$

$\boldsymbol{J_a}$ and $\boldsymbol{J_b}$ are position Jacobians of the two link end points.

3.1.2 Safety constraints for an arbitrarily-shaped convex obstacle

So far, safety constraints have been formulated by considering only point-shaped obstacles. In order to account for more complex obstacles, like for instance work-pieces, tools or human workers cooperating with the robot, the mathematical formalization of safety constraints must be extended to the case of obstacles having more complex geometry. Consider for instance a generic polytopic obstacle $\mathcal{O}$ as shown in Fig. 3.3.

The constraints to be enforced for such an obstacle can be written as follows

$$T_s \boldsymbol{E}\left(\boldsymbol{r_{obst}}\right) \dot{\boldsymbol{q}} \leq \boldsymbol{f}\left(\boldsymbol{r_{obst}}\right), \forall \boldsymbol{r_{obst}} \in \mathcal{O} \tag{3.14}$$

The number of constraints to be enforced at run time is conceptually infinite, i.e. one per each point belonging to $\mathcal{O}$. However, some geometrical properties of the

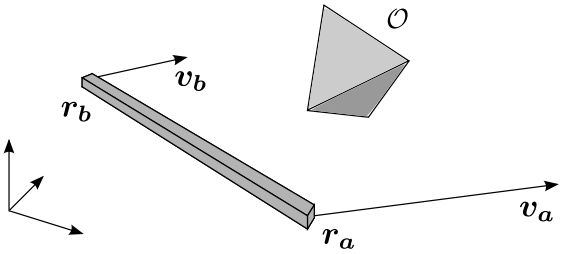


Figure 3.3: A generic polytopic (convex) obstacle.

obstacle can be exploited in order to make the problem tractable.

A sufficient condition for (3.14) to be satisfied for all points $\boldsymbol{r_{obst}} \in \mathcal{O}$ is

$$T_s \boldsymbol{E}\left(\boldsymbol{r_{obst}}\right) \dot{\boldsymbol{q}} \leq d \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \forall \boldsymbol{r_{obst}} \in \mathcal{O} \tag{3.15}$$

where the right hand side term

$$d = \min_{\boldsymbol{r_{obst}} \in \mathcal{O}} \left\| \boldsymbol{f}\left(\boldsymbol{r_{obst}}\right) \right\|_{\infty} \tag{3.16}$$

represents the minimum distance between the link of the robot and the polytopic obstacle $\mathcal{O}$ and can be easily computed using the GJK algorithm [13]. Moreover, notice that the left hand side term is linear with respect to the parameter $\boldsymbol{r_{obst}} \in \mathcal{O}$. Therefore the safety constraints regarding the pair link-obstacle can be written as follows

$$T_s \left(\boldsymbol{r_{obst}}^T \boldsymbol{E_0} + \boldsymbol{E_1} \right) \dot{\boldsymbol{q}} \leq d, \forall \boldsymbol{r_{obst}} \in \mathcal{O} \tag{3.17}$$

For linearity (and thus convexity) the aforementioned constraint (which actually still consists of an infinite number of scalar inequalities) can be equivalently written in terms of the vertices (thus a limited number) of the polytope representing the obstacle $\mathcal{O}$, hence $\forall \boldsymbol{r_{obst}} \in vert\left(\mathcal{O}\right)$.

3.2 Kinematic Model of Human Motion

In the previous section, a mathematical formulation to represent the safety constraint arising in a typical human-robot interaction scenario was presented. As already discussed, even if it is possible to consider arbitrarily geometrically shaped obstacles, their motion is not directly accounted for in the expression of the safety requirement. However, in a HRC set-up, the motion of the human, and particularly the prediction of his/her occupancy, has to be clearly taken into account to safely adjust the trajectory of the robot. In order to compute this prediction, a relatively

simple kinematic model suitable for real-time calculations is used.

3.2.1 Human Walking Kinematics

To properly describe the kinematics of human walking it is convenient to approximate the human being with a single point moving in a 2D environment. Consequently, by fixing a world-base Cartesian frame on the ground plane, the kinematic configuration of a walking human can be described as:

$$\begin{aligned} \boldsymbol{p} &= [x,\ y,\ \theta] \\ \dot{\boldsymbol{p}} &= [\dot{x},\ \dot{y},\ \omega] \end{aligned} \tag{3.18}$$

where:

- x is the coordinate with respect to the world base frame X-axis;
- y is the coordinate with respect to the world base frame Y-axis;
- θ is the angle formed between the tangent to the walking path and the world base frame X-axis;
- $\dot{x}$ is the velocity along the world base frame X-axis;
- $\dot{y}$ is the velocity along the world base frame Y-axis;
- ω is the angular velocity;

However, human locomotion can be modelled as non-holonomic, since the linear velocities $\dot{x}$ and $\dot{y}$ are actually coupled to the body orientation. Consequently, the kinematics of human walking can be described using the following unicycle-based non-holonomic model:

$$\begin{cases} \dot{x} &= v\ \cos(\theta) \\ \dot{y} &= v\ \sin(\theta) \\ \dot{\theta} &= \omega \\ \dot{v} &= a_l \\ \dot{\omega} &= a_\omega \end{cases} \tag{3.19}$$

where v is the tangential velocity directed along θ.

Finally, according to the assumption that both the linear velocity v and the angular velocity ω are piece-wise constant, the linear and angular accelerations,

a_l and a_ω, can be modelled as two independent and uncorrelated Gaussian white noises acting respectively on v and ω:

$$\begin{aligned} a_l &\sim N(0,1) \\ a_\omega &\sim N(0,1) \end{aligned} \tag{3.20}$$

In order to account for lateral walking, it is necessary to extend model (3.19) by removing the non-holomic constraint.

By introducing the orthogonal velocity term $v_\perp$, the following holonomic model is obtained:

$$\begin{cases} \dot{x} &= v\cos\theta \ - \ v_\perp \sin\theta \\ \dot{y} &= v\sin\theta \ + \ v_\perp \cos\theta \\ \dot{\theta} &= \omega \\ \dot{v} &= a_l \\ \dot{\omega} &= a_\omega \end{cases} \tag{3.21}$$

It is worth noticing that model (3.21) can be expressed in terms of a linear formulation by simply considering fully de-coupled linear velocities v_x and v_y:

$$\begin{cases} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= \omega \\ \dot{v} &= a_l \\ \dot{\omega} &= a_\omega \end{cases} \tag{3.22}$$

As a matter of fact, the new formulation is completely equivalent to the previous one, since a unique correspondence exists between the two different sets of velocities:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v \\ v_\perp \end{bmatrix}$$

$$\begin{bmatrix} v \\ v_\perp \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

To summarize, the kinematic configuration of a walking human being can still be described in terms of pose and velocity vectors $\boldsymbol{p}$ and $\dot{\boldsymbol{p}}$ (see (3.18)), but, differ-

ently from the non-holonomic case, velocities components $\dot{x}$ and $\dot{y}$ can be parameterized either in terms of forward and orthogonal components, according to (3.21), or in terms of de-coupled Cartesian components, according to (3.22).

3.2.2 Human Arm Kinematics

In order to describe the human arm kinematics, the kinematic model, originally introduced in [7] and further refined in [50], is used (see Figure 3.4). As a matter of fact, only the first 4 DOFs or the original model are considered since the human arm is roughly approximated with two segments: the first one (upper arm) connecting the shoulder to the elbow and the second one (lower arm) going from the elbow to the wrist.

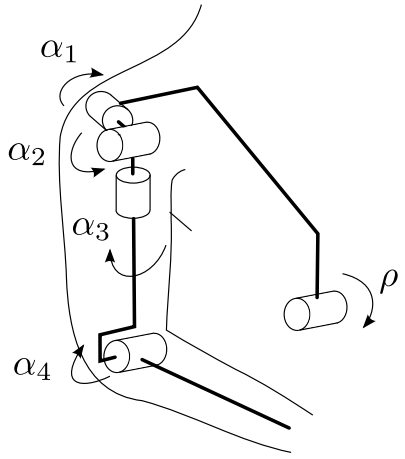


Figure 3.4: Kinematic model of the human (right) arm and torso flexion/extension angle ρ.

On the basis of this approximation, the kinematic model of the human arm can be formulated in terms of four integrators for each joint angle:

$$\begin{cases} \dfrac{d}{dt}\boldsymbol{\alpha} &= \dot{\boldsymbol{\alpha}} \\ \dfrac{d}{dt}\dot{\boldsymbol{\alpha}} &= \ddot{\boldsymbol{\alpha}} \\ \dfrac{d}{dt}\ddot{\boldsymbol{\alpha}} &= \dddot{\boldsymbol{\alpha}} \\ \dfrac{d}{dt}\dddot{\boldsymbol{\alpha}} &= \boldsymbol{\eta} \end{cases} \tag{3.23}$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix} \tag{3.24}$$

is the vector containing the joint angles represented in Figure 3.4, while, under the hypothesis that the jerk is piece-wise constant, $\boldsymbol{\eta}$ components can be modelled as independent and uncorrelated Gaussian white noises:

$$\boldsymbol{\eta} \sim N(\mathbf{0}, \boldsymbol{I}) \tag{3.25}$$

3.2.3 Complete Kinematic Model

Combining together walking and arm kinematics, a model, that describes the full motion of the upper human body, is obtained . More in depth, this new kinematic model is composed of a 3-DOF base moving on the ground plane, one lumped 1-DOF (flexion/extension) torso, a head (fixed) and two 4-DOF arms. A graphical representation of such kinematic model is given in Figure 3.5.

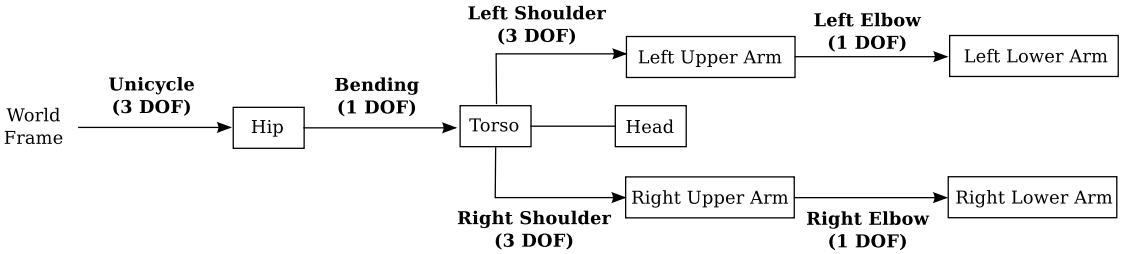


Figure 3.5: Complete human kinematic model: DOFs, frames and bodies

Given the new model, the kinematic configuration $\boldsymbol{p}$ can be extended in the following way:

$$\boldsymbol{p} = \begin{bmatrix} x & y & \theta & \rho & \boldsymbol{\alpha}^{\boldsymbol{right}} & \boldsymbol{\alpha}^{\boldsymbol{left}} \end{bmatrix}^T \tag{3.26}$$

where:

- x, y and θ corresponds to the walking kinematic configuration;
- ρ is the torso bending angle;
- $\boldsymbol{\alpha}^{\boldsymbol{right}}$ and $\boldsymbol{\alpha}^{\boldsymbol{left}}$ are respectively the joint angles vectors for the right arm and for the left arm.

Finally, since models (3.22) and (3.23) are both linear and all the status components are completely de-coupled, also the complete kinematic model can be

formalized by means of a chain of four integrators for each joint variable:

$$\begin{cases} \frac{d}{dt}\boldsymbol{p} &= \dot{\boldsymbol{p}} \\ \frac{d}{dt}\dot{\boldsymbol{p}} &= \ddot{\boldsymbol{p}} \\ \frac{d}{dt}\ddot{\boldsymbol{p}} &= \dddot{\boldsymbol{p}} \\ \frac{d}{dt}\dddot{\boldsymbol{p}} &= \boldsymbol{\eta} \end{cases} \tag{3.27}$$

It is worth noticing that an analogous representation of this kinematic model can be given in terms of a set of 3D points composing a rough scheme of the human skeleton, see Figure 3.6. Since only the motion of the upper part of the human body is considered, the points of interest are: Thorax ($\boldsymbol{T}$), Neck ($\boldsymbol{N}$), Head ($\boldsymbol{H}$), Left Shoulder ($\boldsymbol{LS}$), Right Shoulder ($\boldsymbol{RS}$), Left Elbow ($\boldsymbol{LE}$), Right Elbow ($\boldsymbol{RE}$), Left Wrist ($\boldsymbol{LW}$), and Right Wrist ($\boldsymbol{RW}$).

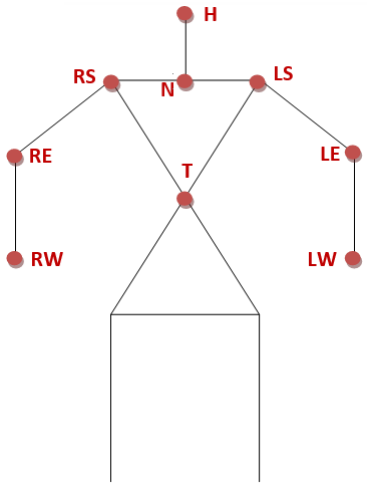


Figure 3.6: Approximated Human Skeleton.

This skeletal representation is particularly useful because several well-known algorithms (for instance [3]) can be used to extract skeletal points from a depth map acquired by either a depth sensor or an RGB-D camera. Consequently, it is convenient to consider the skeletal representation as the output of model (3.27) and to formulate the corresponding forward kinematic calculations.

On the other hand, in order to convert a generic skeletal representation into the corresponding kinematic configuration $\boldsymbol{p}$, it is necessary to formalize also inverse kinematic calculations (for the formulation of the forward and inverse kinematic calculations see [36]).

3.3 Constrained Linear Kalman Filter

In order to gain an estimation of the kinematic state of the human worker from measurements obtained by one or multiple depth sensors, a Linear Kalman Filter [17] is utilized. Depth sensors provide the position (in Cartesian coordinates) of skeletal points highlighted in Figure 3.6, then kinematic inversion is computed to obtain configuration (3.26). Furthermore, the kinematic state estimation provided by the filter is modified so as to satisfy the following bounds:

- $\boldsymbol{p}^{inf}$ and $\boldsymbol{p}^{sup}$: lower and upper bounds on human joint positions;
- $\dot{\boldsymbol{p}}^{inf}$ and $\dot{\boldsymbol{p}}^{sup}$: lower and upper bounds on human joint velocities;
- $\ddot{\boldsymbol{p}}^{inf}$ and $\ddot{\boldsymbol{p}}^{sup}$: lower and upper bounds on human joint accelerations;

3.3.1 Linear Kalman Filter implementation

The adopted process model consists in a discretized version of model (3.27) and it is composed by a chain of three discrete-time integrators for each joint variable. Moreover, the filter state vector $\boldsymbol{s}$ (that already contains joint positions, velocities, accelerations and jerks) is further extended to take into account also the parameters of the human kinematic model that are specific for each individual (i.e. the distances between the skeletal points acquired by the depth sensors). These parameters, labelled as $\boldsymbol{\pi}$ in equation (3.29), can be estimated by simply imposing a constant dynamics. In detail:

$$\boldsymbol{s}_{k+1} = F\boldsymbol{s}_k + \boldsymbol{\eta}_k \tag{3.28}$$

$$\boldsymbol{s}_k = \begin{bmatrix} \boldsymbol{p}_k \\ \dot{\boldsymbol{p}}_k \\ \ddot{\boldsymbol{p}}_k \\ \dddot{\boldsymbol{p}}_k \\ \boldsymbol{\pi}_k \end{bmatrix} \qquad F = \begin{bmatrix} I & \Delta t I & \frac{\Delta t^2}{2} I & \frac{\Delta t^3}{6} I & 0 \\ 0 & I & \Delta t I & \frac{\Delta t^2}{2} I & 0 \\ 0 & 0 & I & \Delta t I & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \tag{3.29}$$

In particular:

$$\boldsymbol{\eta}_k \sim \mathcal{N}(0,\, G) \tag{3.30}$$

models the process noise, whose covariance matrix G can be parametrized as follows. For each block of the state vector the corresponding first truncated element of the Taylor approximations contained in matrix F is considered (see equation

(3.29)), and one obtains:

$$G = \begin{bmatrix} \sigma_{\boldsymbol{p}}^2 \frac{\Delta t^4}{24} I & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{\boldsymbol{p}}}^2 \frac{\Delta t^3}{6} I & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\ddot{\boldsymbol{p}}}^2 \frac{\Delta t^2}{2} I & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dddot{\boldsymbol{p}}}^2 \Delta t I & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.31}$$

where standard deviations $\sigma_{\boldsymbol{p}}$, $\sigma_{\dot{\boldsymbol{p}}}$, $\sigma_{\ddot{\boldsymbol{p}}}$ and $\sigma_{\dddot{\boldsymbol{p}}}$ are tunable parameters.

Regarding the observation model, since the inverse kinematics of the human motion model can be computed in closed-form, all the joint positions and all the kinematic parameters corresponding to the data acquired from the n available depth sensors are considered as observed output. Consequently, the following linear transformation is considered as observation model :

$$\boldsymbol{z}_k = H \boldsymbol{s}_k + \boldsymbol{\zeta}_k \tag{3.32}$$

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{p}_{1,k} \\ \vdots \\ \boldsymbol{p}_{n,k} \\ \boldsymbol{\pi}_{1,k} \\ \vdots \\ \boldsymbol{\pi}_{n,k} \end{bmatrix} \qquad H = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \tag{3.33}$$

where $\boldsymbol{p}_{i,k}$ and $\boldsymbol{\pi}_{i,k}$ represent the set of joint positions and the set of parameters, respectively, computed via inverse kinematics on the basis of the skeletal points acquired by the i-th sensor.

Moreover, $\boldsymbol{\zeta}_k$ models the measurement noise, whose covariance matrix R is given by:

$$\boldsymbol{\zeta}_k \sim \mathcal{N}(0,\, R_k)\, R_k = \sigma_{\boldsymbol{z}}^2 I \tag{3.34}$$

where standard deviation $\sigma_{\boldsymbol{z}}$ can be determined on the basis of the accuracy of the skeletal points acquired by the depth sensors.

A relevant issue in the design of this sensor fusion strategy is the lack of synchronization between the acquisition process and the process implementing the LKF. As a matter of fact, in order to ensure accurate estimation it is needed to run the LKF at a frequency that usually is much greater than the data acquisition frequency. Consequently the filter will execute several iterations on the basis of

the same set of measurements.

A possible solution consists in updating the observation covariance matrix R_k according to the presence (or not) of new measurements, in such a way that each element on the main diagonal of R_k follows a saw tooth shape. More specifically, every time the LKF receives a new set of measurements, the corresponding blocks of matrix R_k are reset to $\sigma_z^2 I$. On the other hand, whenever the same set of measurements is re-used, the corresponding blocks in R_k are updated by adding $\sigma_z^2 I$. In this way it is obtained an uncertainty that is reset to a starting value whenever a new measurement arrives and that grows linearly in time between two consecutive measurements.

Clearly, the described LKF produces an estimation of both the kinematic state of the human worker and of his/her kinematic parameters:

$$\hat{\boldsymbol{s}}_k = \left[\begin{array}{ccccc} \hat{\boldsymbol{p}}_k^T & \dot{\hat{\boldsymbol{p}}}_k^T & \ddot{\hat{\boldsymbol{p}}}_k^T & \dddot{\hat{\boldsymbol{p}}}_k^T & \hat{\boldsymbol{\pi}}_k^T \end{array} \right]^T \tag{3.35}$$

More specifically, in case at time step k there are no valid measurements available, the filter directly outputs the a-priori estimates:

$$\hat{\boldsymbol{s}}_{k|k-1} = F\,\hat{\boldsymbol{s}}_{k-1} \tag{3.36}$$

$$P_{k|k-1} = F\,P_{k-1}\,F^T + G \tag{3.37}$$

$$\hat{\boldsymbol{s}}_k \leftarrow \hat{\boldsymbol{s}}_{k|k-1} \tag{3.38}$$

$$P_k \leftarrow P_{k|k-1} \tag{3.39}$$

while if there is at least one valid measurement, the filter also executes the prediction update and outputs the a-posteriori state estimation:

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - H\,\hat{\boldsymbol{s}}_{k|k-1} \tag{3.40}$$

$$K_k = P_{k|k-1}\,H^T\left(H\,P_{k|k-1}\,H^T + R_k\right)^{-1} \tag{3.41}$$

$$\hat{\boldsymbol{s}}_{k|k} = \hat{\boldsymbol{s}}_{k|k-1} + K_k\tilde{\boldsymbol{y}}_k \tag{3.42}$$

$$P_{k|k} = (I - HK_k)\,P_{k|k-1} \tag{3.43}$$

$$\hat{\boldsymbol{s}}_k \leftarrow \hat{\boldsymbol{s}}_{k|k} \tag{3.44}$$

$$P_k \leftarrow P_{k|k} \tag{3.45}$$

Unfortunately, data acquired from depth sensors can suffer from non-valid measurements due to various reasons (occlusions, human workers being too near with respect to the sensor, human workers leaving the sensor field of view, etc.). Since

the acquisition process is able to detect these situations, each set of measurements $\boldsymbol{p}_{i,k}$ and $\boldsymbol{\pi}_{i,k}$ is accompanied by a boolean variable, named $valid_i$, that is true if the measurements are valid, and false otherwise.

Thanks to this boolean flag, during the calculation of the innovation signal $\widetilde{\boldsymbol{y}}_k$ in equation (3.40), we set to zero all the innovation components corresponding to a non-valid set of measurements:

$$\forall i \in [1, n]\,,\ valid_i = False \implies \widetilde{\boldsymbol{y}}_{i,k} \leftarrow 0 \tag{3.46}$$

In this way non-valid measurements are prevented from affecting the a-posteriori state estimate $\hat{\boldsymbol{s}}_{k|k}$, thus ensuring that it is determined only by valid data. On the other hand both stability and correctness property of the LKF are not invalidated by this calculation since it is equivalent to consider a time-varying observation matrix H_k (see equation (3.33)), whose blocks switch between the identity and the null matrix.

3.3.2 Bounded kinematic state estimation

Since the estimation $\hat{\boldsymbol{s}}_k$, computed by the LKF, does not necessarily satisfy the chosen bounds on joint positions, velocities and accelerations, a hierarchy of optimization problems is introduced.

At first, the set of positions, that are the nearest to LKF estimation and that satisfy position bounds, is founded by solving the following quadratic programming (QP) problem:

$$\min_{\widetilde{\boldsymbol{p}}_k} \|\widetilde{\boldsymbol{p}}_k - \hat{\boldsymbol{p}}_k\|^2 \tag{3.47a}$$

$$\boldsymbol{p}^{inf} \leq \widetilde{\boldsymbol{p}}_k \leq \boldsymbol{p}^{sup} \tag{3.47b}$$

and the estimated joint positions are updated as follows:

$$\hat{\boldsymbol{p}}_k \leftarrow \widetilde{\boldsymbol{p}}_k$$

Then, a second QP problem is introduced, to find the nearest set of velocities with respect to the LKF estimation, that are also inside velocity bounds:

$$\min_{\dot{\widetilde{\boldsymbol{p}}}_k} \left\|\dot{\widetilde{\boldsymbol{p}}}_k - \dot{\hat{\boldsymbol{p}}}_k\right\|^2 \tag{3.48a}$$

$$\dot{\boldsymbol{p}}^{inf} \leq \dot{\boldsymbol{p}}_k \leq \dot{\tilde{\boldsymbol{p}}}^{sup} \tag{3.48b}$$

where bounds $\dot{\tilde{\boldsymbol{p}}}^{inf}$ and $\dot{\tilde{\boldsymbol{p}}}^{sup}$ are computed as

$$\dot{\tilde{p}}_{i,k}^{inf} = max\left(\dot{p}_{i,k}^{inf},\ \left(p_{i,k}^{inf} - \hat{p}_{i,k}\right)/\Delta t\right)$$
$$\dot{\tilde{p}}_{i,k}^{sup} = min\left(\dot{p}_{i,k}^{sup},\ \left(p_{i,k}^{sup} - \hat{p}_{i,k}\right)/\Delta t\right)$$

in order to satisfy velocity bounds and to ensure that the a-priori position estimate at time step $k+1$ stays inside position bounds. Once again, the estimated joint velocities are updated :

$$\dot{\hat{\boldsymbol{p}}}_k \leftarrow \dot{\tilde{\boldsymbol{p}}}_k$$

Finally, the same procedure can be applied to the estimated joint accelerations, thus obtaining a kinematic state of the human worker that is completely coherent with respect to position, velocity and acceleration bounds and that, at the same time, is close to the estimation computed by the LKF(for a detelaid description of the algorithm, see [43, 44]).

3.4 Human Motion Prediction

To guarantee human workers' safety in a HRC set-up, the motion of the human worker has to be clearly taken into account to safely adjust the trajectory of the robot. Moreover, beside predicting the final destination of the human worker moving inside the robotic cell, the control system should also be able to predict the volume occupied by the human, at least when he/she is actually sharing the workspace with the manipulator during cooperative tasks.

Given the human kinematic model (3.27), it is possible to develop a simple algorithm that predicts the space occupied by a human worker within a pre-defined time span (for example the time needed by the robot to stop). This prediction consists in a series of swept volumes, i.e. convex polytopes that represent the entire volume that the human can reach, given a specific kinematic configuration and a chosen time horizon.

To compute the swept volumes, first a reachable set is determined for each DOF composing the kinematic model and then the swept volumes are obtained by superimposing the motion of each joint, according to the reachable set previously determined.

3.4.1 Human Kinematics Reachable Set

The reachable set of each DOF of the human kinematic model is computed on the basis of joint positions and velocities and by considering fixed bounds on joint accelerations.

While the walking parameters x, y and θ are unbounded, the Human Arm DOFs have bounds, due to some intrinsic limitations in the gleno-humeral joint (shoulder) as well as in the elbow. Differently from robots, however, these limits are coupled. In particular, the range of motion of the human arm is limited to the region identified by the following constraints:

$$-9^\circ \leq \alpha_1 \leq 160^\circ \tag{3.49a}$$

$$-43^\circ + \frac{\alpha_1}{3} \leq \alpha_2 \leq 153^\circ - \frac{\alpha_1}{6} \tag{3.49b}$$

$$-90^\circ + \frac{7\alpha_1}{9} - \frac{\alpha_2}{9} + \frac{2\alpha_1\alpha_2}{810} \leq \alpha_3 \leq 60^\circ + \frac{4\alpha_1}{9} - \frac{5\alpha_2}{9} + \frac{5\alpha_1\alpha_2}{810} \tag{3.49c}$$

$$20^\circ \leq \alpha_4 \leq 180^\circ \tag{3.49d}$$

$$-30^\circ \leq \rho \leq 90^\circ \tag{3.49e}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the arm joint angles expressed in degrees and ρ is the torso bending angle also expressed in degrees.

For a generic DOF q, the reachable set is computed as in algorithm 1.

Algorithm 1 Human DOF T_s-reachable set $\boldsymbol{q_{T_s}}$

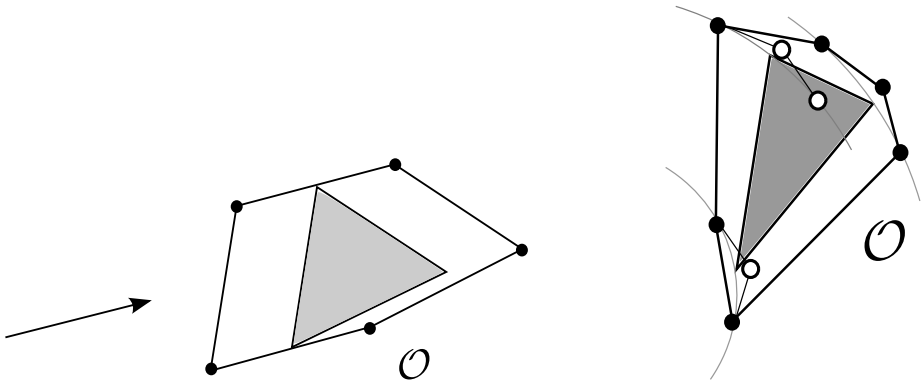
if $\dot{q} + \ddot{q}^{sup} T_s \leq \dot{q}^{sup}$ **then**
  $q^+_{T_s} = q + \dot{q} T_s + \frac{1}{2} \ddot{q}^{sup} T_s^2$
else
  $T^* = \left(\dot{q}^{sup} - \dot{q}\right) / \ddot{q}^{sup}$
  $q^+_{T_s} = q + \dot{q} T^* + \frac{1}{2} \ddot{q}^{sup} \left(T^*\right)^2 + \dot{q}^{sup} \left(T_s - T^*\right)$
end if
if $\dot{q} + \ddot{q}^{inf} T_s \geq \dot{q}^{inf}$ **then**
  $q^-_{T_s} = q + \dot{q} T_s + \frac{1}{2} \ddot{q}^{inf} T_s^2$
else
  $T^* = \left(\dot{q}^{inf} - \dot{q}\right) / \ddot{q}^{inf}$
  $q^-_{T_s} = q + \dot{q} T^* + \frac{1}{2} \ddot{q}^{inf} \left(T^*\right)^2 + \dot{q}^{inf} \left(T_s - T^*\right)$
end if
$q^+_{T_s} = max\left(q^+_{T_s}, q\right)$
$q^+_{T_s} = max\left(q^{inf}, min\left(q^+_{T_s}, q^{sup}\right)\right)$
$q^-_{T_s} = min\left(q^-_{T_s}, q\right)$
$q^-_{T_s} = max\left(q^{inf}, min\left(q^-_{T_s}, q^{sup}\right)\right)$

Notice that, when velocity saturation has to be applied, acceleration is set equal to its maximum/minimum value for the first T^* (time to reach the velocity bound) and for the last $T_s - T^*$ is set equal to zero, so the hypothesis of piecewise constant acceleration holds.

3.4.2 Human Swept Volumes Calculation

Given a generic 3-dimensional convex object $\mathcal{O}$, there are two different swept volumes that can be computed: a translational swept volume and a rotational one, depending on the kind of motion $\mathcal{O}$ is subjected to. If a prismatic joint is considered, the translational swept volume of $\mathcal{O}$ can be computed by applying the corresponding translation to each point belonging to $\mathcal{O}$ and by determining the convex hull of the resulting points, as shown in Figure 3.7a.

In turn, for a rotational DOF, one of the methods described in [45] is exploited. In particular, when applying a rotation to each point belonging to $\mathcal{O}$, a circular arc is obtained. A possible way to approximate this arc with a finite number of points is to construct a triangle as it is shown in Figure 3.8. Therefore, the rotational swept volume of $\mathcal{O}$ can be easily obtained as the convex hull of the vertices of all the triangles, see Figure 3.7b.



(a) (b)

Figure 3.7: Figure 3.7a: translational swept volume of convex object. Figure 3.7b: rotational swept volume of convex object. The corresponding convex hulls are also highlighted.

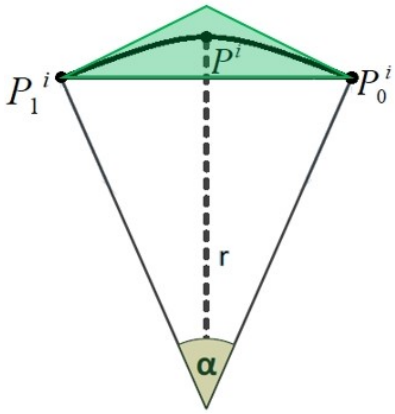


Figure 3.8: Triangular approximation of a circular arc.

Knowing how to determine the translational and rotational swept volumes for a generic set of points and given the reachable set for each DOF of the human model, the prediction of human occupancy can be determined by computing a specific swept volume for each limb:

- head - HD;
- thorax - THX;
- upper left arm (from shoulder to elbow) - ULA;
- lower left arm (from elbow to wrist) - LLA;
- upper right arm - URA;
- lower right arm - LRA.

Assume that a convex object $\mathcal{V}_l$ representing the l-th limb is given by means of its vertices set. Then, referring to the human kinematic model (3.27), a list of DOFs from the current limb to the world frame can be arranged.

Algorithm 2 Swept Volume Calculation

$\mathcal{SV} \leftarrow \emptyset$;
$L = \{HD, THX, ULA, LLA, URA, LRA\}$;
for all $l \in L$ **do**
  $\mathcal{P} = kinematicChain\,(l, \boldsymbol{p})$;
  $\mathcal{V}_l \leftarrow InitSweptVolume\,()$;
  if $isPrismatic\,(p_1)$ **then**
    $\mathcal{V}_l \leftarrow sweepLin\left(\mathcal{V}_l,\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$;
  else
    $\mathcal{V}_l \leftarrow sweepRot\left(\mathcal{V}_l,\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$;
  end if
  for all $p_i \in \mathcal{P}$ **do**
    if $p_i \neq p_1$ **then**
      $\mathcal{V}_l \leftarrow A^i_{i-1} \cdot \mathcal{V}_l = \left\{A^i_{i-1} \cdot \boldsymbol{r}^j_l \mid \boldsymbol{r}^j_l \in \mathcal{V}_l\right\}$;
      if $isPrismatic\,(p_i)$ **then**
        $\mathcal{V}_l \leftarrow sweepLin\left(\mathcal{V}_l,\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$;
      else
        $\mathcal{V}_l \leftarrow sweepRot\left(\mathcal{V}_l,\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$;
      end if
    end if
  end for
  $\mathcal{SV} \leftarrow \mathcal{SV} \cup \{\mathcal{V}_l\}$;
end for

Consequently, the swept volume of the l-th limb can be determined by iteratively applying the proper sweeping strategy to $\mathcal{V}_l$ for each DOF connecting the l-th limb to the world-base frame. Each sweeping operation is calculated on the basis of the upper and lower bounds $\boldsymbol{p}^-_{\boldsymbol{T_s}}$ and $\boldsymbol{p}^+_{\boldsymbol{T_s}}$ previously computed. A pseudo-code version of this procedure is sketched in Algorithm 2, where:

- $kinematicChain\,(l, \boldsymbol{p})$ is a function that computes $\mathcal{P}$, i.e. the set of DOFs connecting the limb to the world-base frame. The set is ordered starting from the limb and going backwards along the kinematic chain until the world-base frame is reached;
- $InitSweptVolume\,()$ is a function that initializes a swept volume by including into the set the endpoints of the link corresponding to the first DOF;
- $isPrismatic\,(p_i)$ is a logical predicate that is true if its argument is a prismatic DOF and false otherwise;

- $p^-_{T_s,i}$ and $p^+_{T_s,i}$ are, respectively, the lower and upper bound of the i-th DOF selected by *kinematicChain*;

- $sweepLin\left(\mathcal{V},\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$ computes the translational swept volume of the set of points $\mathcal{V}$, given the DOF bounds;

- $sweepRot\left(\mathcal{V},\, p^-_{T_s,i},\, p^+_{T_s,i}\right)$ computes the rotational swept volume of the set of points $\mathcal{V}$, given the DOF bounds;

- A^i_{i-1} is the linear transformation matrix from frame i to frame $i-1$, with respect to the output of *kinematicChain*;

- $\boldsymbol{r}^j_l$ is the j-th vertex of the l-th swept volume $\mathcal{V}_l$.

Finally, in order to account for the dimensions of the different human body parts, a radius parameter r is introduced and each convex swept volume $\mathcal{V}$ is augmented by computing the Minkowski sum $\oplus$ of its convex hull and a sphere of radius r:

$$\forall \mathcal{V}_l \in \mathcal{SV} \implies \mathcal{V}^r_l = convhull\left(\mathcal{V}_l\right) \oplus \left\{ b \in \mathbb{R}^3 :\ ||b|| \leq r \right\} \tag{3.50}$$

It is worth mentioning that the augmented swept volumes $\mathcal{V}^r_l$ are used only for visualization purposes. As a matter of fact, according to the formulation of the safety constraints given in Section 3.1, it is possible to incorporate the radius r inside the clearance parameter d of equation (3.17) and to state the safety constraints separately for each swept volume $\mathcal{V}_l$.

3.5 Kinematic Scaling Algorithm

In order to solve the safety-aware trajectory motion planning problem, a kinematic scaling algorithm is proposed, whose block scheme is sketched in Figure 3.9. The algorithm suitably scales a pre-planned trajectory in time in order to guarantee that the robot completely stops before colliding with a generic obstacle $\mathcal{O}$.

Assume the following well-known parametrization of the task with respect to time:

$$\boldsymbol{x}\left(\tau\right) \quad \boldsymbol{x}'\left(\tau\right) = \frac{\partial \boldsymbol{x}}{\partial \tau} \tag{3.51}$$

where τ is the time variable and $\boldsymbol{x}\left(\cdot\right)$ is a differentiable task function specifying the desired trajectory. Given a specific value of τ is it possible to evaluate the motion

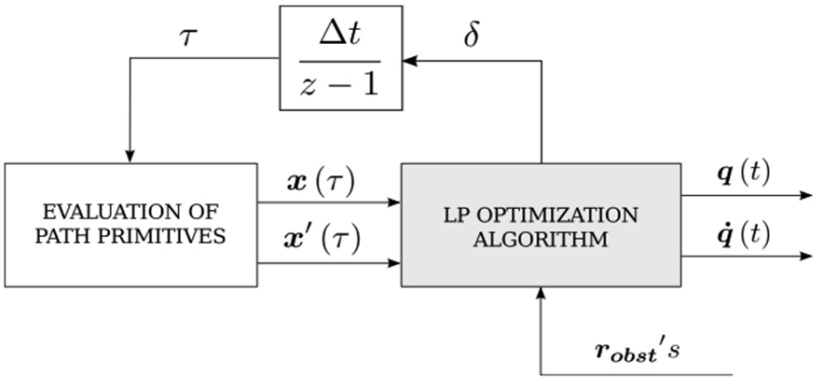


Figure 3.9: Kinematic scaling algorithm block scheme.

primitives, i.e. to determine the values of both $\boldsymbol{x}(\tau)$ and $\boldsymbol{x}'(\tau)$. Let

$$\delta \in [0, 1] \tag{3.52}$$

be a scalar quantity adopted to kinematically scale the trajectory in time. More in depth:

- $\delta = 1 \implies$ nominal trajectory, i.e. path executed at programmed speed;
- $\delta = 0 \implies$ the robot stops;

then, the following Linear Programming (LP) optimization problem is introduced:

$$\max_{\delta, \dot{\boldsymbol{q}}} \delta \tag{3.53a}$$

$$T_s \boldsymbol{E}(\boldsymbol{r_{obst}}) \dot{\boldsymbol{q}} \delta \leq d \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \forall \boldsymbol{r_{obst}} \in \mathcal{O} \tag{3.53b}$$

$$\boldsymbol{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} = \delta \boldsymbol{x}' \tag{3.53c}$$

$$0 \leq \delta \leq 1 \tag{3.53d}$$

where $\boldsymbol{J}(\boldsymbol{q})$ is the manipulator Jacobian matrix, T_s is the manipulator stopping time and the equality constraint (3.53c) guarantees that the robot does not deviate from the pre-programmed path. Moreover, notice that each pair link-obstacle is accounted for in the inequalities (3.53b), that correspond to the safety constraints already introduced in Section 3.1. It is worth noting that, since the prediction of human occupancy introduced in the previous Section 3.4 consists in a set of convex volumes, each swept volume can be treated as a single convex obstacle when formulating the safety constraints.

As a matter of fact, the solution of the LP problem tends to maximize the throughput of the robot, while being consistent with safety requirements, thus resulting in a trade-off between safety and productivity. Notice that the problem always has a trivial solution:

$$\delta = 0$$

which guarantees its solvability in realistic applications.

In order to avoid chattering behaviour of variable δ that would result in multiple activations and suspensions of the task, a hysteresis has been implemented: once δ is set to zero, the output of the trajectory scaling algorithm is forced to zero, until the minimum distance exceeds a predefined threshold. Finally, the computed value of δ is used to perform the update of the time variable τ:

$$\tau_{k+1} = \tau_k + \Delta t \cdot \delta \tag{3.54}$$

where Δt is the control algorithm cycle-time.

Chapter 4

Modifications of the Human Kinematics Reachable Set

One of the main steps to generate the swept volumes of human motion is the computation of the maximum and minimum position, that each DOF of the Human Model can reach in a certain amount of time.

In algorithm 1, presented in Section 3.4, it is assumed that the human operator moves always at maximum or minimum acceleration. But during an everyday activity or a repetitive task, human movements are rarely at maximum/minimum acceleration (see [41]). So the computed reachable set bounds will be bigger than the ones computed using the "real" acceleration of the DOFs, bringing to the generation of more conservative swept volumes, strengthening safety at the expense of productivity (see Fig. 3.1).

In the following sections, two modifications of algorithm 1 are presented: in the former, bounded accelerations are replaced with the ones estimated by the Kalman Filter. The latter is a further enrichment of the first one: reachable sets are computed on the basis of joint positions, velocities and accelerations and by considering fixed bounds on joint jerks.

4.1 Insertion of Estimated Accelerations

The Constrained Linear Kalman Filter, presented in Section 3.3, gives an estimate of each DOF's acceleration (see the vector of the state of (3.29)). Furthermore, the LKF provides, if estimated accelerations are not coherent with their bounds, the nearest set of accelerations, with respect to the estimation, that are also inside bounds. So, to compute the reachable set of each DOF of the Human Kinematics Model, the estimate of accelerations is used, removing the assumption of motions

always at maximum/minimum acceleration.

For a generic DOF q, the reachable set is computed as in algorithm 3.

Algorithm 3 Human DOF T_s-reachable set $\boldsymbol{q_{T_s}}$

if $\dot{q} + \ddot{q}T_s \leq \dot{q}^{sup} \wedge \dot{q} + \ddot{q}T_s \geq \dot{q}^{inf}$ **then**
  $q_{est} = q + \dot{q}T_s + \frac{1}{2}\ddot{q}T_s^2$
else
  if $\dot{q} + \ddot{q}T_s > \dot{q}^{sup}$ **then**
    $T^* = \left(\dot{q}^{sup} - \dot{q}\right)/\ddot{q}$
    $q_{est} = q + \dot{q}T^* + \frac{1}{2}\ddot{q}\left(T^*\right)^2 + \dot{q}^{sup}\left(T_s - T^*\right)$
  end if
  if $\dot{q} + \ddot{q}T_s < \dot{q}^{inf}$ **then**
    $T^* = \left(\dot{q}^{inf} - \dot{q}\right)/\ddot{q}$
    $q_{est} = q + \dot{q}T^* + \frac{1}{2}\ddot{q}\left(T^*\right)^2 + \dot{q}^{inf}\left(T_s - T^*\right)$
  end if
end if
if $q_{est} \geq q$ **then**
  $q^+_{T_s} = q_{est}$
  $q^-_{T_s} = q$
else
  $q^+_{T_s} = q$
  $q^-_{T_s} = q_{est}$
end if
$q^+_{T_s} = max\left(q^{inf}, min\left(q^+_{T_s}, q^{sup}\right)\right)$
$q^-_{T_s} = max\left(q^{inf}, min\left(q^-_{T_s}, q^{sup}\right)\right)$

Notice that, compared to algorithm 1, there is only one equation to compute the reachable set bounds, instead of two (one using the acceleration upper bound value, the other using the lower). Once that the estimate of the T_s-reachable position q_{est} is obtained, the set bounds are given, checking which one is bigger between q_{est} and the actual position q.

When velocity saturation has to be applied, similarly to the original algorithm, acceleration is set equal to its estimated value for the first T^* (time to reach the velocity bound) and for the last $T_s - T^*$ is set equal to zero, so the hypothesis of piecewise constant acceleration still holds.

4.2 Reachable Set Computation using Bounded Jerk

To model the kinematics of the human arm, jerk is assumed to be piecewise constant (see (3.23)). But when reachable sets are calculated, jerk is not considered; in addition, it is the acceleration that is considered piecewise constant.

So, to make the reachable set computation algorithm more consistent with the Human Kinematics Model, fixed bounds on each DOF jerk are introduced. Furthermore, acceleration is no longer considered to be piecewise constant.

The new algorithm is structured in the following way: for each DOF, first, times needed to reach the upper/lower bounds of velocity and acceleration are calculated; then, comparing them with T_s, the reachable set bounds are computed. To build the algorithm, a cue is taken from [52], with the difference that in the paper a position-velocity plane is considered, while here a velocity-acceleration plane is taken into account.

4.2.1 Computation of Times to Saturate Velocity and Acceleration

The case of upper bounds saturation is firstly introduced; for the lower bounds, the formulas are very similar, only some signs change, because it is assumed that velocity, acceleration and jerk bounds are symmetric.

Given actual velocity and acceleration, and velocity, acceleration and jerk upper bounds values, the time to reach the velocity upper bound is calculated as in algorithm 4.

Algorithm 4 Time to reach velocity upper bound

$a = 2\dddot{q}^{sup}$
$b = 4\ddot{q}$
$c = \ddot{q}^2/\dddot{q}^{sup} + 2\left(\dot{q} - \dot{q}^{sup}\right)$
$delta = b^2 - 4a * c$
$sol_1 = \left(-b + \sqrt{delta}\right)/2a$
$sol_2 = \left(-b - \sqrt{delta}\right)/2a$
if $sol_1 \geq 0$ **then**
    $T_{p1} = sol_1$
else
    $T_{p1} = sol_2$
end if
$T_{p2} = \left(\ddot{q}^{sup} - \ddot{q}\right)/\dddot{q}^{sup}$
if $T_{p1} < T_{p2}$ **then**
    $\ddot{q}_{new} = \ddot{q} + \dddot{q}^{sup} T_{p1}$
    $\dot{q}_{new} = \dot{q} + \ddot{q} T_{p1} + \frac{1}{2}\dddot{q}^{sup} T_{p1}^2$
    $T_m = \ddot{q}_{new}/\dddot{q}^{sup}$
    $T_{tot} = T_{p1} + T_m$
else
    $\ddot{q}_{new} = \ddot{q} + \dddot{q}^{sup} T_{p2}$
    $\dot{q}_{new} = \dot{q} + \ddot{q} T_{p2} + \frac{1}{2}\dddot{q}^{sup} T_{p2}^2$
    $T_o = \left(\dot{q}^{sup} - \dot{q}_{new} - \frac{1}{2}\ddot{q}_{new}^2/\dddot{q}^{sup}\right)/\ddot{q}_{new}$
    $T_m = \ddot{q}_{new}/\dddot{q}^{sup}$
    $T_{tot} = T_{p2} + T_o + T_m$
end if

The main idea is to calculate times T_{p1} and T_{p2}. The first one is the time to reach the limit curve, along which, with the jerk set to its minimum value, velocity reaches its upper bound with acceleration equal to zero (see for example the blue curve of Figure 4.1). T_{p2}, instead, is the time to reach acceleration upper bound.

If T_{p1} is smaller than T_{p2}, jerk is set for T_{p1} seconds to its maximum value and for T_m to its minimum. In the other case, jerk is set for T_{p2} seconds to its maximum value, for T_o to zero and for T_m to its minimum value.

Time to reach the velocity lower bound is calculated as in algorithm 5.

If T_{p1} is smaller than T_{p2}, jerk is set for T_{p1} seconds to its minimum value and for T_m to its maximum.

In the other case, jerk is set for T_{p2} seconds to its minimum value, for T_o to zero and for T_m to its maximum (see Figure 4.2).

Algorithm 5 Time to reach velocity lower bound

$a = 2\dddot{q}^{sup}$
$b = -4\ddot{q}$
$c = \ddot{q}^2/\dddot{q}^{sup} + 2\left(-\dot{q} - \dot{q}^{sup}\right)$
$delta = b^2 - 4a * c$
$sol_1 = \left(-b + \sqrt{delta}\right)/2a$
$sol_2 = \left(-b - \sqrt{delta}\right)/2a$
if $sol_1 \geq 0$ **then**
    $T_{p1} = sol_1$
else
    $T_{p1} = sol_2$
end if
$T_{p2} = \left(\ddot{q}^{sup} + \ddot{q}\right)/\dddot{q}^{sup}$
if $T_{p1} < T_{p2}$ **then**
    $\ddot{q}_{new} = \ddot{q} - \dddot{q}^{sup} T_{p1}$
    $\dot{q}_{new} = \dot{q} + \ddot{q} T_{p1} - \frac{1}{2}\dddot{q}^{sup} T_{p1}^2$
    $T_m = -\ddot{q}_{new}/\dddot{q}^{sup}$
    $T_{tot} = T_{p1} + T_m$
else
    $\ddot{q}_{new} = \ddot{q} - \dddot{q}^{sup} T_{p2}$
    $\dot{q}_{new} = \dot{q} + \ddot{q} T_{p2} - \frac{1}{2}\dddot{q}^{sup} T_{p2}^2$
    $T_o = \left(\dot{q}^{sup} + \dot{q}_{new} - \frac{1}{2}\ddot{q}_{new}^2/\dddot{q}^{sup}\right)/\ddot{q}_{new}$
    $T_m = -\ddot{q}_{new}/\dddot{q}^{sup}$
    $T_{tot} = T_{p2} + T_o + T_m$
end if

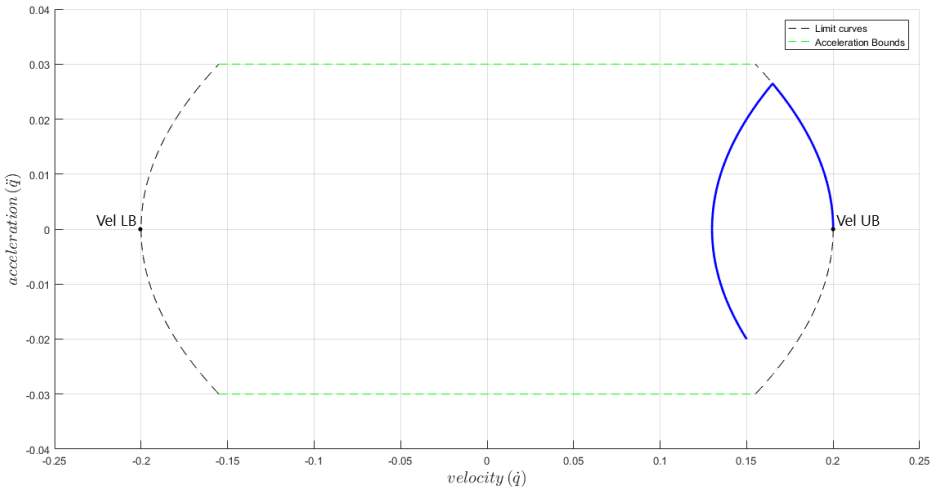


Figure 4.1: Trajectory (in blue), that reaches velocity upper bound without acceleration's saturation.

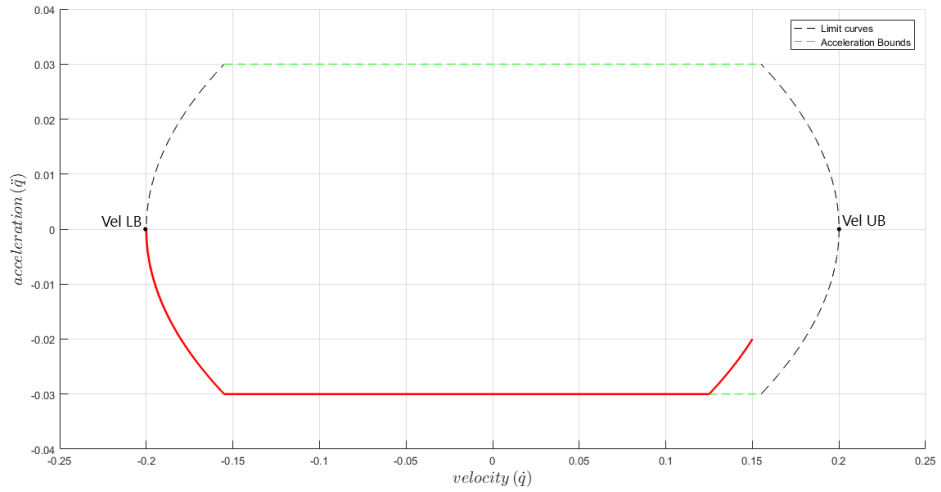


Figure 4.2: Trajectory (in red), that reaches velocity lower bound with acceleration's saturation.

4.2.2 Reachable Set Bounds Computation

Once time T_{tot} is calculated, the reachable set upper bound is computed as in algorithm 6.

To compute the lower bound, the algorithm is very similar to the previous one (see algorithm 7).

Algorithm 6 Human DOF T_s-reachable set upper bound $\boldsymbol{q}^+_{\boldsymbol{T_s}}$

if $T_{p1} < T_{p2}$ **then**
  if $T_s > T_{tot}$ **then**
    $q^+_{T_s} = q + \dot{q}T_{p1} + \frac{1}{2}\ddot{q}T^2_{p1} + \frac{1}{6}\dddot{q}^{sup}T^3_{p1} + \dot{q}_{new}T_m + \frac{1}{2}\ddot{q}_{new}T^2_m - \frac{1}{6}\dddot{q}^{sup}T^3_m + \dot{q}^{sup}\left(T_s - T_{tot}\right)$
  else if $T_s > T_{p1}$ **then**
    $q^+_{T_s} = q + \dot{q}T_{p1} + \frac{1}{2}\ddot{q}T^2_{p1} + \frac{1}{6}\dddot{q}^{sup}T^3_{p1} + \dot{q}_{new}\left(T_s - T_{p1}\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p1}\right)^2 -$
    $-\frac{1}{6}\dddot{q}^{sup}\left(T_s - T_{p1}\right)^3$
  else
    $q^+_{T_s} = q + \dot{q}T_s + \frac{1}{2}\ddot{q}T^2_s + \frac{1}{6}\dddot{q}^{sup}T^3_s$
  end if
else
  if $T_s > T_{tot}$ **then**
    $q^+_{T_s} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T^2_{p2} + \frac{1}{6}\dddot{q}^{sup}T^3_{p2} + \dot{q}_{new}T_o + \frac{1}{2}\ddot{q}_{new}T^2_o + \left(\dot{q}_{new} + \ddot{q}_{new}T_o\right)T_m +$
    $+\frac{1}{2}\ddot{q}_{new}T^2_m - \frac{1}{6}\dddot{q}^{sup}T^3_m + \dot{q}^{sup}\left(T_s - T_{tot}\right)$
  else if $T_s > T_{p2} + T_o$ **then**
    $q^+_{T_s} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T^2_{p2} + \frac{1}{6}\dddot{q}^{sup}T^3_{p2} + \dot{q}_{new}T_o + \frac{1}{2}\ddot{q}_{new}T^2_o +$
    $+\left(\dot{q}_{new} + \ddot{q}_{new}T_o\right)\left(T_s - T_{p2} - T_o\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p2} - T_o\right)^2 - \frac{1}{6}\dddot{q}^{sup}\left(T_s - T_{p2} - T_o\right)^3$
  else if $T_s > T_{p2}$ **then**
    $q^+_{T_s} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T^2_{p2} + \frac{1}{6}\dddot{q}^{sup}T^3_{p2} + \dot{q}_{new}\left(T_s - T_{p2}\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p2}\right)^2$
  else
    $q^+_{T_s} = q + \dot{q}T_s + \frac{1}{2}\ddot{q}T^2_s + \frac{1}{6}\dddot{q}^{sup}T^3_s$
  end if
end if
$q^+_{T_s} = max\left(q^+_{T_s}, q\right)$
$q^+_{T_s} = max\left(q^{inf}, min\left(q^+_{T_s}, q^{sup}\right)\right)$

Algorithm 7 Human DOF T_s-reachable set lower bound $\boldsymbol{q}_{\boldsymbol{T_s}}^{-}$

if $T_{p1} < T_{p2}$ **then**
  if $T_s > T_{tot}$ **then**
    $q_{T_s}^{-} = q + \dot{q}T_{p1} + \frac{1}{2}\ddot{q}T_{p1}^2 - \frac{1}{6}\dddot{q}^{sup}T_{p1}^3 + \dot{q}_{new}T_m + \frac{1}{2}\ddot{q}_{new}T_m^2 + \frac{1}{6}\dddot{q}^{sup}T_m^3 - \dot{q}^{sup}\left(T_s - T_{tot}\right)$
  else if $T_s > T_{p1}$ **then**
    $q_{T_s}^{-} = q + \dot{q}T_{p1} + \frac{1}{2}\ddot{q}T_{p1}^2 - \frac{1}{6}\dddot{q}^{sup}T_{p1}^3 + \dot{q}_{new}\left(T_s - T_{p1}\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p1}\right)^2 +$
    $+\frac{1}{6}\dddot{q}^{sup}\left(T_s - T_{p1}\right)^3$
  else
    $q_{T_s}^{-} = q + \dot{q}T_s + \frac{1}{2}\ddot{q}T_s^2 - \frac{1}{6}\dddot{q}^{sup}T_s^3$
  end if
else
  if $T_s > T_{tot}$ **then**
    $q_{T_s}^{-} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T_{p2}^2 - \frac{1}{6}\dddot{q}^{sup}T_{p2}^3 + \dot{q}_{new}T_o + \frac{1}{2}\ddot{q}_{new}T_o^2 + \left(\dot{q}_{new} + \ddot{q}_{new}T_o\right)T_m +$
    $+\frac{1}{2}\ddot{q}_{new}T_m^2 + \frac{1}{6}\dddot{q}^{sup}T_m^3 - \dot{q}^{sup}\left(T_s - T_{tot}\right)$
  else if $T_s > T_{p2} + T_o$ **then**
    $q_{T_s}^{-} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T_{p2}^2 - \frac{1}{6}\dddot{q}^{sup}T_{p2}^3 + \dot{q}_{new}T_o + \frac{1}{2}\ddot{q}_{new}T_o^2 +$
    $+\left(\dot{q}_{new} + \ddot{q}_{new}T_o\right)\left(T_s - T_{p2} - T_o\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p2} - T_o\right)^2 + \frac{1}{6}\dddot{q}^{sup}\left(T_s - T_{p2} - T_o\right)^3$
  else if $T_s > T_{p2}$ **then**
    $q_{T_s}^{-} = q + \dot{q}T_{p2} + \frac{1}{2}\ddot{q}T_{p2}^2 - \frac{1}{6}\dddot{q}^{sup}T_{p2}^3 + \dot{q}_{new}\left(T_s - T_{p2}\right) + \frac{1}{2}\ddot{q}_{new}\left(T_s - T_{p2}\right)^2$
  else
    $q_{T_s}^{-} = q + \dot{q}T_s + \frac{1}{2}\ddot{q}T_s^2 - \frac{1}{6}\dddot{q}^{sup}T_s^3$
  end if
end if
$q_{T_s}^{-} = min\left(q_{T_s}^{-}, q\right)$
$q_{T_s}^{-} = max\left(q^{inf}, min\left(q_{T_s}^{-}, q^{sup}\right)\right)$

Chapter 5

The Adaptive Kalman Filter

In the Linear Kalman Filter, presented in Section 3.3, the process noise covariance matrix (see 3.31) has standard deviations $\sigma_{\boldsymbol{p}}$, $\sigma_{\dot{\boldsymbol{p}}}$, $\sigma_{\ddot{\boldsymbol{p}}}$ and $\sigma_{\dddot{\boldsymbol{p}}}$, that are tunable parameters. But in all the experiments described in [36], these parameters are fixed.

In the following sections, a modification of the Kalman Filter, that permits to adapt the process and measurements noise covariance matrices, is discussed. First, an overview of different methods to obtain an Adaptive Kalman Filter (AKF), that are available in the literature, is given. Then, the chosen method of adaptation and the modifications to the LKF are presented.

5.1 Introduction to the Adaptation Methods

The adaptive Kalman filtering schemes most frequently found in the literature are Innovation-based Adaptive Estimation (IAE) and Multiple Model Adaptive Estimation (MMAE).

IAE methods estimate the covariance matrix of the process noise Q and/or the measurement noise R and utilize the fact that for the right values of Q and R the innovation sequence of the Kalman filter is white noise. By tuning Q and/or R and studying the resulting innovation sequence one can get an idea of the appropriate values of the covariance matrices. An early example of IAE is found in [26], while more recent attempts are for instance [8] and [48].

MMAE methods handle model uncertainty by implementing a bank of several different models and computing the Bayesian probability for each model to be the true system model given the measurement sequence and under the assumption that one of the models in the model bank is the correct one. The state estimate can be either the output of the most probable model or a weighted sum of the outputs of

all models. This method is suitable for applications such as fault detection where one has some a priori information on the system dynamics. Examples of MMAE algorithms are found for instance in [24] and [47].

A different method, that can be found in [18], is the Optimization-based Estimation (OAE). It produces an estimate of the process noise covariance matrix Q by solving an optimization problem over a short window of data.

The model of the KF is unique, so a MMAE method is not advisable, while for OAE one, an optimization algorithm for a big system should be implemented; this could slow down too much the entire swept volumes generation. So, the chosen method to implement the AKF is the Innovation-based one.

The main steps for the implementation of the method are taken from [26], while for the derivation of Q estimate, two different approaches are followed: one, more analytic, is derived in [6]. The other is an approximation, that is introduced in [27]. The notation for the KF parameters is the one already introduced in Section 3.3.

5.2 Innovation-based Adaptive Kalman Filter

The problem is that true values of Q and R are unknown. It is required to:

1. check whether the Kalman filter built using some estimates of Q and R is close to optimal or not (hypothesis testing).

2. obtain unbiased and consistent estimates of Q and R (statistical estimation)

3. adapt the Kalman filter at regular intervals using all the previous information (adaptive filtering).

To solve these problems, the innovation property of an optimal filter is used:

> For an optimal filter, the innovation sequence (3.40), is a Gaussian white noise sequence.

For a suboptimal KF, it could be proven that the autocorrelation function of $\widetilde{\boldsymbol{y}}_k$ does not depend on k. Therefore, $\widetilde{\boldsymbol{y}}_k$ is a stationary Gaussian random sequence (Gaussian because of linearity) and it can be defined:

$$C_j \equiv E\left[\widetilde{\boldsymbol{y}}_k \widetilde{\boldsymbol{y}}_{k-j}^T\right]$$

Then

$$C_j = HPH^T + R, \qquad j = 0 \tag{5.1}$$

$$C_j = H\left[F\left(I - KH\right)\right]^{j-1} F\left[PH^T - KC_0\right], \qquad j > 0 \tag{5.2}$$

Furthermore,

$$C_{-j} = C_j^T$$

Notice that the optimal choice of K (see (3.41)) makes C_j vanish for all $j \neq 0$ (the innovation property).

Whiteness property can be tested statistically by a method given in [16]. In this method, an estimate of C_j, denoted as $\hat{C}_j$ can be obtained, by using the ergodic property of a stationary random sequence

$$\hat{C}_j = \frac{1}{N} \sum_{k=j}^{N} \tilde{\boldsymbol{y}}_k \tilde{\boldsymbol{y}}_{k-j}^T \tag{5.3}$$

where N is the number of sample points.

Then, estimates of the normalized autocorrelation coefficients ρ_j are obtained by dividing the elements of $\hat{C}_j$ by the appropriate elements of $\hat{C}_0$, e.g.

$$[\rho_j]_{ik} = \frac{\left[\hat{C}_j\right]_{ik}}{\sqrt{\left[\hat{C}_0\right]_{ii}\left[\hat{C}_0\right]_{kk}}} \tag{5.4}$$

where $\left[\hat{C}_j\right]_{ik}$ denotes the elements of the i^{th} row and the k^{th} column of matrix $\hat{C}_j$. Here the diagonal elements of ρ_j are of particular interest for the case of white noise. In fact, the test is performed looking at a set of values for $[\rho_j]_{ii}$, $j > 0$ and checking the number of times they lie outside the band $\pm\left(1.96/\sqrt{N}\right)$. If this number is less than 5 percent of the total, the innovation sequence is white. This test is based on the assumption of large N.

If the test reveals that the filter is suboptimal, the next step is to obtain better estimates of Q and R. This can be done using $\hat{C}_j$ of equation (5.3) and proceeding with three passages:

1. Obtain an estimate of PH^T.

2. Obtain an estimate of R.

3. Obtain an estimate of Q.

An estimate of PH^T can be obtained directly from (5.2):

$$\hat{P}\hat{H}^T = K\hat{C}_0 + A^* \begin{bmatrix} \hat{C}_1 \\ \vdots \\ \hat{C}_n \end{bmatrix} \tag{5.5}$$

where

$$A = \begin{bmatrix} HF \\ HF\left(I - KH\right)F \\ \vdots \\ H\left[F\left(I - KH\right)\right]^{n-1}F \end{bmatrix} \tag{5.6}$$

and A^* is the pseudo-inverse of A.

Then, an estimate of R can be obtained from (5.1):

$$\hat{R} = \hat{C}_0 - H\left(\hat{P}\hat{H}^T\right) \tag{5.7}$$

5.2.1 Estimation of Q

First the analytic solution to derive an estimate of Q is proposed.

State covariance matrix P could be expressed as

$$P = FPF^T + M + \Gamma Q \Gamma^T \tag{5.8}$$

where Γ is the input matrix of dynamic system and

$$M = F\left(-KHP - PH^TK^T + KC_0K^T\right)F^T \tag{5.9}$$

Now a recursive solution for equation (5.8) is considered. In a first substitution one has

$$P = F\left(FPF^T + M + \Gamma Q \Gamma^T\right)F^T + M + \Gamma Q \Gamma^T \tag{5.10}$$

or

$$P = F^2P\left(F^2\right)^T + FMF^T + F\Gamma Q \Gamma^T F^T + M + \Gamma Q \Gamma^T \tag{5.11}$$

and after q substitutions one gets

$$P = F^qP\left(F^q\right)^T + \sum_{j=0}^{q-1} F^jM\left(F^j\right)^T + \sum_{j=0}^{q-1} F^j\Gamma Q \Gamma^T\left(F^j\right)^T \tag{5.12}$$

Before solving equation (5.12) for Q, it is first necessary to extract the set of equations for which only knowledge of PH^T (estimated in (5.5)) is needed. These equations are attained by post-multiplying both sides of (5.12) by H^T and pre-multiplying by HF^{-q}. This gives

$$HF^{-q}PH^T = HP\left(F^q\right)^TH^T + HF^{-q}\sum_{j=0}^{q-1} F^jM\left(F^j\right)^TH^t + HF^{-q}\sum_{j=0}^{q-1} F^j\Gamma Q \Gamma^T\left(F^j\right)^TH^T \tag{5.13}$$

Given that P is symmetric, the HP product to the right of the equal sign can be expressed as $\left(PH^T\right)^T$ and one has

$$\sum_{j=0}^{q-1} HF^{j-q}\Gamma Q\Gamma^T\left(F^j\right)^T H^T = HF^{-q}PH^T - \left(PH^T\right)^T\left(F^q\right)^T H^T - \sum_{j=0}^{q-1} HF^{j-q}M\left(F^j\right)^T H^T \tag{5.14}$$

For convenience both sides are transposed

$$\sum_{j=0}^{q-1} HF^{j}\Gamma Q\Gamma^T\left(F^{j-q}\right)^T H^T = \left(PH^T\right)^T\left(F^{-q}\right)^T H^T - HF^qPH^T - \sum_{j=0}^{q-1} HF^{j}M\left(F^{j-q}\right)^T H^T \tag{5.15}$$

To shorten the previous equation

$$A_j = HF^j\Gamma \tag{5.16}$$

$$B_j = \Gamma^T\left(F^{j-q}\right)^T H^T \tag{5.17}$$

and

$$s_q = \left(PH^T\right)^T\left(F^{-q}\right)^T H^T - HF^qPH^T - \sum_{j=0}^{q-1} HF^jM\left(F^{j-q}\right)^T H^T \tag{5.18}$$

so equation (5.15) becomes

$$\sum_{j=0}^{q-1} A_jQB_j = s_q \tag{5.19}$$

Applying the *vec* operator to both sides of the previous equation one has

$$\sum_{j=0}^{q-1}\left(B_j^T \otimes A_j\right) * vec\left(Q\right) = vec\left(s_q\right) \tag{5.20}$$

where $\otimes$ denotes the Kronecker product. Equation (5.20) can be evaluated for as many q values as one desires, although it is evident that all the equations obtained are not necessarily independent. Furthermore a more compact form for the previous equation could be

$$C * vec\left(Q\right) = S \tag{5.21}$$

where

$$c_q = \sum_{j=0}^{q-1} \left(B_j^T \otimes A_j \right) \tag{5.22}$$

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_q \end{bmatrix} \tag{5.23}$$

$$S = \begin{bmatrix} vec\,(s_1) \\ vec\,(s_2) \\ \vdots \\ vec\,(s_q) \end{bmatrix} \tag{5.24}$$

The matrix Q is symmetrical and can often be assumed diagonal. The constraints of symmetry and/or a diagonal nature of Q can be reflected in a linear transformation of the form

$$vec\,(Q) = T * vec\,(Q_r) \tag{5.25}$$

where $vec\,(Q_r)$ is the vector of unknown entries in Q after all the constraints are imposed. By definition, the true matrix Q is positive semi-definite (i.e., all its eigenvalues are ≥ 0), however, due to approximations, the least square solution may not satisfy this requirement. In the general case one can satisfy positive semi-definitiveness by recasting the problem as an optimization with constraints. Namely, minimize the norm of $(C \times vec\,(Q) - S)$ subject to the constraint that all eigenvalues of $Q \geq 0$. This problem is particularly simple for the case where Q is diagonal.

Compared with the analytic solution, the approximated one is very simple

$$\Gamma Q \Gamma^T \approx K \hat{C}_0 K^T \tag{5.26}$$

Knowing Γ structure, the diagonal elements of Q can be easily extracted from $\Gamma Q \Gamma^T$

5.3 Implementation of AKF for Human Kinematics Estimation

Some modifications have been introduced with respect to the KF model of Section 3.3. Firstly, there is not a unique KF that estimates and filters the entire state (3.29), but four Kalman Filters work in parallel. The state vector of the first one is composed of the four Kinematic DOFs of Human Walking (θ, x, y and ρ), their velocities and accelerations (jerk is not inserted). In the states of the second and the third filters there are the four joints of the Right and Left Arms, their velocities and accelerations, while the last one is composed of parameters $\boldsymbol{\pi}$ (the distances between skeletal points).

So, for the first three filters

$$\boldsymbol{s}_{k+1} = F\boldsymbol{s}_k + \Gamma\boldsymbol{w}_k \tag{5.27}$$

$$\boldsymbol{s}_k = \begin{bmatrix} \boldsymbol{q}_k \\ \dot{\boldsymbol{q}}_k \\ \ddot{\boldsymbol{q}}_k \end{bmatrix} \qquad F = \begin{bmatrix} I & \Delta t I & \frac{\Delta t^2}{2} I \\ 0 & I & \Delta t I \\ 0 & 0 & I \end{bmatrix} \tag{5.28}$$

where $\boldsymbol{w}_k$ is a Gaussian stationary white noise sequence with zero mean and covariance Q. Notice that the model of process noise covariance matrix is changed (see (3.31)). Model of Γ is taken from [7]

$$\Gamma = \begin{bmatrix} \frac{\Delta t^3}{6} I \\ \frac{\Delta t^2}{2} I \\ \Delta t I \end{bmatrix} \tag{5.29}$$

Q is considered diagonal and each diagonal element is the covariance of the jerk related to a Kinematic parameter.

The observation model is very similar to (3.32), only $\boldsymbol{z}_k$ and H are slightly different from (3.33)

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{q}_{1,k} \\ \vdots \\ \boldsymbol{q}_{n,k} \end{bmatrix} \qquad H = \begin{bmatrix} I & 0 & 0 \\ \vdots & \vdots & \vdots \\ I & 0 & 0 \end{bmatrix} \tag{5.30}$$

For the last KF

$$\boldsymbol{\pi}_{k+1} = F\boldsymbol{\pi}_k \qquad F = I \tag{5.31}$$

So, for parameters $\boldsymbol{\pi}$, process noise is not considered. In the observation model

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{\pi}_{1,k} \\ \vdots \\ \boldsymbol{\pi}_{n,k} \end{bmatrix} \qquad H = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \tag{5.32}$$

For the first three filters, after N time steps, matrix Q is adapted using the innovation sequence (matrix R is not adapted, because the covariance of the measurement noise is assumed to be known). Considering the approximated estimation (5.26), to extract diagonal elements of Q from $\Gamma Q \Gamma^T$, it is needed to take the last four diagonal elements of this matrix (that is a 12×12) and divided each of them by Δt^2.

Estimating Q, one could also adapt jerk bounds defined in Section 4.2, using the jerk covariance σ^2 (e.g. $\dddot{q}^{sup} = 3\sigma^2$ and $\dddot{q}^{inf} = -3\sigma^2$).

Chapter 6

Simulations in Matlab Environment

In this chapter, some simulations, used to check the correctness and efficiency of the algorithms introduced in Chapters 4 and 5, are presented. First, a simulation to test algorithm 3 is discussed. Then, algorithms of Section 4.2 are analysed. Finally the Adaptive Kalman filter is tested.

6.1 Simulation of Reachable Set Computation with estimated acceleration

The first Matlab script simulates the Swept Volumes generation of a human who walks in a room. No real data are used, in fact measurements are given by three simulated depth sensors and afterwards are fouled by a random noise. Then, the inverse kinematic is calculated, to obtain the Human Kinematic DOFs of Section 3.2. They are sent to the Kalman filter, that estimates their velocity, acceleration and jerk. Finally, Reachable Sets are computed and Swept Volumes are generated.

The Kalman filter parameters are

Parameter	**Value**
Δt	0.004 [s]
Process noise covariance	1
Measurement noise covariance	10
Initial state covariance	1000

Table 6.1: Parameters of Kalman Filter.

While positions, velocities and accelerations bounds are taken from [37]

Joint	Min Pos.	Max Pos.	Min Vel.	Max Vel.	Min Acc.	Max Acc.
x	$-\infty$ [m]	∞ [m]	0 $[m/s]$	0.80 $[m/s]$	-0.10 $[m/s^2]$	0.10 $[m/s^2]$
y	$-\infty$ [m]	∞ [m]	-0.80 $[m/s]$	0.80 $[m/s]$	-0.10 $[m/s^2]$	0.10 $[m/s^2]$
θ	$-\pi$ [rad]	π [rad]	$-\pi/4$ $[rad/s]$	$\pi/4$ $[rad/s]$	$-\pi/8$ $[rad/s^2]$	$\pi/8$ $[rad/s^2]$
ρ	$-\pi/6$ [rad]	$\pi/2$ [rad]	$-\pi/10$ $[rad/s]$	$\pi/10$ $[rad/s]$	$-\pi/20$ $[rad/s^2]$	$\pi/20$ $[rad/s^2]$
$\alpha_{r,1}$	$-\pi/20$ [rad]	$8\pi/9$ [rad]	$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{r,2}$	inequality 3.49b		$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{r,3}$	inequality 3.49c		$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{r,4}$	$\pi/9$ [rad]	π [rad]	$-3\pi/10$ $[rad/s]$	$3\pi/10$ $[rad/s]$	$-3\pi/20$ $[rad/s^2]$	$3\pi/20$ $[rad/s^2]$
$\alpha_{l,1}$	$-\pi/20$ [rad]	$8\pi/9$ [rad]	$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{l,2}$	inequality 3.49b		$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{l,3}$	inequality 3.49c		$-\pi/2$ $[rad/s]$	$\pi/2$ $[rad/s]$	$-\pi/4$ $[rad/s^2]$	$\pi/4$ $[rad/s^2]$
$\alpha_{l,4}$	$\pi/9$ [rad]	π [rad]	$-3\pi/10$ $[rad/s]$	$3\pi/10$ $[rad/s]$	$-3\pi/20$ $[rad/s^2]$	$3\pi/20$ $[rad/s^2]$

Table 6.2: Positions, Velocities and Accelerations Bounds.

Running the script, with braking time $T_s = 0.50\,s$, one obtains the plot shown in Figure 6.1.

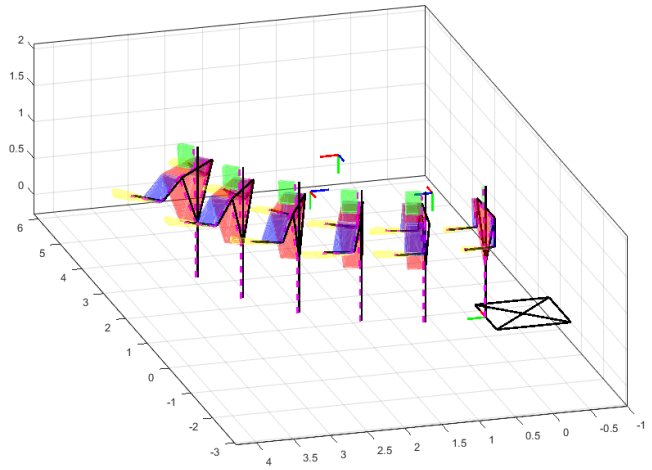
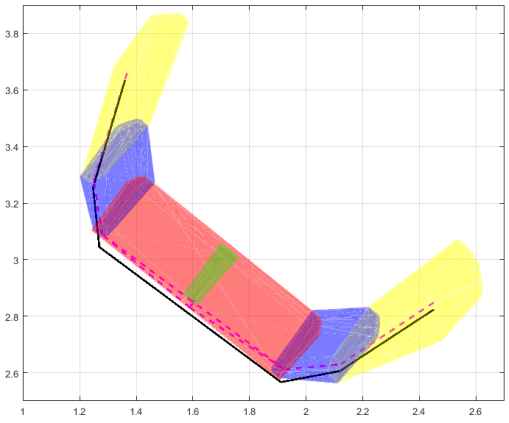
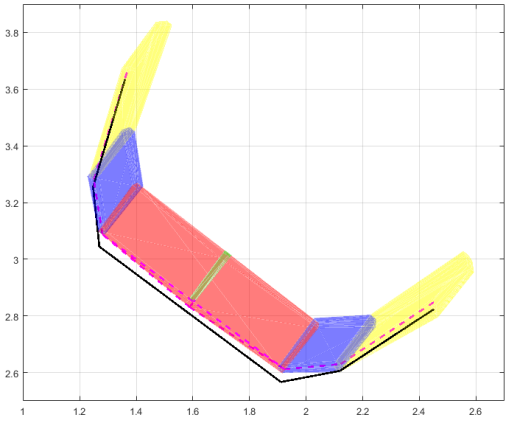


Figure 6.1: Swept volumes of a human who walks in a room.

Comparing Figure 6.2a with Figure 6.2b, it is easy to notice that swept volumes computed with algorithm 3 are smaller.



(a)



(b)

Figure 6.2: Figure 6.2a: Top-view of a human swept volumes computed with reachable set algorithm 1. Figure 6.2b: Top-view of a human swept volumes computed with reachable set algorithm 3.

The second part of this simulation consists in checking if the swept volumes, generated with both Reachable Set algorithms, contain Human position for the next T_s seconds. To test this fact, for each of the four swept volumes shown in Figure 6.3, GJK algorithm (see [13]) is used (notice that each T_s horizon is divided into six parts). It returns the distance between the volumes and skeletal points highlighted in Figure 3.6 (if a point is inside a volume, it returns zero).

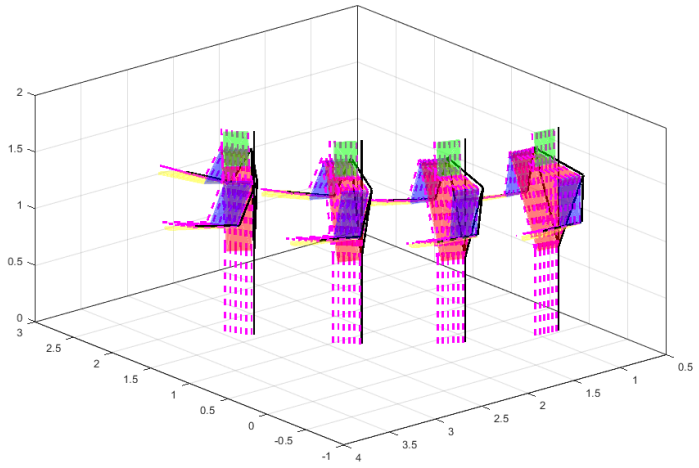
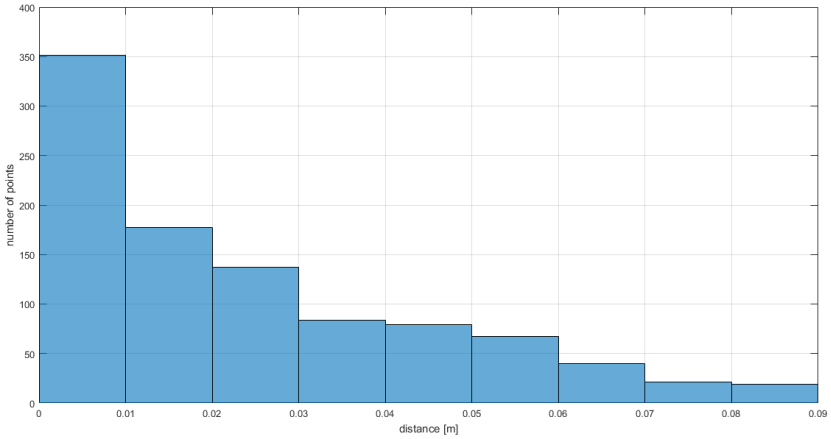


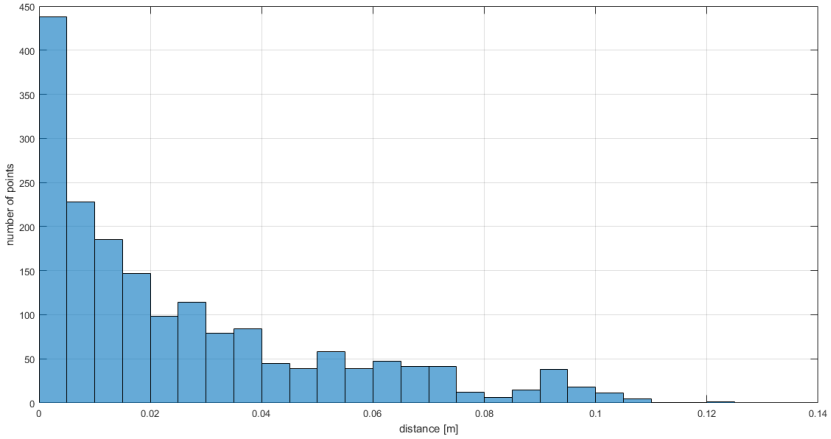
Figure 6.3: Swept volumes of a human, with position of skeletal points (in magenta) for the next T_s seconds.

The script is run ten times: the distances, found using algorithm 1, are collected in the histogram of Figure 6.4a, while the results for algorithm 3 are shown in Figure 6.4b.

Using the original algorithm, about 31% of the skeletal points are outside the swept volumes; the maximum distance is 8 cm and the mean is of 2 cm. For the modified algorithm, the points outside are more, about 57%; the maximum distance is 12 cm, while the mean is still of 2 cm. Notice that, in these tests, swept volumes are not augmented accounting for each body part dimension (see equation 3.50). This choice has been made to better appreciate the differences between the two algorithms, especially for second test, where the majority of points outside swept volumes are very close to them, so they would be inside the augmented spherical swept volumes.



(a)



(b)

Figure 6.4: Figure 6.4a: Distances between skeletal points and swept volumes using algorithm 1. Figure 6.4b: Distances between skeletal points and swept volumes using algorithm 3.

6.2 Simulations of Reachable Set Computation with bounded jerk

To test algorithms 6 - 7, first a simple simulation of trajectory generation is set up. Jerk is given and considered piecewise constant; from it acceleration, velocity and position are obtained. Then Reachable Set bounds are computed using algorithm 1 and algorithms 6 - 7.

Observing Figure 6.5, it can be noticed that the bounds computed with bounded acceleration are bigger than the ones calculated with bounded jerk.

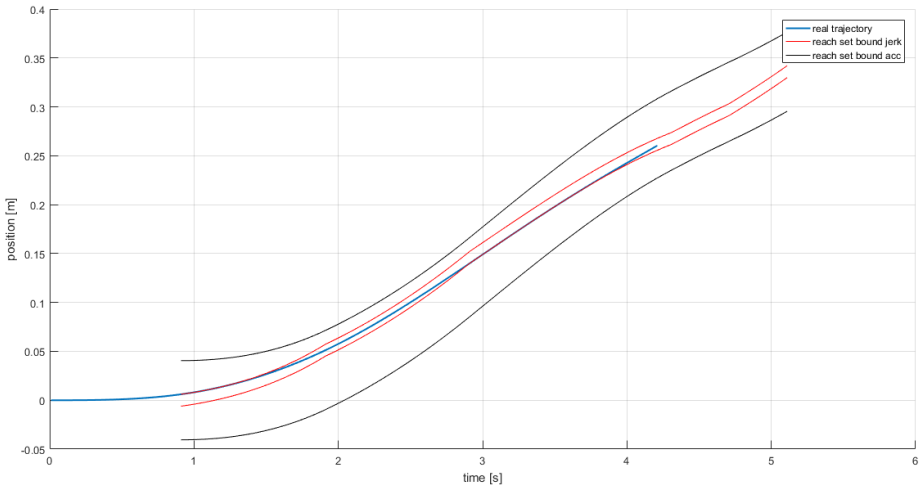


Figure 6.5: a trajectory, with Reachability bounds computed with bounded acceleration or bounded jerk.

The second Matlab script simulates the swept volumes generation of a seated human, who is performing some assembly operations. In this case, measurements are given by a single, but real, Kinect camera and logged in a file that the KF takes to estimate the state.

The only parameter that is changed from Table 6.1 is the discretization time Δt, that is now set equal to 0.03 s. DOF **x** minimum velocity is changed from 0 m/s to $-0.80\,m/s$, to maintain the hypothesis of symmetrical bounds given in Section 4.2; the other values of Table 6.2 remain unvaried. Jerk bounds are obtained taking maximum and minimum values of each DOF's estimated jerk and are summarized in Table 6.3.

Joint	**Min Jerk**	**Max Jerk**
x	-0.050 $[m/s^3]$	0.050 $[m/s^3]$
y	-0.050 $[m/s^3]$	0.050 $[m/s^3]$
θ	$-\pi/16$ $[rad/s^3]$	$\pi/16$ $[rad/s^3]$
ρ	$-\pi/40$ $[rad/s^3]$	$\pi/40$ $[rad/s^3]$
$\alpha_{r,1}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{r,2}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{r,3}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{r,4}$	$-3\pi/40$ $[rad/s^3]$	$3\pi/40$ $[rad/s^3]$
$\alpha_{l,1}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{l,2}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{l,3}$	$-\pi/8$ $[rad/s^3]$	$\pi/8$ $[rad/s^3]$
$\alpha_{l,4}$	$-3\pi/40$ $[rad/s^3]$	$3\pi/40$ $[rad/s^3]$

Table 6.3: Jerks Bounds.

In Figure 6.6, two swept volumes, obtained by running the script, are shown.

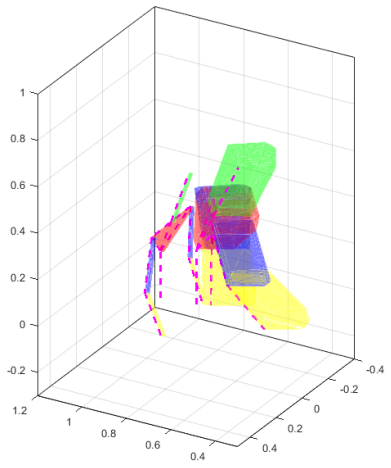
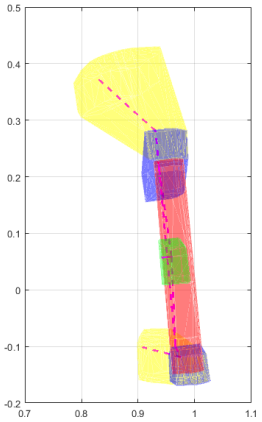


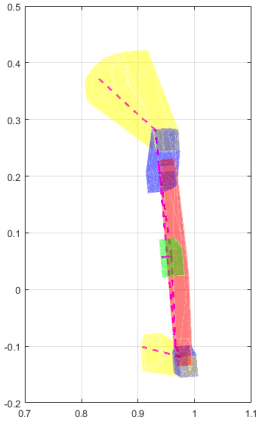
Figure 6.6: Swept volumes of a seated human, who picks up something from a table.

Comparing Figure 6.7a with Figure 6.7b, the difference between the two swept volumes is quite evident.

Notice that, if in some instants of time, estimated accelerations were near to their bounds, swept volumes computed with bounded jerk could be very similar to the ones computed with bounded accelerations.



(a)

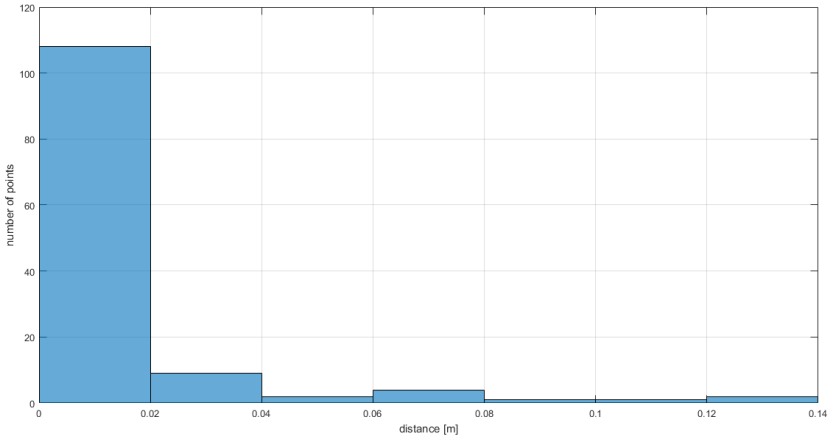


(b)

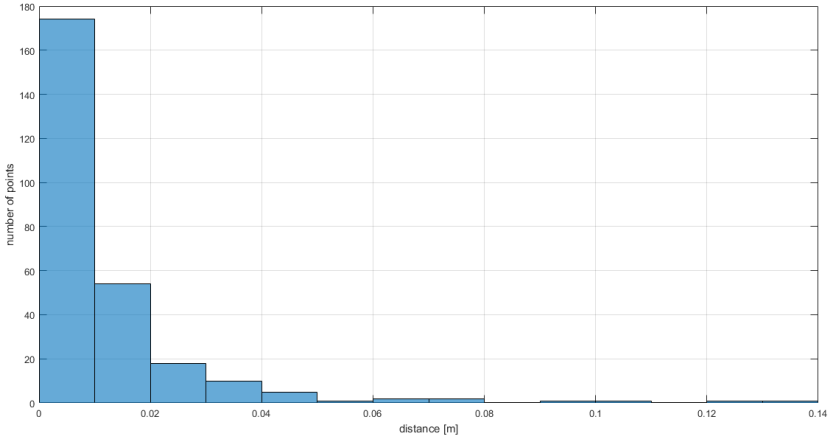
Figure 6.7: Figure 6.7a: Top-view of a human swept volumes computed with reachable set algorithm 1. Figure 6.7b Top-view of a human swept volumes computed with reachable set algorithms 6 - 7.

The same test, to check if the swept volumes contain human position for the next T_s seconds, is performed. The obtained distances are shown in Figure 6.8a and Figure 6.8b.

Using the original algorithm, about 24% of the skeletal points are outside the swept volumes; the maximum distance is 13 cm and the mean is of 1 cm. For the modified algorithm, the points outside are more, about 52%; while the maximum distance and the mean are still of 13 cm and 1 cm respectively.



(a)

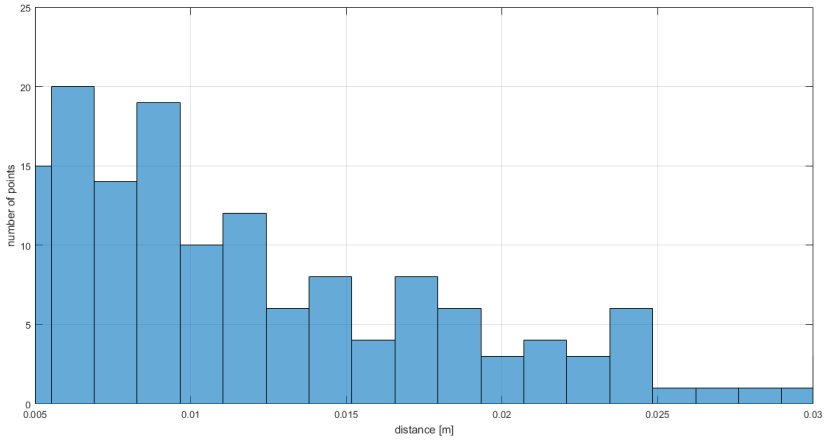


(b)

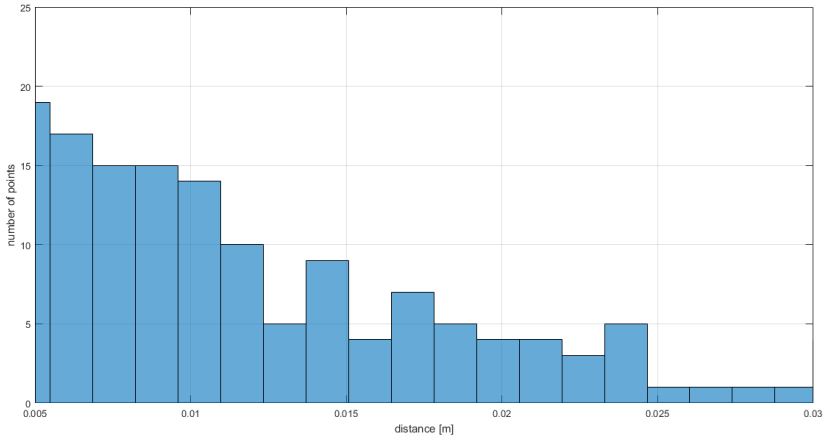
Figure 6.8: Figure 6.8a: Distances between skeletal points and swept volumes using algorithm 1. Figure 6.8b: Distances between skeletal points and swept volumes using algorithms 6 - 7.

Using the two simulations presented in this section, also a comparison between algorithms 3 and 6 - 7 has been made. The visual difference between the swept volumes generated with the two algorithms is practically null. But, performing second test, the number of points out of swept volumes is greater for algorithm 3, as shown in Figures 6.9a - 6.9b, even though the percentage difference is very small, around 2 % (in the two histograms, only the interval [0.005, 0.03] is reported, to better appreciate the difference). For this reason, but mostly because acceleration is generally not constant also for small windows of time (see studies on velocity and acceleration profiles of human arms, presented in [31]), algorithm with bounded jerk approximates better the Reachable Set of Human motion. Also for these simulations not-augmented swept volumes are considered, for the same reasons

described in previous section.



(a)



(b)

Figure 6.9: Figure 6.9a: Distances between skeletal points and swept volumes using algorithm 3. Figure 6.9b: Distances between skeletal points and swept volumes using algorithms 6 - 7.

6.3 Simulations of Adaptive Kalman Filter

The first test for the AKF is the estimation of the dynamics of a simulated system that has three DOFs. The state is composed of their positions, velocities and accelerations, while jerk is modelled as a zero mean Gaussian noise and only positions are measurable.

True values of process and measurement noise covariance matrices Q and R are

given and are

$$Q_{true} = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \end{bmatrix} \quad R_{true} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.8 \end{bmatrix} \tag{6.1}$$

But they are unknown to the AKF, that starts the estimation with

$$Q_0 = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \quad R_0 = \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.16 \end{bmatrix} \tag{6.2}$$

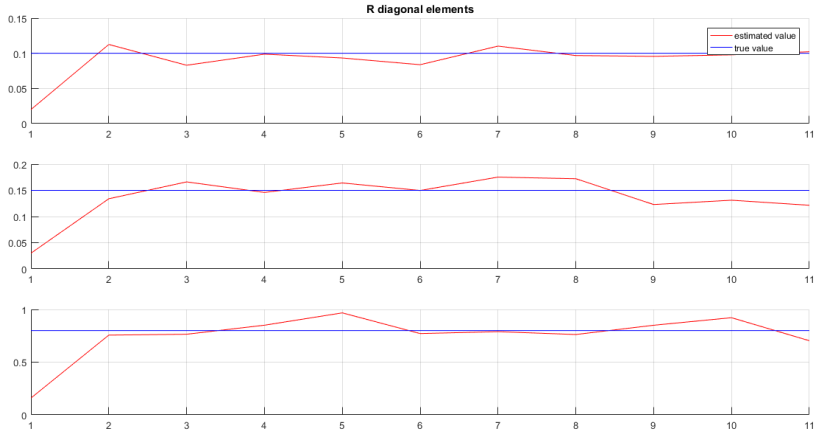
The innovation batches dimension is of 400; R is adapted using (5.7), while for Q both methods presented in subsection 5.2.1 are tested.

To minimize the norm of (5.21), subjected to the constraint that all Q values must be ≥ 0, Matlab function "fmincon" is utilized. For a $\Delta t = 0.5\,s$, R is estimated quite well (see Figure 6.10a); for Q, results are acceptable (see Figure 6.10b). But, if Δt is reduced, e.g. set equal to the one of the second simulation of the previous section ($\Delta t = 0.03\,s$), the Q estimate is totally wrong.

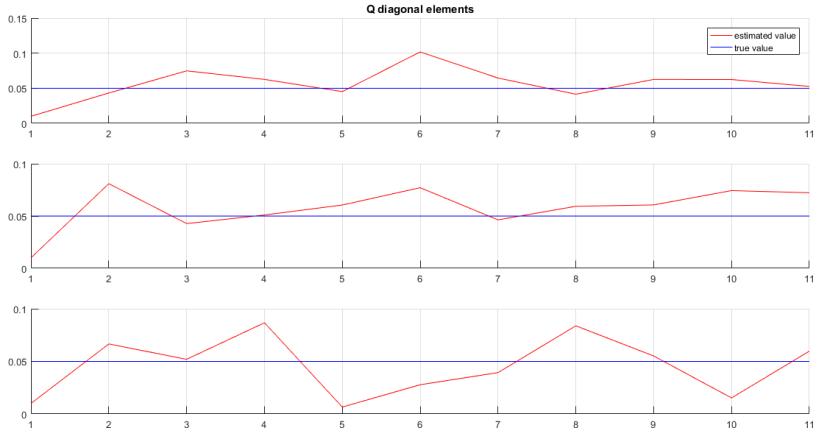
Instead, using approximation (5.26), Q estimate is good also for $\Delta t = 0.03\,s$ (see Figure 6.11). So, to adapt Q in the swept volume generation case, equation (5.26) will be utilized, while the first method is discarded.

Second simulation is the test of AKF insertion in swept volumes generation script illustrated in the previous section. The AKF structure is the one introduced in Section 5.3.

First, the whiteness test described in Section 5.2 is run: it reveals that the innovation sequence, for a batch of 400 samples, is not white. So matrix Q is adapted (R is kept fixed), using approximation (5.26) and, for each DOF, jerk bounds are computed as $\pm 3\sigma^2$, where σ^2 is the jerk covariance. After some adaptations, diagonal elements of Q take very small values (around 10^{-9}); the presence of very small values in the process noise covariance matrix seems to cause an under-estimation of the state with respect to the one obtained in previous section. However, a comparison cannot be made, because true values of the state are unknown.



(a)



(b)

Figure 6.10: Figure 6.10a: Estimation of measurement noise covariance matrix R. Figure 6.10b Estimation of process noise covariance matrix Q.

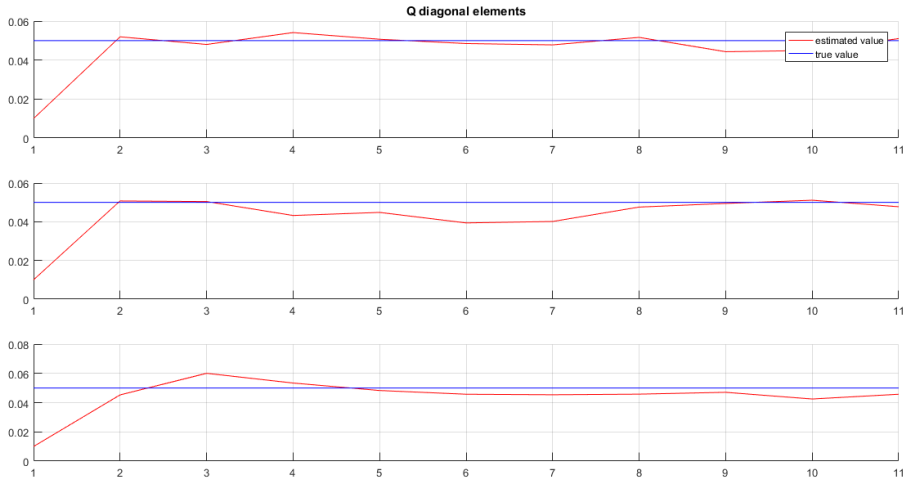


Figure 6.11: Estimation of process noise covariance matrix Q, using equation (5.26).

Chapter 7

Experimental Validations

In this chapter an experiment of HRC is described and results, obtained by applying techniques developed in this thesis, are presented. The robot has to accomplish a simple pick and place plus visual inspection task while cooperating with the human worker.

7.1 Experimental Setup

For the implementation and validation of the modified Reachable Set Computation algorithm and of the Adaptive Kalman Filter, experimental setup consists of:

- ABB FRIDA robot: a prototype of the currently commercially available ABB YuMi. It is a two arms collaborative robot; each arm has 7 joints and can be considered as an independent robot. A pair of grippers is mounted on the left arm, while on the right there is a camera, to perform visual inspection (see Figure 7.1). The robot is connected to an External PC through an Ethernet-based interface.

- Microsoft Kinect V2: a RGB-D motion sensing camera used to detect the presence of a human worker inside the robotic cell and to perform skeletal tracking (highlighted in Figure 7.1).

- External PC: a workstation responsible of computing all the quantities to generate swept volumes of the human worker, to compute safety constraints introduced in Section 3.1 and consequently to scale both arms' velocity. More in depth: once that human worker is tracked, measurements are exploited by the KF, that estimates human velocity, acceleration and jerk; then swept volumes are generated. After that, based on robot joints actual positions and velocities, and on swept volumes, safety constraints are computed. Finally,

velocity scaling factor δ is calculated and sent to the robot (see Section 3.5). Last two operations are performed for both robot's arms separately.

Figure 7.1: ABB FRIDA robot and Microsoft Kinect V2 (highlighted in green).

In Table 7.1, values of the different radii used to determine the sphere-swept volumes are displayed. Position bounds are unchanged with respect to the ones of Table 6.2, velocity, acceleration and jerk bounds instead are varied (for new values of velocity and acceleration a cue is taken from [41]).

The new values are reported in Table 7.2. Other relevant parameters values are:

$$T_s = 0.30\,s \tag{7.1}$$

$$Process\,noise\,covariance = 0.5 \tag{7.2}$$

$$Measurement\,noise\,covariance = 0.0001 \tag{7.3}$$

Body part	**radius [m]**
Head	0.1
Thorax	0.1
Upper Arm	0.03
Lower Arm	0.03

Table 7.1: Values of radii, used to generate spherical swept volumes.

Joint	**Min Vel.**	**Max Vel.**	**Min Acc.**	**Max Acc.**	**Min Jerk**	**Max Jerk**
x	-0.10 $[m/s]$	0.10 $[m/s]$	-0.50 $[m/s^2]$	0.50 $[m/s^2]$	-0.025 $[m/s^3]$	0.025 $[m/s^3]$
y	-0.10 $[m/s]$	0.10 $[m/s]$	-0.50 $[m/s^2]$	0.50 $[m/s^2]$	-0.025 $[m/s^3]$	0.025 $[m/s^3]$
θ	-0.25π $[rad/s]$	0.25π $[rad/s]$	$-\pi$ $[rad/s^2]$	π $[rad/s^2]$	-0.05π $[rad/s^3]$	0.05π $[rad/s^3]$
ρ	-0.1π $[rad/s]$	0.1π $[rad/s]$	-0.5π $[rad/s^2]$	0.5π $[rad/s^2]$	-0.025π $[rad/s^3]$	0.025π $[rad/s^3]$
$\alpha_{r,1}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{r,2}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{r,3}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{r,4}$	-0.15π $[rad/s]$	0.15π $[rad/s]$	-0.3π $[rad/s^2]$	0.3π $[rad/s^2]$	-0.015π $[rad/s^3]$	0.015π $[rad/s^3]$
$\alpha_{l,1}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{l,2}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{l,3}$	-0.25π $[rad/s]$	0.25π $[rad/s]$	-0.6π $[rad/s^2]$	0.6π $[rad/s^2]$	-0.03π $[rad/s^3]$	0.03π $[rad/s^3]$
$\alpha_{l,4}$	-0.15π $[rad/s]$	0.15π $[rad/s]$	-0.3π $[rad/s^2]$	0.3π $[rad/s^2]$	-0.015π $[rad/s^3]$	0.015π $[rad/s^3]$

Table 7.2: Velocities, Accelerations and Jerks Bounds.

The work cell is shown in Figure 7.2. The robot performs the following tasks:

1. Left arm takes a board from the feeder (highlighted in green) and places it in the area in the blue circle.
2. Right arm performs a visual inspection.
3. Left arm takes the board again and places it on the slide (highlighted in yellow).

While human operator:

1. takes a fuse from the red box on the right and mounts it on the board.
2. takes a chip from the red box on the left and affixes it on the board.
3. puts the assembled board on the feeder and takes a new board.

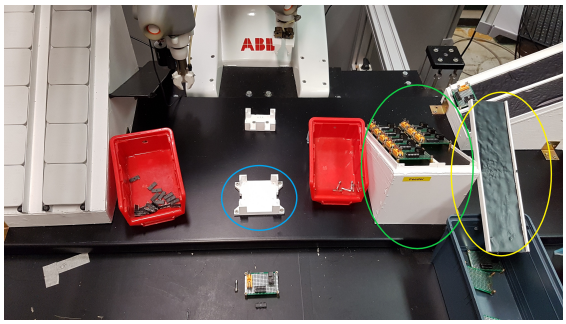


Figure 7.2: A view of the work cell where human operator and robot collaborate.

To better understand the whole operation, in Figure 7.3 there are some screenshots of the tasks accomplished by the robot, without operator presence, while in Figure 7.4 robot and human operator are working together.

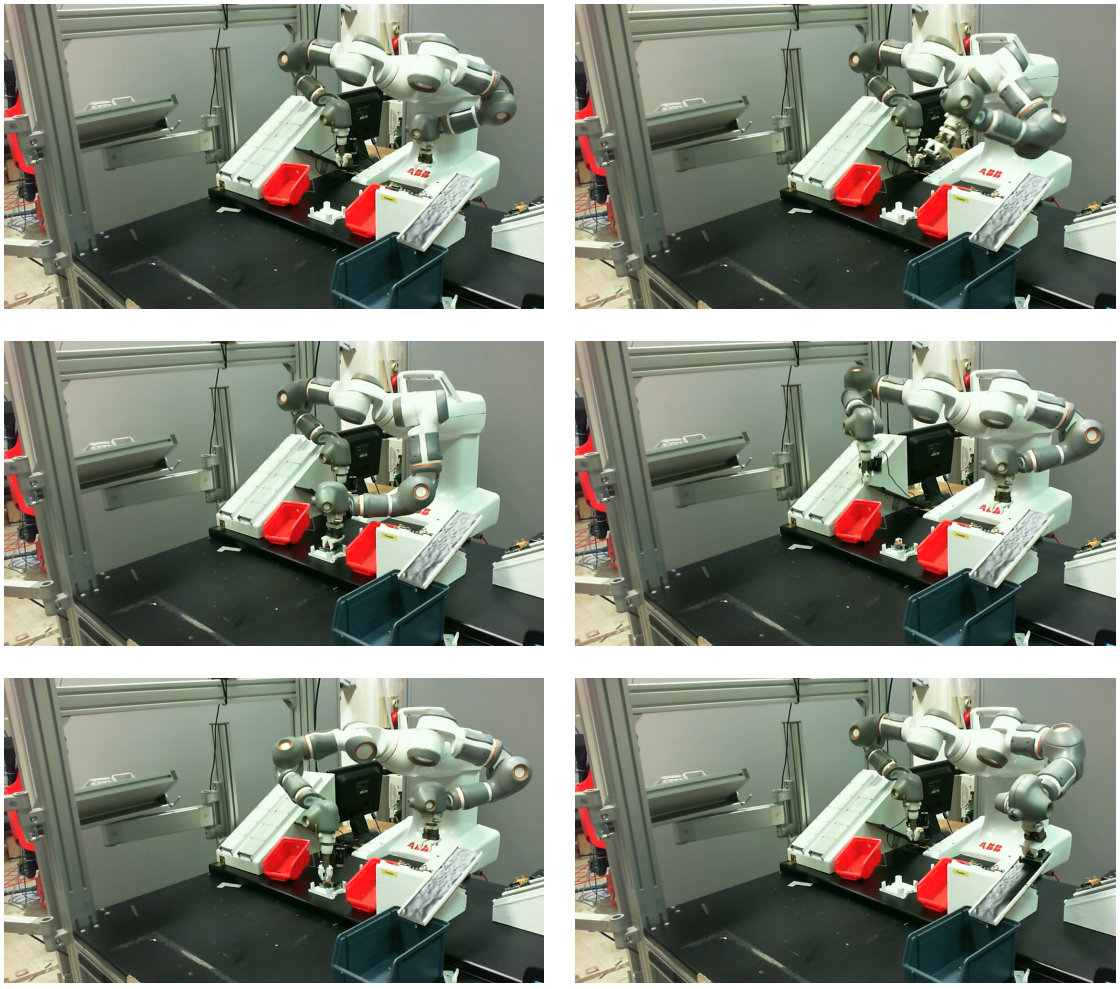


Figure 7.3: Some screenshots taken during the experiment, without human operator presence.

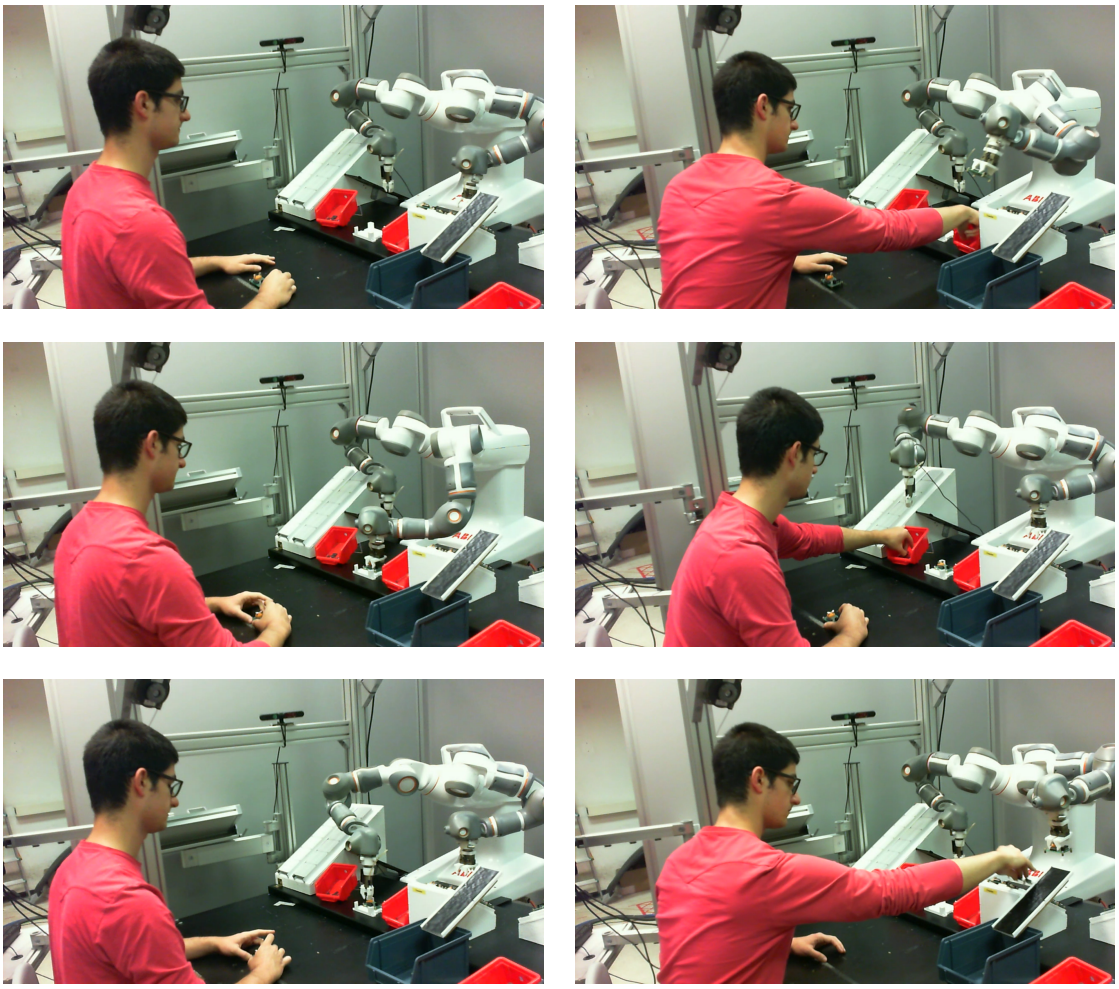


Figure 7.4: Some screenshots taken during the HRC experiment.

7.2 Results

The experiment is performed three times: in the first one Reachable Set is computed using algorithm 1. Second time algorithms 6 - 7 are utilized. In the last test Reachable set is still computed with bounded jerks, but the Adaptive Kalman Filter is inserted and jerk bounds are adapted accordingly. Human operator tries to perform the same sequence of movements for all the three experiments, to have repeatability, but it is impossible to avoid variations.

A video of these three experiments can be found in [35]. In the first experiment, where bounded accelerations are used, after one of the robot arms has stopped to avoid collisions with human operator, it takes a long time to start again its motion, even though the worker completes his task quite quickly. Furthermore, sometimes robot does not stop when the worker is performing his task, due to a not perfect estimate of human movements: no collision happens, but it would be dangerous if

the carried object was bigger than an electronic board.

In the second test, when bounded jerks are used, after a stop, robot arm motion restarts faster, because the generated swept volumes are smaller. Like for the first experiment, sometimes the robot does not stop, again due to an imprecise estimation.

Finally, when the AKF is introduced, robot always stops when the operator is performing a task, so the estimate is improved. Moreover, the robot movements restart quite fast after a stop, like in the previous experiment.

Now the experiments are examined more in depth, considering a time window of 100 seconds. Every time the right arm (called Robot1) has to stop to avoid collisions, the interval, in which its speed scaling factor δ is equal to zero, lasts on average 2 seconds, when bounded accelerations are used. Furthermore, once over four cycles, Robot1 does not halt, even if the operator is performing his task (δ behaviour is shown in Figure 7.5, the black arrows approximately indicate the time instant in which the operator enters in robot workspace). In the second test, where bounded jerks are used, the interval is smaller, about 0.9 seconds and Robot1 stops its motion whenever human worker is taking a fuse from left box (see Figure 7.6). Finally, when AKF is utilized, δ is equal to zero for approximately 1 second and Robot1 halts during all four cycles, as one can notice from Figure 7.7.

Considering the left arm (called Robot2), differences between the three tests are more evident. In the first experiment, each interval, in which speed scaling factor is null, lasts on average 2 seconds. Moreover, human operator enters in the scene seven times, but Robot2 stops only six times; this fact also underlines that, using bounded accelerations, three and a half cycles are completed during the considered time window (human operator performs a task twice a cycle in Robot2 workspace). Using bounded jerks, each interval lasts 0.7 seconds, but Robot2 stops only six times out of eight, because of an imprecise estimate. However, notice that four complete cycles are accomplished, unlike the previous test. In the last experiment, δ is kept equal to zero for a longer time compared with previous case, in fact intervals last on average 1 seconds, nevertheless, four cycles are completed and Robot2 halts whenever human worker is performing a task, so the estimation of his motion is more precise. Plots of δ for the three experiments are shown in Figures 7.8, 7.9, 7.10.

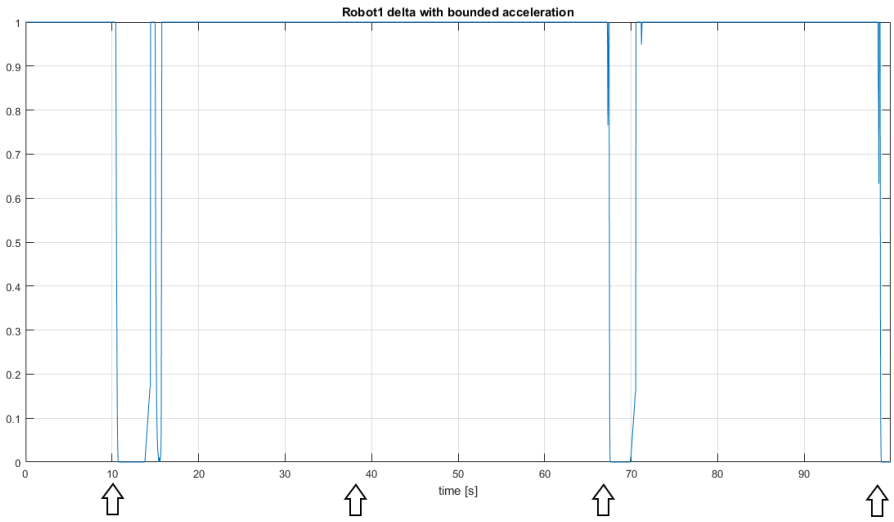


Figure 7.5: Robot1 speed scaling factor for bounded accelerations test.

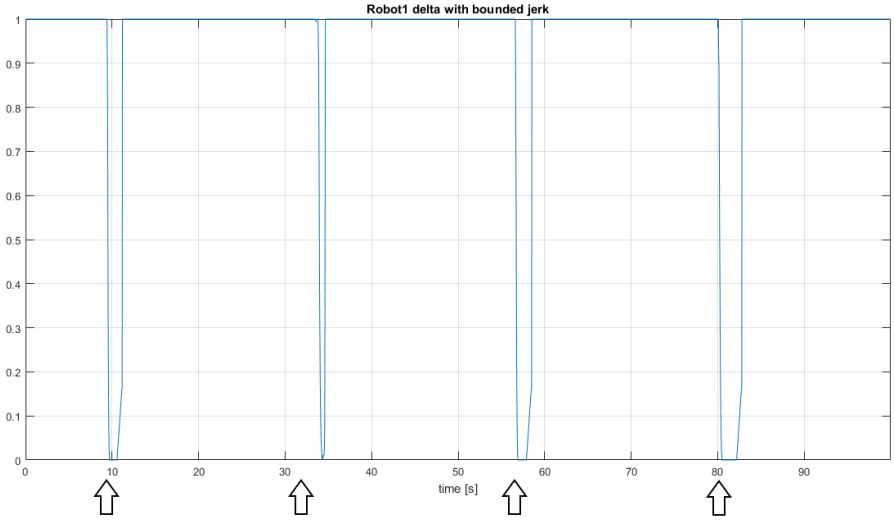


Figure 7.6: Robot1 speed scaling factor for bounded jerks test.

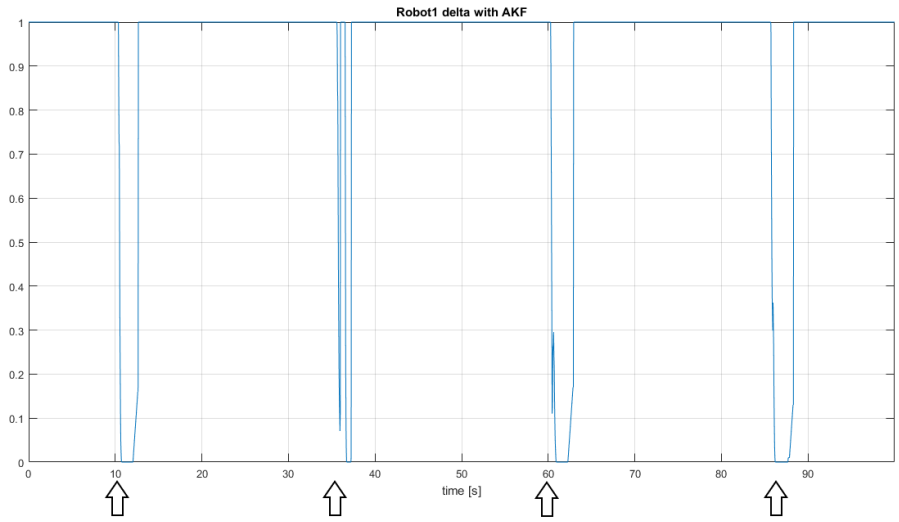


Figure 7.7: Robot1 speed scaling factor for AKF test.

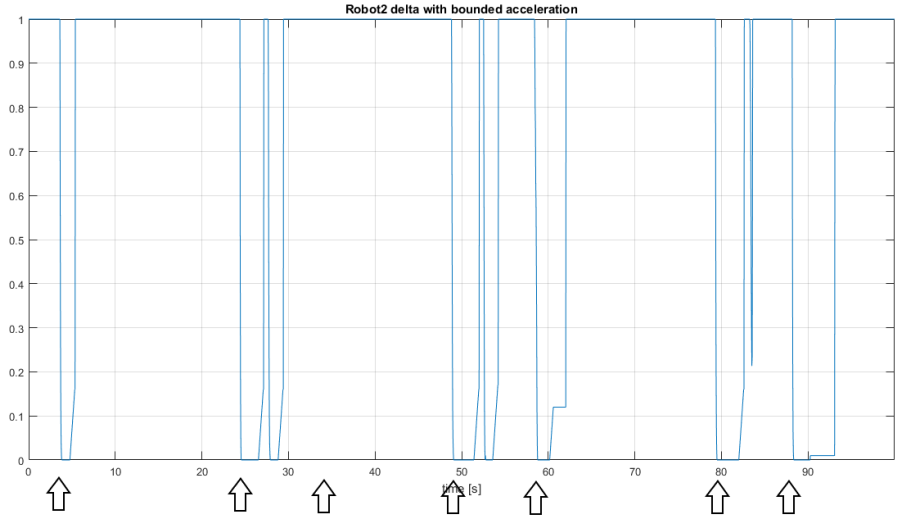


Figure 7.8: Robot2 speed scaling factor for bounded accelerations test.

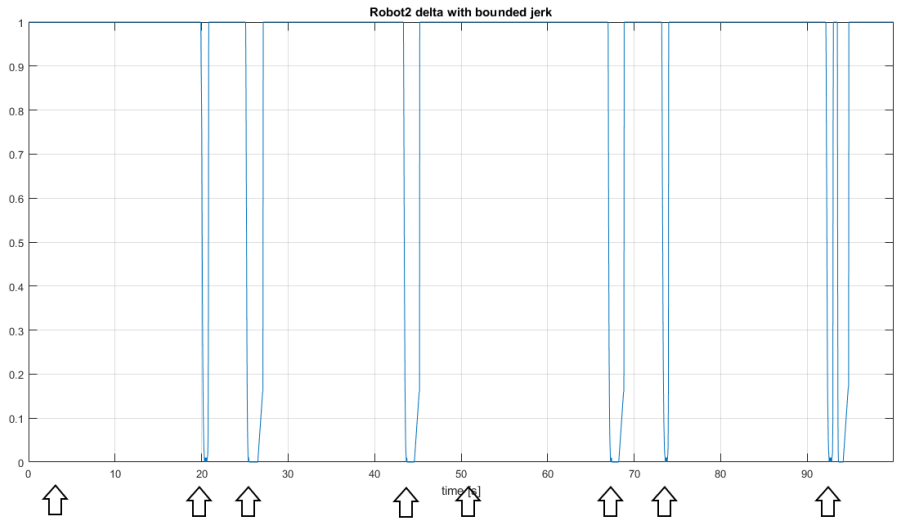


Figure 7.9: Robot2 speed scaling factor for bounded jerks test.

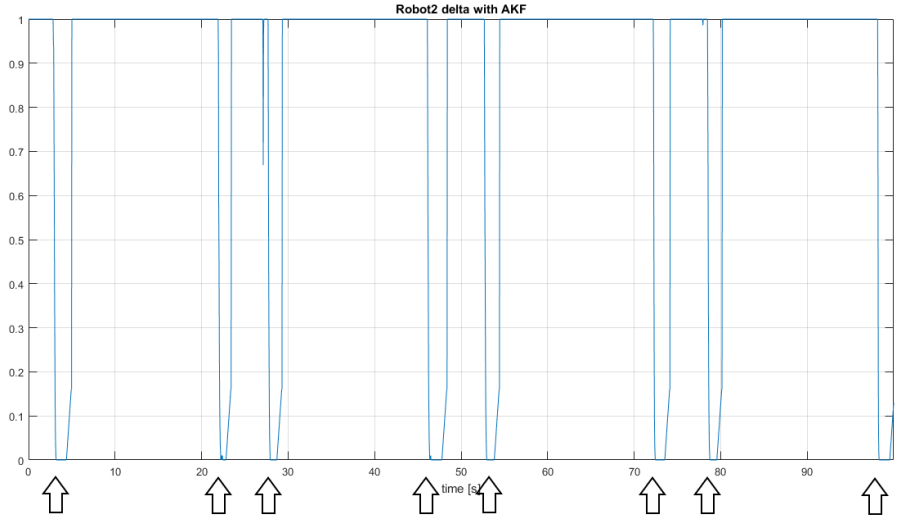


Figure 7.10: Robot2 speed scaling factor for AKF test.

Robot1 and Robot2 joints velocities can be found in Figure 7.11; time intervals during which δ is null are highlighted in black.

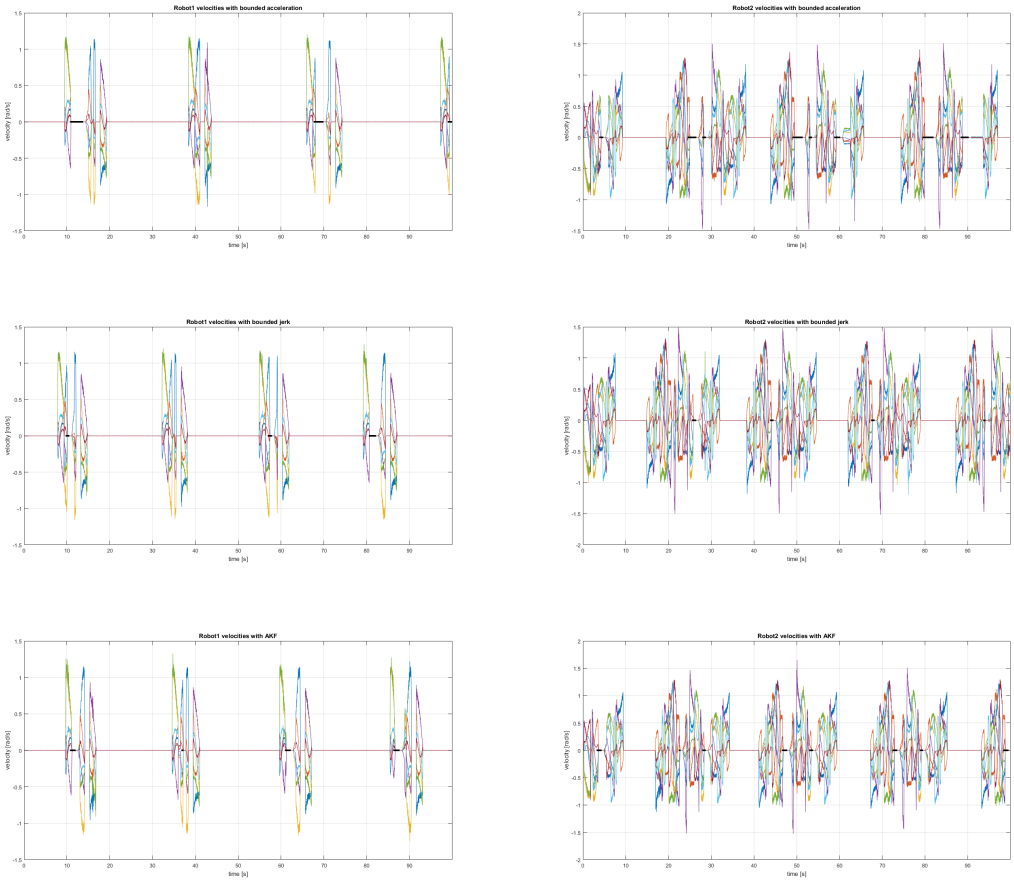


Figure 7.11: On top, there are joints velocities of Robot1 and Robot2 for bounded accelerations test. In the middle, one can found velocities for bounded jerks test. At the bottom, velocities for AKF test are shown.

Finally, swept volumes of the human operator are generated for three noticeable time instants (the ones in which the worker is performing a task). It is easy to notice that swept volumes generated with bounded accelerations are bigger than the ones generated with bounded jerks (see Figure 7.12). Swept volumes obtained in the AKF test are not shown, because differences with respect to the second test are minimum, due to the fact that jerk contribution for Reachable Sets computation is very small, so, also adapting jerk bounds, quite the same swept volumes are obtained.

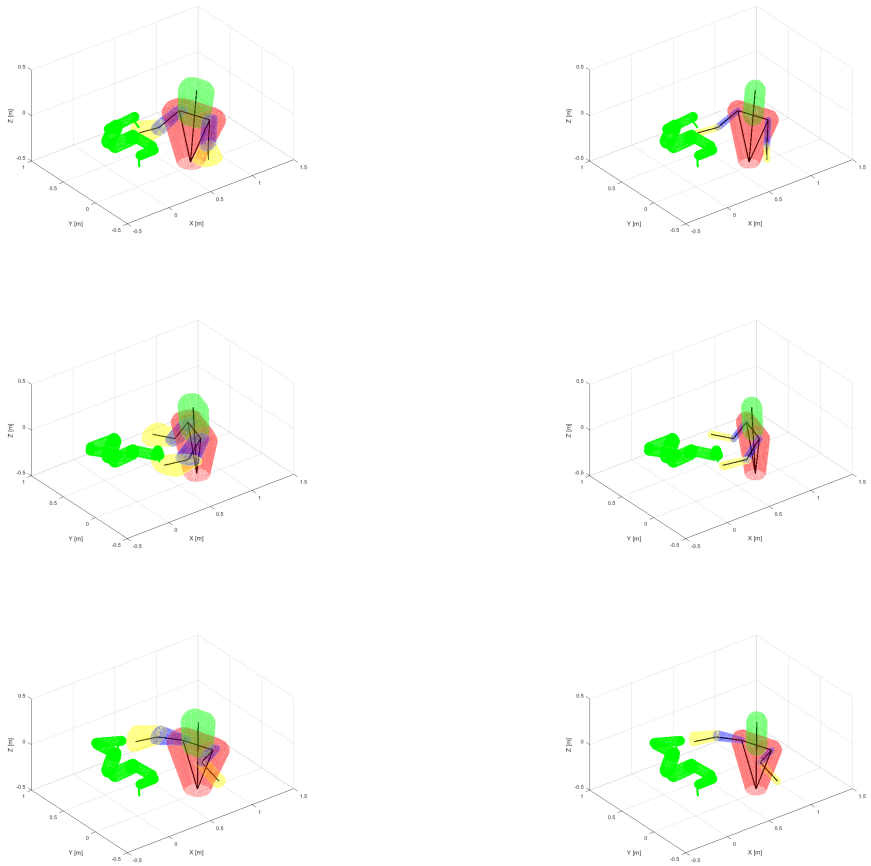


Figure 7.12: On top, there are swept volumes of human operator taking a fuse. In the middle, one can found swept volumes of the human picking up a chip. At the bottom, swept volumes of the worker, putting a board on the feeder, are shown.

Chapter 8

Conclusions

This thesis aims at giving a contribution to make the Swept Volumes generation adaptive and less conservative, in order to increase robot's productivity, while preserving human operator safety, during cooperative tasks. Two main solutions have been developed. The former is a modification of the algorithm computing Reachable Set of Human Motion: lower and upper bounds are computed on the basis of human joints position, velocity and acceleration, while jerk is supposed to be bounded. The latter is the introduction of an Adaptive Kalman Filter, that substitutes the Linear Kalman Filter. The peculiarity of AKF is that process noise covariance matrix is adapted online, to retrieve its true value, so a better estimate is obtained and, considering jerk as process noise, jerk bounds previously defined can be adapted from jerk covariance.

After a brief introduction to Human-Robot Collaboration and an overview of the state of the art of Human Detection & Tracking and Intention Estimation (Chapter 2), a review of the background on Swept Volumes generation has been given (Chapter 3). To better understand algorithms and techniques introduced in next chapters, Section 3.3 and Section 3.4 are of particular importance. In the former LKF implementation has been explained, and in the latter, Reachable Set computation algorithm with bounded acceleration has been defined.

An effective contribution of the thesis has been the modification of Reachable Set computation algorithm (Chapter 4). At first, bounded accelerations have been substituted with estimated ones, and then, fixed bounds on jerk have been introduced. In Chapter 5 the Adaptive Kalman Filter has been proposed: the chosen method is Innovation-Based and two different strategies to estimate process noise covariance matrix have been given.

After this theoretical part, some Matlab simulations have been carried out

(Chapter 6). The swept volumes, generated with new algorithm, are less conservative, while the AKF works well in a 3 DOFs simulated system, but a comparison with the LKF in the swept volumes generation case cannot be given, because the estimated positions, velocities and accelerations values are different in the two case, but the true values are unknown.

Finally, experiments of HRC have been performed (Chapter 7). From the analysis of these tests, it has been proven that robot productivity has been increased by the algorithm with bounded jerk. It has also been demonstrated that the AKF provides a better estimate, in fact robot always stopped when human operator was entering in its workspace, while using a LKF sometimes it did not halt.

Considering possible future works, one of the most interesting might be the introduction of Machine Learning strategies into Swept Volumes generation. Human operator usually performs repetitive tasks, so swept volumes could be adapted according to the knowledge of what he/she is doing. Other works could focus their attention on a further improvement of the Kalman Filter estimate, for example introducing new sensors, as for instance accelerometers, or accounting for time-delayed measurements.

Bibliography

[1] *ANSI/RIA R15 06 1999 "Safety requirements for industrial robots and robot systems".*

[2] *ISO 10218-1:2011 "Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots".*

[3] OpenNI - the standard framework for 3D sensing. `www.openni.ru`, 2013.

[4] I. Al-Naimi and C. B. Wong. Indoor human detection and tracking using advanced smart floor. In *Information and Communication Systems (ICICS), 2017 8th International Conference on*, 2017.

[5] R. Asaula, D. Fontanelli, and L. Palopoli. Safety provisions for human/robot interactions using stochastic discrete abstractions. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2175–2180, Oct 2010.

[6] Y. Bulut, D. Vines-Cavanaugh, and D. Bernal. Process and measurement noise estimation for Kalman filtering. In *IMAC-XXVIII*, 2010.

[7] F. De Groote, T. De Laet, I. Jonkers, and J. De Schutter. Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis. *Journal of Biomechanics*, 41(16):3390 – 3398, 2008.

[8] J. W. Ding and C. Rizos. Improving adaptive kalman estimation in gps/ins integration. *Journal Of Navigation*, 60:517–529, 2007.

[9] A. Durdu, I. Erkmen, and A. M. Erkmen. Estimating and reshaping human intention via human-robot interaction. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24:88–104, 2016.

[10] K. Ehlers and K. Brama. A human-robot interaction interface for mobile and stationary robots based on real-time 3d human body and hand-finger pose estimation, 2016.

[11] M. Elshafie and G.M. Bone. Markerless human tracking for industrial environments. In *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*, pages 001139–001144, May 2008.

[12] F. Flacco, T. Kröger, A. De Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *IEEE International Conference on Robotics and Automation*, 2012.

[13] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *Robotics and Automation, IEEE Journal of*, 4(2):193–203, 1988.

[14] D. Han, H. Nie, J. Chen, and M. Chen. Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection. *Robotics and Computer-Integrated Manufacturing*, 2017.

[15] R. Hayne, R. Luo, and D. Berenson. Considering avoidance and consistency in motion planning for human-robot manipulation in a shared workspace. In *Robotics and Automation (ICRA), IEEE International Conference on*, 2016.

[16] G. M. Jenkins and Watts D. G. Spectral analysis and its applications. *San Francisco: Holden Day Publ.*, 1968.

[17] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[18] M. Karasalo and X. Hu. An optimization approach to adaptive Kalman filtering. *Automatica*, 47(8):1785–1793, 2011.

[19] Dana Kulić and Elizabeth Croft. Pre-collision safety strategies for human-robot interaction. *Autonomous Robots*, 22(2):149–164, 2007.

[20] Y. Li and S. Ge. Human-robot collaboration based on motion intention estimation. In *IEEE/ASME Trans. Mechatron*, volume 19, pages 1007–1014, 2014.

[21] G. Liang, X. Lan, H. Zhang, X. Chen, and N. Zheng. Intention-based human robot collaboration. In *10th International Conference, ICIRA*, pages 605–613, 2017.

[22] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters. Phase estimation for fast action recognition and trajectory generation in human-robot collaboration. In *Sage journals*, 2015.

[23] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.

[24] P. S. Maybeck and P. Hanlon. Performance enhancement of a multiple model adaptive estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 31(4):1240–1254, 1995.

[25] A. Meguenani, V. Padois, J. Da Silva, A. Hoarau, and P. Bidaud. Energy based control for safe human-robot physical interaction. `https://hal.archives-ouvertes.fr/hal-01398790`, 2016.

[26] R. Mehra. On the identification of variances and adaptive kalman filtering. *IEEE Transactions on Automatic Control*, 15:175–184, 1970.

[27] A. H. Mohamed and Schwarz K. P. Adaptive Kalman filter for INS/GPS. *Journal of Geodesy*, 73:193–203, 1999.

[28] A. Mohammed, B. Schmidt, and L. Wang. Active collision avoidance for human-robot collaboration driven by vision sensors. *International Journal of Computer Integrated Manufacturing*, 2017.

[29] R. Mosberger and H. Andreasson. An inexpensive monocular vision system for tracking humans in industrial environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5850–5857, May 2013.

[30] M. Munaro, C. Lewis, D. Chambers, P. Hvass, and E. Menegatti. Rgb-d human detection and tracking for industrial environments. In *13th International Conference on Intelligent Autonomous Systems (IAS-13)*, July 2014.

[31] H. Nagasaky. Asymmetric velocity and acceleration profiles of human arm movements. *Experimental Brain Research*, 74:319–326, 1989.

[32] N. Najmaei, M.R. Kermani, and M.A. Al-Lawati. A new sensory system for modeling and tracking humans within industrial work cells. *Instrumentation and Measurement, IEEE Transactions on*, 60(4):1227–1236, April 2011.

[33] S. Pellegrinelli, F. L. Moro, N. Pedrocchi, L. Molinari Tosatti, and T. Tolio. A probabilistic approach to workspace sharing for human-robot cooperation in assembly tasks. *CIRP Annals - Manufacturing Technology*, 2016.

[34] A. Pereira and M. Althoff. Overapproximative arm occupancy prediction for human-robot co- -existence built from archetypal movements. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2016.

[35] MERLIN Lab POLIMI. Swept volumes generation in a HRC scenario. `https://youtu.be/omNNIHGbIoE`.

[36] M. Ragaglia. *Towards a safe interaction between humans and industrial robots through perception algorithms and control strategies.* PhD thesis, Politecnico di Milano, 2016.

[37] M. Ragaglia, A. M. Zanchettin, and P. Rocco. Trajectory generation algorithm for safe human-robot interaction based on multiple depth sensor measurements. *Automation Science and Engineering, IEEE Transactions on*, 2016.

[38] H. Ravichandar and A. Dani. Human intention inference using expectation-maximization algorithm with online model learning. In *IEEE Transactions on Automation Science and Engineering*, volume 14, 2017.

[39] H. Ravichandar, A. Kumar, and A. Dani. Bayesian human intention inference through multiple model filtering with gaze-based priors. In *19th International Conference on Information Fusion*, 2016.

[40] P. Rocco and A. M. Zanchettin. Robotica: la collaborazione uomo-macchina per creare nuovo valore. `https://www.fabbricafuturo.it`, 2016.

[41] J. Rosen, J. C. Perry, N. Manning, S. Burns, and B. Hannaford. The human arm kinematics and dynamics during daily activities, toward a 7 dof upper limp powered exoskeleton. In *12th International Conference on Advanced Robotics (ICAR)*, 2005.

[42] B. Schmidt and L. Wang. Contact-less and Programming-less Human Robot Collaboration. In *Forty Sixth CIRP Conference on Manufacturing Systems*, pages 545–550, 2013.

[43] D. Simon and Tien Li Chia. Kalman filtering with state equality constraints. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(1):128–136, Jan 2002.

[44] D. Simon and D.L. Simon. Kalman filtering with inequality constraints for turbofan engine health estimation. *Control Theory and Applications, IEEE Proceedings -*, 153(3):371–378, May 2006.

[45] H. Taubig, B. Bauml, and U. Frese. Real-time swept volume and distance computation for self collision detection. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011.

[46] K. Wakita, J. Huang, P. Di, K. Sekiyama, and T. Fukuda. Human-walking intention-based motion control of an omnidirectional-type cane robot. In *IEEE/ASME Trans. Mechatron*, volume 18, pages 285–296, 2013.

[47] X. Wang and Syrmos V. Vehicle health monitoring system using multiple model adaptive estimation. In *11th IEEE Mediterranean conference on control and automation*, 2003.

[48] Y. Yang and W. Gao. An optimal adaptive kalman filter. *Journal of Geodesy*, 80(4):177–183, 2006.

[49] A. M. Zanchettin and P. Rocco. Path-consistent safety in mixed human-robot collaborative manufacturing environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.

[50] A.M. Zanchettin, L. Bascetta, and P. Rocco. Achieving humanlike motion: Resolving redundancy for anthropomorphic industrial manipulators. *Robotics Automation Magazine, IEEE*, 20(4):131–138, 2013.

[51] A.M. Zanchettin, N.M. Ceriani, P. Rocco, H. Ding, and B. Matthias. Safety in human-robot collaborative manufacturing environments: Metrics and control. *Automation Science and Engineering, IEEE Transactions on*, PP(99):1–12, 2015.

[52] A.M. Zanchettin and P. Rocco. Constrained-based control strategy and reactive motion planning for robotic manipulators. *Robotics, IEEE Transactions on*, 2015.