Constraint Violation Probability Minimization for a Robot Manipulator

Scientific thesis for the procurance of the degree B.Sc.
from the Department of Electrical and Computer Engineering at the
Technical University of Munich.

Supervised by Univ.-Prof. Dr.-Ing./Univ. Tokio habil. Martin Buss
Michael Fink
Chair of Automatic Control Engineering

Submitted by cand. ing. Sirine Ayadi
Steinickeweg 7/60
80798 Munich
01788279686

Submitted on Munich, 11.06.2021

TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR STEUERUNGS- UND REGELUNGSTECHNIK
UNIV.-PROF. DR.-ING./UNIV. TOKIO HABIL. MARTIN BUSS
DR.-ING. MARION LEIBOLD

May 4, 2021

BACHELOR THESIS
for
Sirine Ayadi
Student ID 03714861, Degree EI

Constraint Violation Probability Minimization for a Robot Manipulator

Problem description:

Model Predictive Control (MPC) is a control method for a dynamical system that handles constraints [1]. In the presence of uncertainties, more advanced methods are necessary such as Robust Model Predictive Control (RMPC) and Stochastic Model Predictive Control (SMPC). However, both methods have disadvantages such as conservativeness for RMPC and risk for SMPC. A novel method was introduced recently which provides a minimization of the constraint violation (CVPM) probability for an MPC problem with uncertainty [2].
The CVPM method has the advantage that it is able to deal with changing constraints as well as to handle an uncertain system. Therefore, it is suitable for collision avoidance of a robot manipulator. In [3], linear MPC is already used on a feedback linarization of a manipulator. The aim of this work is to extend the MPC method for a robot manipulator to the CVPM-MPC method.

Tasks:

- Literature research for MPC, CVPM and robot manipulator
- Derive the robot dynamics
- Define a Feedback linearization for the robot
- Derive a MPC controller for the robot
- Extend the MPC controller to the CVPM method
- Evaluate the results

Bibliography:

[1] James B. Rawlings and David Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[2] Tim Brüdigam, Victor Gaßmann, Dirk Wollherr, and Marion Leibold. Minimization of constraint violation probability in model predictive control. *arXiv preprint arXiv:2006.02337*, 2020.

[3] Gian Paolo Incremona, Antonella Ferrara, and Lalo Magni. MPC for robot manipulators with integral sliding modes generation. *IEEE/ASME Transactions on Mechatronics*, 22(3):1299–1307, 2017.

Supervisor:	M. Sc. Michael Fink
Start:	06.04.2021
Intermediate Report:	11.06.2021
Delivery:	22.08.2021

M. Lei

(M. Leibold)
Akad. Rat

Abstract

Many systems in safety critical applications, such as human-robot interaction, rely on controllers that can handle constraints and cope with uncertainties. Model Predictive Control is a feedback control algorithm that ensures an optimal evolution of deterministic systems while fulfilling the constraints. This thesis deals with a multi-loop control scheme to solve control problems for a robot manipulator. The inner loop linearizes the robot dynamics and the external loop is closed relying on model predictive control such that the robot follows a reference trajectory. In the presence of uncertainties, the controller is extended to a method that not only bounds the constraint violation, but also minimizes it. The constraints are handled in a robust manner if they can be satisfied with zero probability of violation. Otherwise, the constraints are handled in a probabilistic way and the method yields a solution with minimal constraint violation probability. Afterwards, the solution is optimized with respect to an other control objective in order to follow the reference trajectory.

Contents

Chapter 1

Introduction

Model-Predictive Control (MPC) is a feedback control algorithm that uses a model to make predictions about future outputs of a plant, and it solves an optimization problem iteratively in order to select the optimal control action. MPC has many advantages compared to other controllers such as handling Multiple-Input Multiple-Output (MIMO) systems that have interactions between the inputs and outputs. It also handles constraints that are crucial for some systems. The constraints include input constraints that are enforced by the plant, an example is the physical limitation of the actuators that have limits on acceleration. There are also state constraints that are specified by the controller such as safety measures of human-robot-interaction or safety distance between cars in autonomous driving. Since MPC does not compensate for plant-model mismatch and persistent disturbances, some extensions such as Robust Model Predictive Control (RMPC) and Stochastic Model Predictive Control (SMPC) were introduced in order to address this issue. The RMPC approach assumes that the disturbance is bounded. Therefore, it is ensured that there is no violation of the constraints even in the presence of the worst-case uncertainty. The SMPC approach assumes that the distribution of the disturbance is known. A key feature of SMPC is the inclusion of probabilistic constraints, also known as chance constraints. It allows a small violation of the constraints by introducing a risk parameter. However, these two methods have some drawbacks. Applying RMPC for systems with high uncertainty leads to a conservative control law and a lack of performance. Also, SMPC has the disadvantage of ensuring recursive feasibility if the unlikely uncertainty occurs. Another improved form of MPC was introduced in [BGWL20], which is the Constraint Violation Probability Minimization (CVPM) method. In this approach, the constraint violation probability is minimized as a preprocessing step while ensuring that the MPC optimization problem remains feasible. A major advantage of this approach is that it can handle the changes over time of the support of uncertainty. Therefore, CVPM is beneficial for safety-critical systems. In this thesis, the method is used for a robot manipulator that is aimed to track a reference trajectory while minimizing the probability of colliding with an obstacle in the environment.

The thesis is organized as follows. The remaining of this Chap. 1 presents the problem statement and an overview of related work. The Chap. 2 introduces different approaches such as the dynamics of the robot and Inverse Dynamics approach. Afterwards, Chap. 3 illustrates the control scheme of the CVPM method. This method can be combined with standard linear MPC as well as RMPC. Additionally, it introduces an example how the workspace of the robot that includes an obstacle is transformed to the configuration space. Chapter 4 discusses the assumptions needed to prove recursive feasibility of the approach. Chapter 5 is devoted to present the results of simulation. Finally, Chap. 6 and Chap. 7 contain the discussion of the method and the conclusion respectively.

1.1 Problem Statement and Contribution

In this thesis, we investigate at first the behaviour of the robot under linear MPC. The linear MPC approach assumes a linear model of the plant. Since the robot is a non-linear dynamic system, a linearization of the plant is needed. In [IFM17], a feedback linearization of the robot using Inverse Dynamics approach is introduced, in order to compensate non-linearity and to reduce the system to decoupled Single-Input Single-Ouput (SISO) systems. This linearization happens in an inner loop and then the MPC optimization on the linearized plant is executed in an outer loop. In the presence of uncertainties due to unmodelled dynamics, MPC is combined with an Integral Sliding Mode (ISM) in [IFM17]. Instead of the ISM, we combine in this thesis the control scheme with the CVPM method introduced in [BGWL20] This extension has many advantages. If a change of probabilistic constraint occurs, the robot needs to leave the reference trajectory in order to reach the probabilistic constraint with zero probability of violation if possible, otherwise with a minimal constraint violation probability. The probabilistic constraints in this work represent the state set that guarantees a collision-free motion of the robot. This is important for safety-critical applications. For example, while the robot is moving and tracking a pre-defined trajectory, an obstacle can occur in the environment and it must be avoided. The system satisfies the constraints and minimizes the collision in which it reaches a collision-free space as well as it ensures that the trajectory of the robot stays as close as possible to the reference trajectory.

1.2 Related Work

We present in this section different relevant literature related to this thesis. MPC has significantly affected control engineering practice due to its natural ability to systematically handle constraints and optimize performance. It is discussed in many books and papers. The fundamentals of the method are introduced in [RMD17]. Different properties of the controller such as stability and recursive feasibility are investigated in [MRRS00]. Linear MPC is applied on the linearized robot manipulator

plant. This linearization is done using the Inverse Dynamics approach introduced in [SSVO10]. Solving Inverse Dynamics problem is useful for manipulator trajectory planning and control algorithm implementation. Once a joint trajectory is specified, Inverse Dynamics computations allow the computation of the torques to be applied to the joints to obtain the desired motion. The book [SSVO10] also explains the kinematics and dynamics of the robot manipulator and how the planning and control in general are implemented. The construction of the configuration space of the robot from a given workspace is introduced in [CLH$^+$05].
Since standard MPC cannot deal with the presence of persistent uncertainties, different extensions were developed to deal with this issue. For example, RMPC assumes that the disturbance is bounded and therefore considers the presence of the worst case uncertainty in the system. A tube-based RMPC approach is presented in [MSR05]. Another approach is SMPC that introduces probabilistic constraints. An overview of the different SMPC algorithms and the theoretical challenges are discussed in [Mes16]. A recent approach combining the benefits of RMPC and SMPC was first discussed in [BGWL20]. In this paper, the method is used in order to avoid an obstacle with an uncertain position. The CVPM method is also investigated for linear systems with linear constraints in [Fin20]. This work discusses the single-step approach, in which the optimization of the violation probability is only considered in the next predicted state. Also, the multi-step approach of the method is investigated, where the optimization probability is done for the next N_p estimated states. N_p is the prediction horizon of CVPM and it is smaller or equal to the prediction horizon N of MPC. Furthermore, it investigates stability and recursive feasibility of the CVPM method with underlying MPC.

Chapter 2

Background

$$\begin{aligned}
\boldsymbol{U}_N^*\left(\boldsymbol{x}_k\right) = &\arg\min_{\boldsymbol{U}_N} \sum_{i=0}^{N-1} [(\boldsymbol{x}_{k+i} - \boldsymbol{x}_{k+i,\mathrm{ref}})^\top \boldsymbol{Q} (\boldsymbol{x}_{k+i} - \boldsymbol{x}_{k+i,\mathrm{ref}}) && \text{(2.1a)} \\
&+ (\boldsymbol{u}_{k+i} - \boldsymbol{u}_{k+i,\mathrm{ref}})^\top \boldsymbol{R} (\boldsymbol{u}_{k+i} - \boldsymbol{u}_{k+i,\mathrm{ref}})] && \text{(2.1b)} \\
&+ (\boldsymbol{x}_N - \boldsymbol{x}_{N,\mathrm{ref}})^\top \boldsymbol{Q}_\mathrm{f} (\boldsymbol{x}_N - \boldsymbol{x}_{N,\mathrm{ref}}) && \text{(2.1c)} \\
&\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && \text{(2.1d)} \\
&\qquad\;\; \boldsymbol{U}_N \in \mathcal{U}_\mathrm{ad} && \text{(2.1e)}
\end{aligned}$$

This chapter is organized as follows. Section 2.1 introduces the dynamic system. In Sec. 2.2, the model and the dynamics of the robot are discussed. Section 2.3 describes the Inverse Dynamics approach. Section 2.4 introduces MPC and the optimization problem to be solved for the task of trajectory tracking and Sec. 2.5 is devoted for RMPC.

2.1 Dynamic System

The linear, time-invariant, discrete-time system description is

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{G}\boldsymbol{w}_k && \text{(2.2a)} \\
\boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k && \text{(2.2b)} \\
\boldsymbol{x}_0 &= \boldsymbol{x}(0) && \text{(2.2c)}
\end{aligned}$$

with time step $k \in \mathbb{I}$, where $\mathbb{I}$ is the set of positive integers, state transition matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, input matrix $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, output matrix $\boldsymbol{C} \in \mathbb{R}^{q\times m}$ and the initial state $\boldsymbol{x}_0$. The system is both controllable and observable. The states and the inputs of the system are polyhedrally bounded, such that $\boldsymbol{u}_k \in \mathcal{U}$ and $\boldsymbol{x}_k \in \mathcal{X}$. Both $\mathcal{U}$ and $\mathcal{X}$ are convex sets and contain at least the origin. The input set is defined as

$$\mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^m \mid \boldsymbol{A}_u\boldsymbol{u} \leq \boldsymbol{b}_u\} \tag{2.3}$$

and the state set is defined as

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}_x \boldsymbol{x} \leq \boldsymbol{b}_x\} \tag{2.4}$$

where $\boldsymbol{A}_x$ and $\boldsymbol{A}_u$ are matrices and $\boldsymbol{b}_x$ and $\boldsymbol{b}_u$ are vectors. The terminal set $\mathcal{X}_f$ is needed for the MPC optimization in Sec 2.4 and it is defined as

$$\mathcal{X}_f = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}_f \boldsymbol{x} \leq \boldsymbol{b}_f\}. \tag{2.5}$$

The disturbance $\boldsymbol{w}_k \in \mathcal{W}$ is also polyhedrally bounded. The system (2.2) is a model of the plant. It is used in order to make predictions about the future states over a prediction horizon. The predicted states over i-time steps starting from the current state $\boldsymbol{x}_0$ can be determined using recursive elimination as following

$$\boldsymbol{x}_i = \boldsymbol{A}^i \boldsymbol{x}_0 + \sum_{j=0}^{i-1} \boldsymbol{A}^{j-i-1} \boldsymbol{B} \boldsymbol{u}_j + \sum_{j=0}^{i-1} \boldsymbol{A}^{j-i-1} \boldsymbol{G} \boldsymbol{w}_j. \tag{2.6}$$

Using this equation, the system (2.2) can be rewritten with respect to the whole state sequence over i-time steps starting from the initial state $\boldsymbol{x}_0$. The lifted system dynamics results in

$$\boldsymbol{X}_i = \tilde{\boldsymbol{A}}_i \boldsymbol{x}_0 + \tilde{\boldsymbol{B}}_i \boldsymbol{U}_i + \tilde{\boldsymbol{G}}_i \boldsymbol{W}_i \tag{2.7}$$

with a stacked version of the state sequence, the input sequence and the disturbance sequence described as

$$\boldsymbol{X}_i = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_i \end{bmatrix}, \qquad \boldsymbol{U}_i = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{i-1} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{W}_i = \begin{bmatrix} \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{i-1} \end{bmatrix} \tag{2.8}$$

respectively. The lifted system matrix and the lifted input and disturbance matrices are

$$\tilde{\boldsymbol{A}}_i = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^i \end{bmatrix}, \qquad \tilde{\boldsymbol{B}}_i = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}^{i-1}\boldsymbol{B} & \boldsymbol{A}^{i-2}\boldsymbol{B} & \dots & \boldsymbol{B} \end{bmatrix} \tag{2.9}$$

$$\text{and} \qquad \tilde{\boldsymbol{G}}_i = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{G} & \boldsymbol{G} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}^{i-1}\boldsymbol{G} & \boldsymbol{A}^{i-2}\boldsymbol{G} & \dots & \boldsymbol{G} \end{bmatrix} \tag{2.10}$$

respectively. Additionally, the lifted constraints form should be determined. The constraints of the input sequence and state sequence over the prediction horizon i are

$$\underline{\boldsymbol{A}}_u \boldsymbol{U}_i \leq \underline{\boldsymbol{b}}_u \tag{2.11a}$$

$$\underline{\boldsymbol{A}}_x \boldsymbol{X}_i \leq \underline{\boldsymbol{b}}_x \tag{2.11b}$$

where the lifted matrices and vectors are

$$\underline{\boldsymbol{A}}_u = \begin{bmatrix} \boldsymbol{A}_u & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_u & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{A}_u \end{bmatrix}, \quad \underline{\boldsymbol{A}}_x = \begin{bmatrix} \boldsymbol{A}_x & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_x & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{A}_f \end{bmatrix}, \tag{2.12a}$$

$$\underline{\boldsymbol{b}}_u = \begin{bmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_u \\ \vdots \\ \boldsymbol{b}_u \end{bmatrix} \quad \text{and} \quad \underline{\boldsymbol{b}}_x = \begin{bmatrix} \boldsymbol{b}_x \\ \boldsymbol{b}_x \\ \vdots \\ \boldsymbol{b}_f \end{bmatrix}. \tag{2.12b}$$

2.2 Dynamics of the Robot

In this section, the model of the robot and its dynamics are introduced. The non linear system used is a robot manipulator. It consists of a chain of n-independently driven revolute joints. For each i-joint, $i = 1, 2, ..n$, we denote L_i length of i-th segment, $\boldsymbol{q}_i$ is the angle between $(i-1)$-th link and i-th link, $i = 2, 3, ..n$, except for $\boldsymbol{q}_1$ which denotes orientation of the first link to the x-axis of the base frame O-$\{x, y, z\}$ clockwise positive. The segments are assumed to be rigid bodies with a homogeneous mass distribution.

In contrast to the kinematic equations of the robot such as forward kinematics and inverse kinematics, that study the motion of the robot without taking the forces and moments into consideration, the dynamic model explicitly describes the relationship between force and motion. The modelling of holonomic mechanical systems can be effected with help of the Euler-Lagrange equations that give the analytical relationship between the joint torques and the joint positions, velocities and accelerations. In order to find these equations, the Lagrangian function $\boldsymbol{L}$ of the system needs to be computed as following

$$\boldsymbol{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{U}(\boldsymbol{q}) \tag{2.13}$$

where $\boldsymbol{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ denotes the kinetic energy and $\boldsymbol{U}(\boldsymbol{q})$ the potential energy, $\boldsymbol{q} \in \mathbb{R}^n$ and $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the joint angles vector and velocities vector, respectively. The kinetic energy can be determined using the positive definite mass matrix $\boldsymbol{M}(\boldsymbol{q})$

$$\boldsymbol{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{\boldsymbol{1}}{\boldsymbol{2}} \dot{\boldsymbol{q}} \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}}. \tag{2.14}$$

The Euler-Lagrange equation is then computed for the joints as

$$
\begin{aligned}
\boldsymbol{\tau} &= \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\partial \boldsymbol{L}}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{q}}\right) && \text{(2.15a)}\\
&= \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\partial \boldsymbol{K}}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{q}}\right) + \left(\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{q}}\right) && \text{(2.15b)}\\
&= \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\partial \boldsymbol{K}}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{q}}\right) + \boldsymbol{G}(\boldsymbol{q}) && \text{(2.15c)}\\
&= \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\boldsymbol{\partial}}{\partial \dot{\boldsymbol{q}}}\left(\frac{\boldsymbol{1}}{\boldsymbol{2}}\dot{\boldsymbol{q}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right) - \frac{\boldsymbol{\partial}}{\partial \boldsymbol{q}}\left(\frac{\boldsymbol{1}}{\boldsymbol{2}}\dot{\boldsymbol{q}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) + \boldsymbol{G}(\boldsymbol{q}) && \text{(2.15d)}\\
&= \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{\boldsymbol{1}}{\boldsymbol{2}}\begin{bmatrix} \dot{\boldsymbol{q}}^\intercal \frac{\partial \boldsymbol{M}}{\partial q_1}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}}^\intercal \frac{\partial \boldsymbol{M}}{\partial q_2}\dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^\intercal \frac{\partial \boldsymbol{M}}{\partial q_n}\dot{\boldsymbol{q}} \end{bmatrix} + \boldsymbol{G}(\boldsymbol{q}) && \text{(2.15e)}\\
&= \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) && \text{(2.15f)}
\end{aligned}
$$

where $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}$ is the inertial torque $\boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the centripetal and Coriolis torque, $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational torque and $\boldsymbol{\tau}$ is the motor torque.

2.3 Inverse Dynamics Approach

In this section, we introduce the Inverse Dynamics approach. The applied controller in this work is a linear MPC, which is appropriate for linear systems with linear constraints. Therefore, it is important to linearise the non-linear robot system at a first step. In order to solve the non-linearity, an inner loop of feedback linearisation can be used. The Inverse Dynamics approach consists of finding the joint torques $\boldsymbol{\tau}(t)$ needed to generate a specific motion. Applying this feedback linearisation in an inner loop, the non-linearities of the robot dynamic model are compensated. In this way, the non-linear MIMO system is transformed to n-decoupled linear SISO subsystems corresponding to each degree of freedom of the robot in the joint space, where n is the number of joints of the robot manipulator.
The control law of the Inverse Dynamics approach can be expressed as following

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{u} + \hat{\boldsymbol{V}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \hat{\boldsymbol{G}}(\boldsymbol{q}) \tag{2.16}$$

where $\hat{\boldsymbol{M}}(\boldsymbol{q})$, $\hat{\boldsymbol{V}}(\boldsymbol{q},\dot{\boldsymbol{q}})$ and $\hat{\boldsymbol{G}}(\boldsymbol{q})$ are estimates of the parameters of the dynamic model (2.15), $\boldsymbol{u}$ is an appropriate control law computed by the MPC controller in an outer loop. Applying the equation (2.16) into the formulation of Lagrange (2.15) yields

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \hat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{u} + \hat{\boldsymbol{V}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \hat{\boldsymbol{G}}(\boldsymbol{q}) \tag{2.17a}$$

$$\ddot{\boldsymbol{q}} = (\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q}))\boldsymbol{u} + \boldsymbol{M}^{-1}(\boldsymbol{q})[(\hat{\boldsymbol{V}}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})) + (\hat{\boldsymbol{G}}(\boldsymbol{q}) - \boldsymbol{G}(\boldsymbol{q})] \tag{2.17b}$$

$$\ddot{\boldsymbol{q}} = (\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q}))\boldsymbol{u} + \boldsymbol{\epsilon} \tag{2.17c}$$

where $\boldsymbol{\epsilon}$ is an error term. Assuming perfect estimates of the parameters, the nonlinear dynamics can be exactly compensated and the equation is reduced to

$$\ddot{\boldsymbol{q}} = \boldsymbol{u} \tag{2.18}$$

which is a linear second order system.

2.4 Model Predictive Control

In this section we introduce Model Predictive Control for linear systems. MPC uses a model of the linearised plant that represents the behaviour of the control system and it solves an online optimization problem in the prediction horizon N. This optimization problem solving is repeated at every time step k. In every iteration, the current state of the system $\boldsymbol{x}_k$ is fed back to the controller, MPC estimates a number of future states in the prediction horizon N using the model of the plant and outputs a sequence of control actions $\boldsymbol{U}_N$ that represents the optimal solution. MPC only applies the first input of the optimal sequence and disregards the rest. The same procedure is repeated at the next sampling instant with updated process measurements and a shifted horizon with one time step. The uncertainty $\boldsymbol{w}_k$ is disregarded in the model system (2.2) since standard linear MPC discussed here deals well with deterministic systems. The cost function to minimize at every time step-k is

$$J_N(\boldsymbol{x}_k, \boldsymbol{U}_N) = \sum_{i=0}^{N-1} l(\boldsymbol{x}_i, \boldsymbol{u}_i) + J_f(N, \boldsymbol{x}_k) \tag{2.19}$$

where $\boldsymbol{x}_k$ is the current state of the system, $l(\boldsymbol{x}_i, \boldsymbol{u}_i)$ is the stage cost and $J_f(N, \boldsymbol{x}_k)$ is the terminal cost.

MPC is used in this work for the trajectory tracking application. Given a reference trajectory $\boldsymbol{x}_{\text{ref}}$ and a reference input $\boldsymbol{u}_{\text{ref}}$, the optimization problem can be formulated as

$$
\begin{aligned}
\boldsymbol{U}_N^*\left(\boldsymbol{x}_k\right) = &\arg\min_{\boldsymbol{U}_N} \sum_{i=0}^{N-1} [(\boldsymbol{x}_{k+i} - \boldsymbol{x}_{k+i,\mathrm{ref}})^\top \boldsymbol{Q} (\boldsymbol{x}_{k+i} - \boldsymbol{x}_{k+i,\mathrm{ref}}) && (2.20a)\\
&+(\boldsymbol{u}_{k+i} - \boldsymbol{u}_{k+i,\mathrm{ref}})^\top \boldsymbol{R} (\boldsymbol{u}_{k+i} - \boldsymbol{u}_{k+i,\mathrm{ref}})] && (2.20b)\\
&+(\boldsymbol{x}_N - \boldsymbol{x}_{N,\mathrm{ref}})^\top \boldsymbol{Q}_\mathrm{f} (\boldsymbol{x}_N - \boldsymbol{x}_{N,\mathrm{ref}}) && (2.20c)\\
\text{s.t.}\quad & \boldsymbol{x}_{i+1} = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i, \quad i \in \{0..N-1\} && (2.20d)\\
& \boldsymbol{u}_i \in \mathcal{U}, \quad i \in \{0..N-1\} && (2.20e)\\
& \boldsymbol{x}_i \in \mathcal{X}, \quad i \in \{1..N-1\} && (2.20f)\\
& \boldsymbol{x}_N \in \mathcal{X}_\mathrm{f} && (2.20g)
\end{aligned}
$$

where $\boldsymbol{x}_{i,\mathrm{ref}}$ and $\boldsymbol{u}_{i,\mathrm{ref}}$ are the values of the reference trajectory $\boldsymbol{x}_\mathrm{ref}$ and input trajectory $\boldsymbol{u}_\mathrm{ref}$ at time step i respectively. The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weights for the states and inputs respectively and $\boldsymbol{Q}_f$ is the terminal cost and it is chosen appropriately in order to prove stability [RMD17]. The optimal solution is then $\boldsymbol{U}_N^*(\boldsymbol{x}_k) = [\boldsymbol{u}^*(0, \boldsymbol{x}_k), \boldsymbol{u}^*(1, \boldsymbol{x}_k), ..., \boldsymbol{u}^*(N-1, \boldsymbol{x}_k)]^\top$. The notation $\boldsymbol{u}^*(j, \boldsymbol{x}_k)$ denotes the optimal control input at time step-j of the computed optimal control sequence starting from state $\boldsymbol{x}_k$. Only the first component $\boldsymbol{u}^*(0, \boldsymbol{x}_k)$ is applied to the plant. The lifted state constraints can be integrated into the lifted input constraints (2.11) with help of the lifted dynamic model, i.e., (2.7)

$$
\begin{aligned}
\underline{\boldsymbol{A}}_x \left(\tilde{\boldsymbol{A}}\boldsymbol{x}_0 + \tilde{\boldsymbol{B}}\boldsymbol{U}_N\right) &\leq \underline{\boldsymbol{b}}_x && (2.21a)\\
\underline{\boldsymbol{A}}_x \tilde{\boldsymbol{B}}\boldsymbol{U}_N &\leq \underline{\boldsymbol{b}}_x - \underline{\boldsymbol{A}}_x \tilde{\boldsymbol{A}}\boldsymbol{x}_0. && (2.21b)
\end{aligned}
$$

In this way, we define the admissible input sequence set that depends on the initial state $\boldsymbol{x}_0$ as

$$
\mathcal{U}_\mathrm{ad}(\boldsymbol{x}_0) = \left\{ \boldsymbol{U}_N \;\middle|\; \begin{bmatrix} \tilde{\boldsymbol{A}}_u \\ \underline{\boldsymbol{A}}_x \tilde{\boldsymbol{B}} \end{bmatrix} \boldsymbol{U}_N \leq \begin{bmatrix} \underline{\boldsymbol{b}}_u \\ \underline{\boldsymbol{b}}_x \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \underline{\boldsymbol{A}}_x \tilde{\boldsymbol{A}} \end{bmatrix} \boldsymbol{x}_0 \right\}. \qquad (2.22)
$$

The admissible set simplifies the formulation of the optimization problem since it takes the state constraints, input constraints and system dynamics into consideration. Given the system dynamic model of the states sequence $\boldsymbol{X}_N$ as mentioned in

(2.7), the cost function can be simplified to

$$\begin{aligned}
J_N(\boldsymbol{x}_k, \boldsymbol{U}_N) =& \left(\boldsymbol{X}_N - \boldsymbol{X}_{N,ref}\right)^\top \tilde{\boldsymbol{Q}} \left(\boldsymbol{X}_N - \boldsymbol{X}_{N,ref}\right) && \text{(2.23a)}\\
&+ \left(\boldsymbol{U}_N - \boldsymbol{U}_{N,ref}\right)^\top \tilde{\boldsymbol{R}} \left(\boldsymbol{U}_N - \boldsymbol{U}_{N,ref}\right) && \text{(2.23b)}\\
=& \boldsymbol{U}_N^\top \underbrace{\left(\tilde{\boldsymbol{B}}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{B}} + \tilde{\boldsymbol{R}}\right)}_{\boldsymbol{H}} \boldsymbol{U}_N + 2 \underbrace{\left(\boldsymbol{x}_k^\top \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{B}} - \tilde{\boldsymbol{X}}_{N,ref}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{B}} - \boldsymbol{U}_{N,ref}^\top \tilde{\boldsymbol{R}}\right)}_{\boldsymbol{f}^\top} \boldsymbol{U}_N && \text{(2.23c)}\\
&+ \underbrace{\boldsymbol{x}_k^\top \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{A}} \boldsymbol{x}_k - 2\boldsymbol{x}_k^\top \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{X}}_{N,ref} + \tilde{\boldsymbol{X}}_{N,ref}^\top \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{X}}_{N,ref} + \boldsymbol{U}_{N,ref}^\top \tilde{\boldsymbol{R}} \boldsymbol{U}_{N,ref}}_{\text{const}} && \text{(2.23d)}\\
=& \boldsymbol{U}_N^\top \boldsymbol{H} \boldsymbol{U}_N + 2\boldsymbol{f}^\top \boldsymbol{U}_N + const && \text{(2.23e)}
\end{aligned}$$

with the weights matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ in the lifted form

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{Q} & & \\ & \ddots & \\ & & \boldsymbol{Q}_\mathrm{f} \end{bmatrix} \qquad \tilde{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{R} & & \\ & \ddots & \\ & & \boldsymbol{R} \end{bmatrix}. \tag{2.24}$$

The last term in (2.23) is independent of the input sequence $\boldsymbol{U}_N$ and therefore can be disregarded. The optimization problem can then be reformulated as a Quadratic Programming (QP) problem with a quadratic cost function and subject to linear constraints as following

$$\begin{aligned}
\boldsymbol{U}_N^*(\boldsymbol{x}_k) =& \arg\min_{\boldsymbol{U}_N} \boldsymbol{U}_N^\top \boldsymbol{H} \boldsymbol{U}_N + 2\boldsymbol{f}^\top \boldsymbol{U}_N && \text{(2.25a)}\\
\text{s.t.} \quad & \boldsymbol{U}_N \in \mathcal{U}_\mathrm{ad}(\boldsymbol{x}_k). && \text{(2.25b)}
\end{aligned}$$

2.5 Robust Model Predictive Control

This section introduces RMPC for linear systems. It is an extension of standard MPC in order to deal with uncertainties. RMPC uses an uncertain system that models the real behaviour of the plant described as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{G}\boldsymbol{w}_k \tag{2.26}$$

where it is assumed that the disturbance $\boldsymbol{w} \in \mathcal{W}$ is bounded. Also, it uses a nominal system described as

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{A}\bar{\boldsymbol{x}}_k + \boldsymbol{B}\bar{\boldsymbol{u}}_k \tag{2.27}$$

where the nominal system is controlled with the input $\bar{\boldsymbol{u}}_k$, which represents a control input computed under standard linear MPC. Then, the uncertain system is controlled such that it approaches the nominal system. The input of the uncertain system system is defined as a PD feedback control law

$$\boldsymbol{u}_k = \boldsymbol{K}(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) + \bar{\boldsymbol{u}}_k \tag{2.28}$$

where $\boldsymbol{K}$ is a matrix that stabilizes the closed-loop matrix $(\boldsymbol{A} + \boldsymbol{BK})$. The stabilization step of RMPC is important in order to stabilize the error between the uncertain system and the nominal system. The dynamics of the error are

$$\begin{aligned} \boldsymbol{e}_{k+1} &= \boldsymbol{x}_{k+1} - \bar{\boldsymbol{x}}_{k+1} && (2.29)\\ &= \boldsymbol{Ax}_k + \boldsymbol{Bu}_k + \boldsymbol{Gw}_k - \boldsymbol{A}\bar{\boldsymbol{x}}_k - \boldsymbol{B}\bar{\boldsymbol{u}}_k && (2.30)\\ &= \boldsymbol{Ax}_k + \boldsymbol{B}(\boldsymbol{K}(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) + \bar{\boldsymbol{u}}_k) + \boldsymbol{Gw}_k - \boldsymbol{A}\bar{\boldsymbol{x}}_k - \boldsymbol{B}\bar{\boldsymbol{u}}_k && (2.31)\\ &= (\boldsymbol{A} + \boldsymbol{BK})\boldsymbol{e}_k + \boldsymbol{Gw}_k. && (2.32) \end{aligned}$$

If the nominal system is optimized such that the states fulfill the state constraint $\bar{\boldsymbol{x}}_k \in \mathcal{X}$, it needs to be guaranteed that the states of the uncertain system also fulfill the states constraints $\boldsymbol{x}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{e}_k \in \mathcal{X}$, even if the worst case disturbance occurs. Therefore, new tightened constraints are defined for the nominal system as following

$$\begin{aligned} \bar{\boldsymbol{x}}_k &\in \mathcal{X} \ominus \mathcal{Z} && (2.33a)\\ \bar{\boldsymbol{x}}_N &\in \mathcal{X}_\mathrm{f} \ominus \mathcal{Z} && (2.33b)\\ \bar{\boldsymbol{u}}_k &\in \mathcal{U} \ominus \boldsymbol{K} \circ \mathcal{Z} && (2.33c) \end{aligned}$$

where $\mathcal{Z}$ is the invariant disturbance set. The terminal set $\mathcal{X}_\mathrm{f}$ is a control invariant set. The operations $\ominus$ and $\circ$ are Pontryagin Difference and Affine Mapping respectively. The operations are defined for convex polyhedral sets and are presented in [BBM17]. The invariant disturbance set $\mathcal{Z}$ is defined as

$$(\boldsymbol{A} + \boldsymbol{BK}) \circ \mathcal{Z} \oplus \boldsymbol{G} \circ \mathcal{W} \subseteq \mathcal{Z} \tag{2.34}$$

which is an upper-bound of the disturbance set and bounds the error $\boldsymbol{e}_k$.

Chapter 3

CVPM Approach for Robot Manipulator

The CVPM method with underlying MPC was first introduced in [BGWL20] and is referred as the CVPM-MPC approach. It handles systems that include uncertainties by introducing probabilistic constraints. This probabilistic constraint is referred to as a chance constraint $\mathcal{X}_\mathrm{p}$ and it is a convex polyhedral set for which it is guaranteed that the robot executes a collision-free motion. The method also deals with systems in which the probabilistic constraint changes suddenly over time. The approach comes with many advantages for safety-critical systems, for example a moving robot in an uncertain environment that plans to follow a reference trajectory and to avoid obstacles. A visualization of the method for robot manipulator is shown in Fig. 3.1. It represents the multi-loop control scheme. First, an inner loop of feedback linearisation based on the Inverse Dynamics approach linearises the non-linear robot manipulator. The linearised plant is then used for the CVPM-MPC optimization in an outer loop. The CVPM part is considered as a preprocessing step of the MPC and has a prediction horizon N_p. It tightens the admissible input sequence set $\mathcal{U}_{ad}$ in order to ensure the minimization of violating the constraints for the next N_p states. This set is tightened such that it considers all disturbances of the disturbance set $\mathcal{W}$. The resulting set is $\mathcal{U}_\mathrm{opt}$. It contains the input sequences that fulfill the hard constraint with zero probability if this case is possible. Otherwise, it contains the input sequence that minimizes the probability of violation. In the next step, the tightened set $\mathcal{U}_\mathrm{opt}$ is incorporated into MPC in order to optimize the control objective function discussed in Sec. 2.4. Furthermore, CVPM with underlying RMPC can also be investigated for a more robust performance of the system.

The remainder of the chapter is structured as follows. In Sec. 3.1, the proposed control scheme of MPC for the linearised robot manipulator is discussed. In Sec. 3.2, MPC is extended to CVPM-MPC approach in order to deal with uncertainties. It introduces the optimization of the violation probability problem. Section 3.3 discusses the extension of RMPC to the CVPM-RMPC method. Finally, Sec. 3.4 introduces a

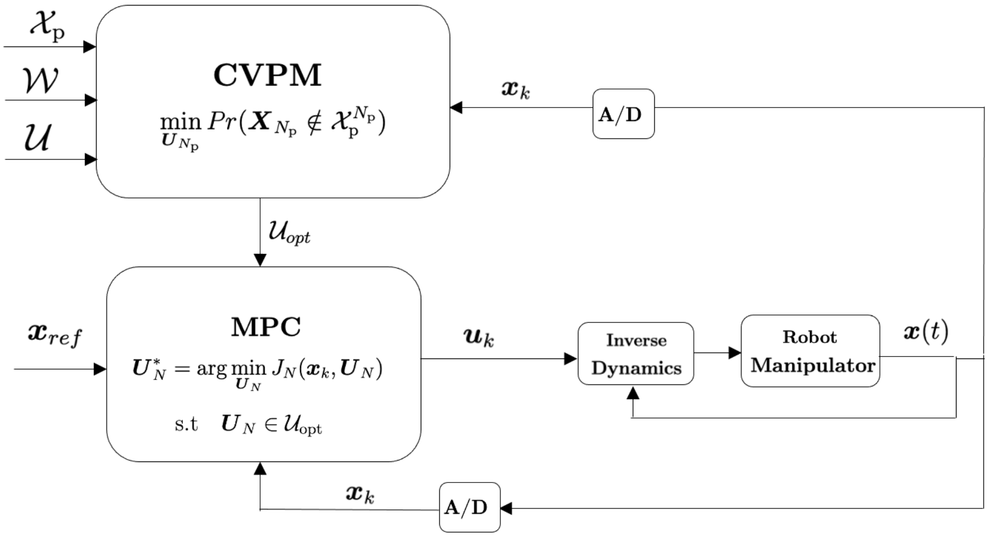

$\mathcal{X}_\mathrm{p}$
$\mathcal{W}$
$\mathcal{U}$
CVPM
$\min_{\boldsymbol{U}_{N_\mathrm{p}}} Pr(\boldsymbol{X}_{N_\mathrm{p}} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}})$
$\boldsymbol{x}_k$
A/D
$\mathcal{U}_{opt}$
$\boldsymbol{x}_{ref}$
MPC
$\boldsymbol{U}_N^* = \arg\min_{\boldsymbol{U}_N} J_N(\boldsymbol{x}_k, \boldsymbol{U}_N)$
s.t $\boldsymbol{U}_N \in \mathcal{U}_\mathrm{opt}$
$\boldsymbol{u}_k$
Inverse Dynamics
Robot Manipulator
$\boldsymbol{x}(t)$
$\boldsymbol{x}_k$
A/D

Figure 3.1: Scheme of the multi loop control of robot manipulator with an inner loop of feedback linearization and outer loop for CVPM-MPC optimization.

reformulation of the control problem from the workspace to the configuration space.

3.1 MPC for Robot Manipulator

The robot manipulator introduced in Sec. 2.2 with n-joints can be represented with the following state vector

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}, \qquad \text{with} \qquad \boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \qquad \text{and} \qquad \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{3.1}$$

representing the joints angles $\boldsymbol{q}$ and velocities $\dot{\boldsymbol{q}}$ respectively. After applying the Inverse Dynamics approach from Sec. 2.3, the MIMO system is reduced to n-SISO decoupled linear systems. Assuming that the estimation parameters of the feedback linearization are very similar to the real parameters, we can model the robot plant with a double integrator system and each SISO system is described as following

$$\dot{\boldsymbol{x}}_{\boldsymbol{i}}(t) = \begin{bmatrix} \dot{q}_i(t) \\ \ddot{q}_i(t) \end{bmatrix} \tag{3.2a}$$

$$= \boldsymbol{A}_g \begin{bmatrix} q_i(t) \\ \dot{q}_i(t) \end{bmatrix} + \boldsymbol{b}_g u_i(t) \tag{3.2b}$$

$$= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_i(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i(t) \tag{3.2c}$$

where $i \in \{1, 2, .., n\}$ and t is the time instant. The matrix $\boldsymbol{A}_g$ and the vector $\boldsymbol{b}_g$ define each decoupled continuous SISO system. The input u_i is the auxiliary control input used for feedback linearization. The model is discretized using Euler-Method with sampling time T_s. The resulting system description for the i-th joint is

$$\boldsymbol{x}_{i,k+1} = (\mathbb{I}_2 + T_s\boldsymbol{A}_g)\boldsymbol{x}_{i,k} + (T_s\boldsymbol{b}_g)u_{i,k}. \tag{3.3}$$

For example, the overall MIMO discrete-time system of a robot manipulator with 2-joints and a sampling time $T_s = 0.1$ can be formulated as following

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \boldsymbol{u}_k. \tag{3.4}$$

The linearized discrete model is used in MPC in order to make the robot follow the reference trajectory $\boldsymbol{x}_{\text{ref}}$, where MPC optimizes the cost function (2.20) with respect to the presented constraints.

3.2 CVPM-MPC for Robot Manipulator

We introduce in this section the optimization problem of CVPM and how it is incorporated into MPC. MPC is recursively feasible, if all feasible initial states feasibility is guaranteed at every step. Since the presence of uncertainties can lead to infeasibility, we extend MPC to the CVPM-MPC approach in order to deal with this issue. We consider the discrete-time system

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{G}\boldsymbol{w}_k \tag{3.5}$$

where $\boldsymbol{w}_k$ is the disturbance in the polyhedral bounded set $\mathcal{W}$, the inputs $\boldsymbol{u}_k \in \mathcal{U}$ for all time steps k. However, the state is considered to be bounded by a probabilistic constraint $\mathcal{X}_\text{p}$. This set can be changed over time and it is allowed to violate it without loss of recursive feasibility. Recursive feasibility is an important property of the approach and needs to be guaranteed. The CVPM-MPC approach is recursive feasible if the underlying MPC is recursive feasible [BGWL20]. For that reason, the state constraints are not directly considered in the MPC optimization part. The state constraints are handled by the CVPM part as probabilistic constraints in $\mathcal{X}_\text{p}$. Since MPC does not depend on these state constraints, the admissible input set is computed differently than the formulation in (2.22). Instead, the admissible input set of MPC is defined as

$$\mathcal{U}_x = \underbrace{\mathcal{U} \times \mathcal{U} \times \mathcal{U} \ldots}_{N \quad times} = \mathcal{U}^N. \tag{3.6}$$

This is the cartesian product of the input set $\mathcal{U}$ in the prediction horizon N. The CVPM part minimizes the probability that the next N_p-predicted states violate the chance constraint $\mathcal{X}_\mathrm{p}$. N_p is the horizon of the CVPM whereas as mentioned before, N is the prediction horizon of MPC. Therefore, the probability of violation can be for example minimized for only the next predicted state $\boldsymbol{x}_1$, but it can also be minimized for the next N_p-states where $0 < N_\mathrm{p} \leq N$. The violation probability to be minimized is

$$
\begin{aligned}
& \min_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr(\boldsymbol{x}_1 \notin \mathcal{X}_\mathrm{p}, \boldsymbol{x}_2 \notin \mathcal{X}_\mathrm{p}, \ldots) && \text{(3.7a)} \\
& = \min_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr \left(\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N_\mathrm{p}} \end{bmatrix} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}} \right) && \text{(3.7b)} \\
\text{s.t.} \quad & \boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}} && \text{(3.7c)}
\end{aligned}
$$

where $\boldsymbol{U}_{N_\mathrm{p}} = [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_\mathrm{p}-1}]^\top$ is the lifted input sequence. $\mathcal{U}_{x,N_\mathrm{p}}$ is the admissible input set for the CVPM optimization and is defined as

$$
\mathcal{U}_{x,N_\mathrm{p}} = \mathcal{U}^{N_\mathrm{p}}. \tag{3.8}
$$

3.2.1 Different Cases for CVPM-MPC approach

There are two cases that need to be considered in the constraint violation probability minimization. Case 1 is applied when there exists an input sequence $\boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}}$ that ensures that the probability of violation of the chance constraint $\mathcal{X}_\mathrm{p}$ is zero. If case 1 is not possible, i.e the constraint cannot be satisfied, then case 2 is applied. Case 2 computes the input sequence that minimizes the violation probability. The two cases are explained in more details in the following.

Case 1:

$$
\exists \quad \boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}} \quad \text{s.t.} \quad \Pr\left(\boldsymbol{X}_{N_\mathrm{p}} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}\right) = 0. \tag{3.9}
$$

Case 1 is applied, when it is guaranteed that the whole state sequence of the future predicted states $\boldsymbol{X}_{N_\mathrm{p}}$ is in the probabilistic set $\mathcal{X}_\mathrm{p}^{N_\mathrm{p}}$. This case is possible, since the disturbance lies in the bounded polyhedral set $\mathcal{W}$. The computed set in this case that considers all disturbances is determined as following

$$
\begin{aligned}
\mathcal{U}_{p,N_\mathrm{p}} &= \{\boldsymbol{U}_{N_p} \mid \boldsymbol{X}_{N_p} \in \mathcal{X}_\mathrm{p}^{N_p}, \quad \forall \quad \boldsymbol{W}_{Np} \in \mathcal{W}^{N_p}\} && \text{(3.10a)}\\
&= \{\boldsymbol{U}_{N_p} \mid \tilde{\boldsymbol{A}}_{N_p}\boldsymbol{x}_0 + \tilde{\boldsymbol{B}}_{N_p}\boldsymbol{U}_{N_p} + \tilde{\boldsymbol{G}}_{N_p}\boldsymbol{W}_{N_p} \in \mathcal{X}_\mathrm{p}^{N_p}, \quad \forall \quad \boldsymbol{W}_{Np} \in \mathcal{W}^{N_p}\} && \text{(3.10b)}\\
&= \{\boldsymbol{U}_{N_p} \mid \tilde{\boldsymbol{A}}_{N_p}\boldsymbol{x}_0 + \tilde{\boldsymbol{B}}_{N_p}\boldsymbol{U}_{N_p} \in \mathcal{X}_\mathrm{p}^{N_p} \ominus \tilde{\boldsymbol{G}}_{N_p} \circ \mathcal{W}^{N_p}\} && \text{(3.10c)}\\
&= \{\boldsymbol{U}_{N_p} \mid \tilde{\boldsymbol{B}}_{N_p}\boldsymbol{U}_{N_p} \in \mathcal{X}_\mathrm{p}^{N_p} \ominus \tilde{\boldsymbol{G}}_{N_p} \circ \mathcal{W}^{N_p} \oplus \{-\tilde{\boldsymbol{A}}_{N_p}\boldsymbol{x}_0\}\} && \text{(3.10d)}\\
&= \{\boldsymbol{U}_{N_p} \mid \boldsymbol{U}_{N_p} \in \left(\mathcal{X}_\mathrm{p}^{N_p} \ominus \tilde{\boldsymbol{G}}_{N_p} \circ \mathcal{W}^{N_p} \oplus \{-\tilde{\boldsymbol{A}}_{N_p}\boldsymbol{x}_0\}\right) \circ \tilde{\boldsymbol{B}}_{N_p}\} && \text{(3.10e)}\\
&= \left(\mathcal{X}_\mathrm{p}^{N_\mathrm{p}} \ominus \tilde{\boldsymbol{G}}_{N_\mathrm{p}} \circ \mathcal{W}^{N_\mathrm{p}} \oplus \{-\tilde{\boldsymbol{A}}_{N_\mathrm{p}}\boldsymbol{x}_0\}\right) \circ \tilde{\boldsymbol{B}}_{N_p}. && \text{(3.10f)}
\end{aligned}
$$

The operation $\oplus$ is Minkowski Sum defined for convex polyhedral sets in [BBM17]. In order to apply case 1, the computed set $\mathcal{U}_{p,N_\mathrm{p}}$ must be non-empty and included in the admissible input set $\mathcal{U}_{x,N_\mathrm{p}}$. For that reason it is important that

$$\mathcal{U}_\mathrm{opt} = \mathcal{U}_{p,N_\mathrm{p}} \cap \mathcal{U}_{x,N_\mathrm{p}} \neq \emptyset. \tag{3.11}$$

The obtained set $\mathcal{U}_\mathrm{opt}$ is then the input set for the MPC optimization as shown in Fig. 3.1. MPC optimizes over the set $\mathcal{U}_\mathrm{opt}$ that ensures a feasible solution and non-violation of the chance constraint. If the intersection of the two sets is empty, case 2 is considered.

Case 2:

$$\Pr\left(\boldsymbol{X}_{N_\mathrm{p}} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}\right) > 0 \quad \forall \quad \boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}}. \tag{3.12}$$

The input sequence that minimizes the violation is selected in this case. We assume that the disturbance $\boldsymbol{w}_k$, $k \in \{0, \ldots, N_\mathrm{p}-1\}$, is a random variable and it has a multivariate Gaussian distribution $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{w}})$. The lifted disturbance vector has then a distribution with zero mean and diagonal covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{W}_{\boldsymbol{N_p}}}$ with entries $\boldsymbol{\Sigma}_{\boldsymbol{w}}$. According the discrete-time system description in (3.5), the mean and covariance matrix of the state sequence can be described as

$$
\begin{aligned}
\bar{\boldsymbol{X}}_{N_\mathrm{p}} &= \tilde{\boldsymbol{A}}_{N_p}\boldsymbol{x}_0 + \tilde{\boldsymbol{B}}_{N_p}\boldsymbol{U}_{N_p} && \text{(3.13a)}\\
\boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}} &= \tilde{\boldsymbol{G}}_{N_p}\boldsymbol{\Sigma}_{\boldsymbol{W}_{\boldsymbol{N_p}}}\tilde{\boldsymbol{G}}_{N_p}^\top. && \text{(3.13b)}
\end{aligned}
$$

The optimization problem (3.7) becomes

$$
\begin{aligned}
&\min_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr(\boldsymbol{X}_{N_\mathrm{p}} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}) && \text{(3.14a)}\\
\text{s.t.}\quad &\boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}} && \text{(3.14b)}\\
&\boldsymbol{X}_{N_\mathrm{p}} \sim \mathcal{N}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}, \boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}}). && \text{(3.14c)}
\end{aligned}
$$

The solution $\boldsymbol{U}^*_{N_\mathrm{p}}$ to this optimization is a unique input sequence that optimizes the probability, i.e.,

$$\mathcal{U}_{p,N_\mathrm{p}} = \{\boldsymbol{U}^*_{N_\mathrm{p}}\} \tag{3.15}$$

and the overall constraint for the input sequence that is given to the MPC part is then

$$\mathcal{U}_\mathrm{opt} = \mathcal{U}_{p,N_\mathrm{p}} \cap \mathcal{U}_{x,N_\mathrm{p}}. \tag{3.16}$$

Different approaches for the optimization of (3.14), when case 2 is applied, are described in details in [Fin20]. We discuss briefly in the next Sec. 3.2.2 how the optimization of (3.14) is solved using QP approach.

3.2.2 Probability Optimization of Case 2

A target set $\mathcal{X}_\mathrm{T}$ is defined for the optimization of case 2 as

$$\mathcal{X}_T = \mathcal{X}_\mathrm{p}^{N_\mathrm{p}} \ominus \tilde{\boldsymbol{G}}_{N_\mathrm{p}} \circ \mathcal{W}^{N_\mathrm{p}}. \tag{3.17}$$

This is a tightened constraint set for the probability optimization of case 2 such that it considers all disturbances even when the worst case disturbance occurs. The goal of this optimization is to make the states of the robot as close as possible to the target set's nearest point in order to reach the probabilistic constraint $\mathcal{X}_\mathrm{p}$. The probability $\Pr(\boldsymbol{X}_{N_\mathrm{p}} \in \mathcal{X}_\mathrm{p}^{N_\mathrm{p}})$ denotes the probability that the whole state sequence $\boldsymbol{X}_{N_\mathrm{p}}$ lies within the probabilistic set $\mathcal{X}_\mathrm{p}$ and it can be calculated as

$$\Pr(\boldsymbol{X}_{N_\mathrm{p}} \in \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}) = \frac{1}{\sqrt{(2\pi)^{nN_\mathrm{p}} \det \boldsymbol{\Sigma}_{\boldsymbol{X}_{N_\mathrm{p}}}}} \int_{\mathcal{X}_\mathrm{p}^{N_\mathrm{p}}} \mathrm{e}^{\left(-\frac{1}{2}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}-\boldsymbol{\xi})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{X}_{N_\mathrm{p}}}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}-\boldsymbol{\xi})\right)} d\boldsymbol{\xi} \tag{3.18}$$

where $\boldsymbol{\xi}$ here is an integration variable. In order to approximate the integration with a multiplication , the probability density function for a certain $\boldsymbol{\xi} \in \mathcal{X}_T$ is multiplied with the hyper volume of the polytope $V(\mathcal{X}_T)$ and the integral is simplified to

$$\Pr(\boldsymbol{X}_{N_\mathrm{p}} \in \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}) \approx \frac{1}{\sqrt{(2\pi)^{nN_\mathrm{p}} \det \boldsymbol{\Sigma}_{\boldsymbol{X}_{N_\mathrm{p}}}}} \mathrm{e}^{\left(-\frac{1}{2}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}-\boldsymbol{\xi})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{X}_{N_\mathrm{p}}}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}-\boldsymbol{\xi})\right)} V(\mathcal{X}_T) \tag{3.19}$$

where the condition $\boldsymbol{\xi} \in \mathcal{X}_T$ must hold [Fin20].
Since the exponential function is entirely non-decreasing and the volume $V(\mathcal{X}_T)$ is independent of $\boldsymbol{U}_{N_\mathrm{p}}$, these terms can be neglected. This allows a reformulation of

the optimization of (3.14), i.e.,

$$\begin{aligned} \boldsymbol{U}_{N_\mathrm{p}}^{*} &= \arg\max_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr(\boldsymbol{X}_{N_\mathrm{p}} \in \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}) && \text{(3.20a)}\\ &= \arg\min_{\boldsymbol{U}_{N_\mathrm{p}}} (\bar{\boldsymbol{X}}_{N_\mathrm{p}} - \boldsymbol{\xi})^\top \boldsymbol{\Sigma}_{\boldsymbol{X}_{N_\mathrm{p}}}^{-1} (\bar{\boldsymbol{X}}_{N_\mathrm{p}} - \boldsymbol{\xi}) && \text{(3.20b)}\\ \text{s.t.}\quad & \boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}} && \text{(3.20c)}\\ & \boldsymbol{X}_{N_\mathrm{p}} \sim \mathcal{N}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}, \boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}}) && \text{(3.20d)}\\ & \boldsymbol{\xi} \in \mathcal{X}_T. && \text{(3.20e)} \end{aligned}$$

This is a quadratic program in which the system approaches the target set's nearest point. The distance between the mean of the state sequence $\boldsymbol{X}_{N_\mathrm{p}}$ and $\boldsymbol{\xi}$ is weighted by the inverse of matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}_{N_\mathrm{p}}}$. This weight matrix has the same role as the weight matrix $\boldsymbol{Q}$ used for MPC optimization.

3.3 CVPM-RMPC for Robot Manipulator

The CVPM with underlying RMPC approach is introduced in this section. In contrast to CVPM-MPC approach that only handles state constraints as probabilistic constraints in the CVPM optimization, CVPM-RMPC can handle both probabilistic constraints and hard constraints. The difference between probabilistic constraints and hard constraints is that the probabilistic constraints are handled by CVPM part and can be violated, whereas the hard constraints must be fulfilled. Therefore, the admissible input set defined for the underlying RMPC is

$$\mathcal{U}_x = \left\{ \boldsymbol{U}_N \in \mathbb{R}^{n_u N} \,\middle|\, \begin{array}{l} \bar{\boldsymbol{u}}_k \in \mathcal{U} \ominus \boldsymbol{K} \circ \mathcal{Z} \\ \bar{\boldsymbol{x}}_k \in \mathcal{X} \ominus \mathcal{Z} \\ \bar{\boldsymbol{x}}_N \in \mathcal{X}_f \ominus \mathcal{Z} \end{array} \right\}. \tag{3.21}$$

This is a set that considers the hard constraints and the disturbance invariant set contrarily to the admissible input set of the underlying MPC in Sec. 3.2. The admissible input set used for the CVPM optimization is the projection of the admissible set $\mathcal{U}_x$ in the domain of the sequence $\boldsymbol{U}_{N_\mathrm{p}}$ as defined in [BBM17]. The CVPM optimization guarantees that the nominal state sequence $\bar{\boldsymbol{X}}_{N_\mathrm{p}}$ does not violate a tightened version of the probabilistic set, in order to guarantee that the real state sequence of the uncertain system $\boldsymbol{X}_{N_\mathrm{p}}$ does not violate the probabilistic set $\mathcal{X}_\mathrm{p}$. Therefore, the violation probability to be minimized is

$$\min_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr\left(\boldsymbol{X}_{N_\mathrm{p}} \notin \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}\right) \Leftrightarrow \min_{\boldsymbol{U}_{N_\mathrm{p}}} \Pr\left(\bar{\boldsymbol{X}}_{N_\mathrm{p}} \notin (\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_\mathrm{p}}\right) \tag{3.22}$$

subject to $\boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}}$. Case 1 is considers all possible disturbances and is applied when every state in the state sequence $\boldsymbol{X}_{N_\mathrm{p}}$ reaches the tightened set $(\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})$.

The computed set that includes all input sequences that guarantee zero-violation of the probabilistic constraints is

$$\begin{aligned}
\mathcal{U}_{p,N_\mathrm{p}} &= \{\boldsymbol{U}_{N_p} \mid \bar{\boldsymbol{X}}_{N_p} \in (\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_p}\} && \text{(3.23a)}\\
&= \{\boldsymbol{U}_{N_\mathrm{p}} \mid \tilde{\boldsymbol{A}}_{N_p}\bar{\boldsymbol{x}}_0 + \tilde{\boldsymbol{B}}_{N_\mathrm{p}}\boldsymbol{U}_{N_\mathrm{p}} \in (\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_p}\} && \text{(3.23b)}\\
&= \{\boldsymbol{U}_{N_p} \mid \tilde{\boldsymbol{B}}_{N_p}\boldsymbol{U}_{N_p} \in (\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_p} \oplus \{-\tilde{\boldsymbol{A}}_{N_p}\bar{\boldsymbol{x}}_0\}\} && \text{(3.23c)}\\
&= \{\boldsymbol{U}_{N_p} \mid \boldsymbol{U}_{N_p} \in \left((\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_p} \oplus \{-\tilde{\boldsymbol{A}}_{N_p}\bar{\boldsymbol{x}}_0\}\right) \circ \tilde{\boldsymbol{B}}_{N_p}\} && \text{(3.23d)}\\
&= \left((\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_p} \oplus \{-\tilde{\boldsymbol{A}}_{N_p}\bar{\boldsymbol{x}}_0\}\right) \circ \tilde{\boldsymbol{B}}_{N_p}. && \text{(3.23e)}
\end{aligned}$$

Case 1 is applied when it is guaranteed that

$$\mathcal{U}_\mathrm{opt} = \mathcal{U}_{p,N_\mathrm{p}} \cap \mathcal{U}_{x,N_p} \neq \emptyset. \tag{3.24}$$

Otherwise, if $\mathcal{U}_\mathrm{opt}$ is empty, case 2 is considered in order to minimize the violation. The optimization problem of CVPM when case 2 is applied has the same reformulation as for the CVPM-MPC approach in Sec. 3.2, except that the target set $\mathcal{X}_T$ and the covariance matrix of the state sequence $\boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N_p}}}$ are differently computed. On one hand, the target set is a robust version of the probabilistic set $\mathcal{X}_\mathrm{p}$ and it depends on the disturbance invariant set,i.e.,

$$\mathcal{X}_\mathrm{T} = (\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})^{N_\mathrm{p}}. \tag{3.25}$$

The optimization of case 2 approaches $\mathcal{X}_T$ such that the whole state sequence $\boldsymbol{x}_{N_\mathrm{p}}$ lies within this tightened set. Therefore, the target set of each state $\boldsymbol{x}_k$ in the state sequence $\boldsymbol{X}_{N_\mathrm{p}}$ is the same and is equal to $(\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})$. On the other hand, the lifted covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}_{N_\mathrm{p}}}$ of the uncertain system is a diagonal matrix of size N_p, where the diagonal entries are the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x_1}}$. This is the covariance matrix of the first predicted state of the uncertain system and it can be calculated as following

$$\begin{aligned}
\boldsymbol{\Sigma}_{\boldsymbol{x_1}} &= \mathrm{Var}(\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{G}\boldsymbol{w}_0) && \text{(3.26a)}\\
&= \mathrm{Var}(\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}(\boldsymbol{K}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0) + \bar{\boldsymbol{u}}_0) + \boldsymbol{G}\boldsymbol{w}_0) && \text{(3.26b)}\\
&= \mathrm{Var}((\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}_0 + \boldsymbol{B}\bar{\boldsymbol{u}}_0 - \boldsymbol{B}\boldsymbol{K}\bar{\boldsymbol{x}}_0 + \boldsymbol{G}\boldsymbol{w}_0)) && \text{(3.26c)}\\
&= (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{\Sigma}_{\boldsymbol{x_0}}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^\top + \boldsymbol{G}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{G}^\top. && \text{(3.26d)}
\end{aligned}$$

In order to calculate the steady-state solution of the covariance matrix, it is assumed that $\boldsymbol{\Sigma}_{\boldsymbol{x_0}} = \boldsymbol{\Sigma}_{\boldsymbol{x_1}}$ and therefore, the covariance matrix of the first predicted state is the following Lyapunov equation

$$\boldsymbol{\Sigma}_{\boldsymbol{x_1}} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{\Sigma}_{\boldsymbol{x_1}}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^\top + \boldsymbol{G}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{G}^\top. \tag{3.27}$$

The optimization problem for the CVPM with underlying RMPC is

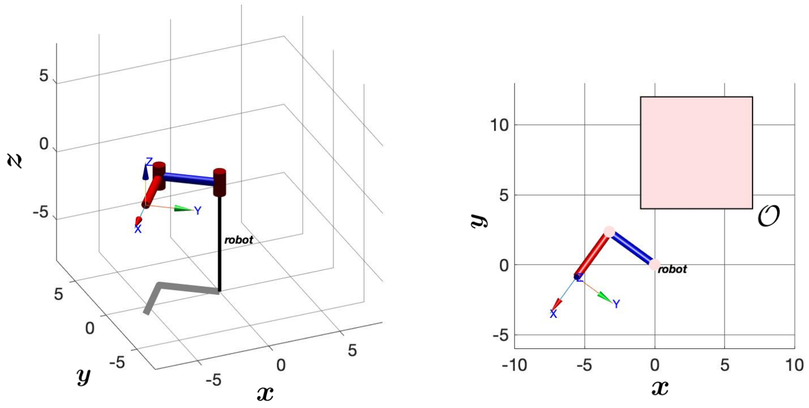
$$
\begin{aligned}
\bar{\boldsymbol{U}}^*_{N_\mathrm{p}} &= \min_{\bar{\boldsymbol{U}}_{N_\mathrm{p}}} \Pr(\boldsymbol{X}_{N_\mathrm{p}} \in \mathcal{X}_\mathrm{p}^{N_\mathrm{p}}) && \text{(3.28a)}\\
&= \min_{\boldsymbol{U}_{N_\mathrm{p}}} (\bar{\boldsymbol{X}}_{N_\mathrm{p}} - \boldsymbol{\xi})^\top \boldsymbol{\Sigma}^{-1}_{\boldsymbol{X}_{N_\mathrm{p}}} (\bar{\boldsymbol{X}}_{N_\mathrm{p}} - \boldsymbol{\xi}) && \text{(3.28b)}\\
\text{s.t.} \quad & \boldsymbol{U}_{N_\mathrm{p}} \in \mathcal{U}_{x,N_\mathrm{p}} && \text{(3.28c)}\\
& \boldsymbol{X}_{N_\mathrm{p}} \sim \mathcal{N}(\bar{\boldsymbol{X}}_{N_\mathrm{p}}, \boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}}) && \text{(3.28d)}\\
& \boldsymbol{\xi} \in \mathcal{X}_T. && \text{(3.28e)}
\end{aligned}
$$

where the nominal state sequence $\bar{\boldsymbol{X}}_{N_\mathrm{p}}$ is the mean of the state sequence $\boldsymbol{X}_{N_\mathrm{p}}$. The computed input sequence $\bar{\boldsymbol{U}}^*_{N_\mathrm{p}}$ is the input for the nominal state. Since only the first component $\bar{\boldsymbol{u}}^*_0$ of this sequence is considered, the input applied to the robot manipulator is

$$\boldsymbol{u}^*_k = \boldsymbol{K}(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) + \bar{\boldsymbol{u}}^*_0. \tag{3.29}$$

3.4 Workspace to Configuration-Space

The configuration space is a key concept for motion planning problems since it allows a complete specification of the position of every point in the robot manipulator. Therefore, it allows a point representation of the robot that captures the whole structure and is considered as an important minimal representation of the system. The CVPM approach introduced in Sec.3.2 and Sec. 3.3 is investigated in the configuration space of the robot where the state vector is $\boldsymbol{x} = [\boldsymbol{q} \quad \dot{\boldsymbol{q}}]^\top$. However, it is also important to visualize how the robot's performance is in the workspace when an obstacle occurs. Given an obstacle in the workspace, for example a space near the robot where the human moves, it needs to be guaranteed that the robot has a collision free motion. We introduce in this chapter an example that shows how the problem is reformulated from the workspace to the configuration space of the robot and how to choose a collision free set $\mathcal{X}_\mathrm{p}$ in the configuration space that must be reached to ensure safety. The transformation from workspace to configuration space is done in a grid-based manner as discussed in [CLH+05]. We consider a robot with 2-revolute joints. Both joints rotate around the z-axis. Therefore, the motion is done on the $x - y$ plane. The length of the joints is $L_1 = L_2 = 4$. A spatial view of the robot is shown in Fig. 3.2(a). In the presence of an obstacle space $\mathcal{O}$ in the workspace, the robot must avoid the collision with it. The set $\mathcal{O}$ needs to be transformed into the configuration space of the robot. The configuration space depends both on the model of the robot as well as its position with respect to the obstacle. The planar workspace is visualized in Fig. 3.2(b). The base frame is in the origin $(0, 0)$ and the set $\mathcal{O}$ is defined such that $x \in [-1, 7]$ and $y \in [4, 12]$. The shown position of the robot is $(\boldsymbol{q}_1, \boldsymbol{q}_2) = (144°, 90°)$. Initially, the configuration space $\mathcal{C}$ is an empty space of 2-dimensions, where the first dimension represents the possible



(a) spatial view

(b) planar view

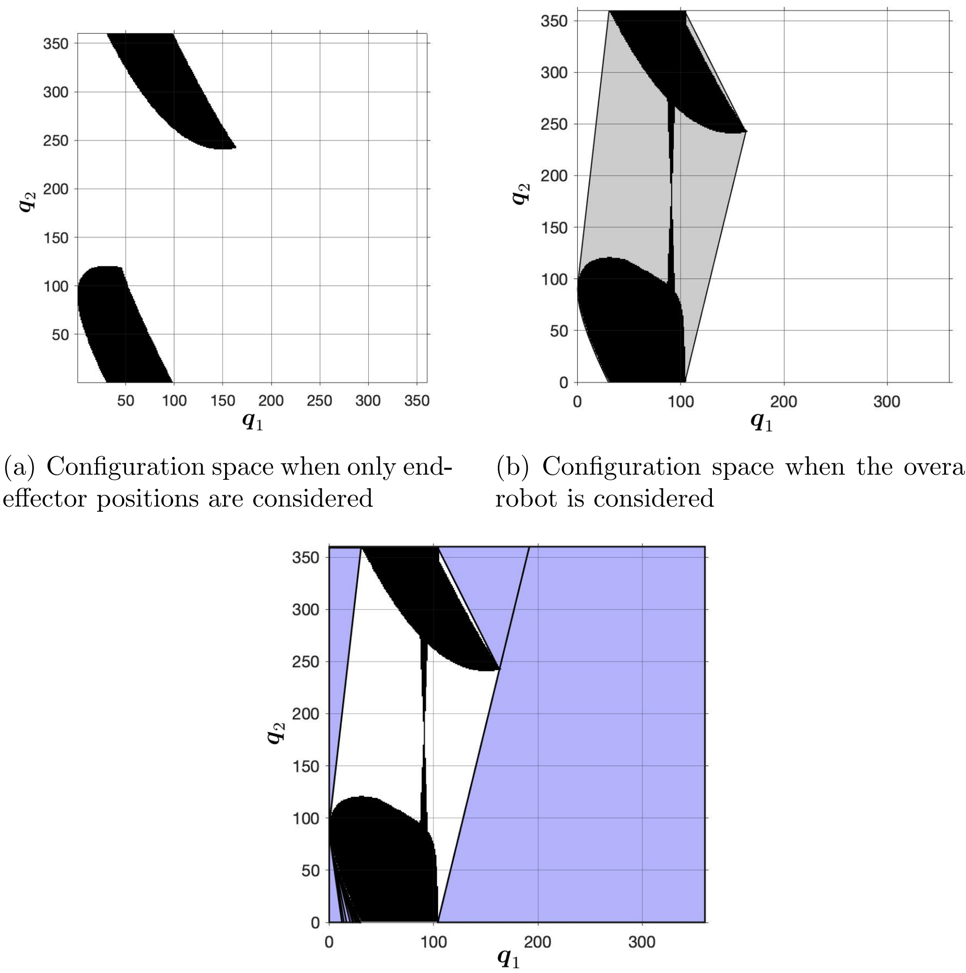
Figure 3.2: spatial and planar view of the robot

values of first joint angle $\boldsymbol{q}_1$ and the second dimension has the possible values of second joint $\boldsymbol{q}_2$. Both joints vary in the interval $[0, 360°]$. The configuration space is partitioned to a grid with a sampling interval of length $0.1°$ on the first dimension and second dimension. Every element of the grid represents a pair $(\boldsymbol{q}_{1,i}, \boldsymbol{q}_{2,i})$. The forward kinematics of the corresponding joints angle pair $(\boldsymbol{q}_{1,i}, \boldsymbol{q}_{2,i})$ are calculated to determine the position of the end-effector in the workspace,i.e.,

$$\boldsymbol{x}_i = L_1 \cos(\boldsymbol{q}_{1,i}) + L_2 \cos(\boldsymbol{q}_{1,i} + \boldsymbol{q}_{2,i}) \tag{3.30a}$$

$$\boldsymbol{y}_i = L_1 \sin(\boldsymbol{q}_{1,i}) + L_2 \sin(\boldsymbol{q}_{1,i} + \boldsymbol{q}_{2,i}). \tag{3.30b}$$

Then, it is checked whether the position $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ can collide with the set $\mathcal{O}$ in the workspace. If the obstacle is reached, the element of the grid corresponding to the joints angles is replaced with a black element. The resulting configuration space of the workspace in Fig. 3.2(b) is represented in Fig. 3.3(a). The figure represents two black surfaces that contain all possible configurations for which the end-effector of the robot reaches the set $\mathcal{O}$ in the workspace. The two black surfaces have the same shape and are symmetric. One represents the elbow-up configurations of the robot and the other one represents the elbow-down configurations. However, it is also important to consider that for a given configuration, not only the end-effector must not collide with the obstacle, but also the two joints must not collide with the set $\mathcal{O}$. Therefore, for a given configuration pair $(\boldsymbol{q}_{1,i}, \boldsymbol{q}_{2,i})$, the endeffector position $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is calculated with the forward dynamics equations (3.30) and then a collision check is done for the points of the joints. This is also done in a discrete way where every joint is partitioned into equally distant points and the collision check is done for every single point. If one point of the joint angles has a collision, the corresponding configuration pair is replaced with a black element. The resulting configuration space is shown in Fig. 3.3(b). The black surfaces that represent the collision are larger than the surfaces in Fig. 3.3(a). Also, the two surfaces are no longer



(a) Configuration space when only end-effector positions are considered

(b) Configuration space when the overall robot is considered

(c) Collision-free convex polyhedrons

Figure 3.3: Configuration space of the robot

symmetric and the lower one is larger because it is more likely that the elbow-up configurations collides with the obstacle. Since the constraint sets considered in this work are convex polyhedral sets, it is important to approximate a convex polyhedron in the free space of $\mathcal{C}$. This is the probabilistic convex set $\mathcal{X}_{\text{p}}$ and it represents all possible configurations that guarantee a collision free path of the robot. There are many possibilities to chose a convex set $\mathcal{X}_{\text{p}}$ in the free space. In this work, the choice of $\mathcal{X}_{\text{p}}$ is done using polyhedral operators of the Multi-Parametric Toolbox 3.0 (MPT3) introduced in [HKJM13] as described in the following. First, the black surfaces are enclosed with a convex polyhedron as shown is Fig. 3.3(b) in grey. The resulting set is considered as a configuration obstacle denoted as $\mathcal{O}_{\text{c}}$. Therefore, the safe set $\mathcal{X}_{\text{p}}$ must be constructed such that it is convex and it does not collide with the configuration obstacle. For this reason, we use the Set Difference operator, i.e.,

$$\mathcal{X}_{\text{safe}} = \mathcal{C} \setminus \mathcal{O}_c = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{C}, \boldsymbol{x} \notin \mathcal{O}_c\} \tag{3.31}$$

where $\mathcal{C}$ is the configuration space of bounds $[0, 360°]$, $\mathcal{O}_c$ is the configuration obstacle and $\mathcal{X}_{\text{safe}}$ is the collision-free set in $\mathcal{C}$. The set $\mathcal{X}_{\text{safe}}$ is partitioned to convex polyhedral sets, such that their union is the complete collision-free space. This can be shown in Fig. 3.3(c). It represents convex polyhedrons in the safe space of the configuration space. We choose the set $\mathcal{X}_{\text{p}}$ such that it is the largest set among the sets of $\mathcal{X}_{\text{safe}}$. The reason is that the CVPM approach defines a tightened target set $\mathcal{X}_{\text{T}}$ that tightens the probabilistic set $\mathcal{X}_{\text{p}}$ depending on the disturbance set $\mathcal{W}$ and therefore it must be guaranteed that the tightened set is non-empty.

Chapter 4

Recursive Feasibility

Recursive feasibility of MPC is a property which guarantees that if the optimization problem in (2.20) has a solution in the current step, the next step optimization is also feasible. Recursive feasibility of standard MPC is guaranteed, if the terminal set is a control invariant set [BBM17], i.e.,

$$\boldsymbol{x} \in \mathcal{X}_f \Longrightarrow \exists \boldsymbol{u} \in \mathcal{U} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \in \mathcal{X}_f. \tag{4.1}$$

This chapter investigates the recursive feasibility property for CVPM with underlying MPC in Sec. 4.1 and CVPM with underlying RMPC in Sec. 4.2.

4.1 Recursive Feasibility of CVPM with underlying MPC

The proof of recursive feasibility of CVPM-MPC approach is given if it is guaranteed that the underlying MPC is recursively feasible and also if it is guaranteed that a solution exists for the optimization of case 2.

Assumption 1. *The disturbance matrix $\boldsymbol{G}$ has linearly independent rows.*

This assumption is necessary to guarantee that the covariance matrix of the state sequence $\boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N_p}}}$ is invertible. The covariance matrix is defined as

$$\boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}} = \tilde{\boldsymbol{G}}_{N_p} \boldsymbol{\Sigma}_{\boldsymbol{W}_{\boldsymbol{N_p}}} \tilde{\boldsymbol{G}}_{N_p}^\top \tag{4.2}$$

where its inverse $\boldsymbol{\Sigma}^{-\mathbf{1}}_{\boldsymbol{X}_{\boldsymbol{N}_\mathrm{p}}}$ is used as a weight matrix for the optimization of case 2. Under Ass. 1, $\boldsymbol{G}$ has linearly independent rows. This implies that the lifted disturbance matrix $\tilde{\boldsymbol{G}}_{N_\mathrm{p}}$ defined in Sec. 2.1 also has linearly independent rows. Using the lifted covariance matrix of the disturbance $\boldsymbol{\Sigma}_{\boldsymbol{W}_{\boldsymbol{N_p}}}$ with non-zero diagonal elements, the matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}_{\boldsymbol{N_p}}}$ is invertible and can be used for the optimization of case 2.

Assumption 2. *The target set $\mathcal{X}_T$ is a non-empty set.*

The target set used for the optimization of case 2 in CVPM-MPC approach is

$$\mathcal{X}_T = \mathcal{X}_{\mathrm{p}}^{N_{\mathrm{p}}} \ominus \tilde{\boldsymbol{G}}_{N_{\mathrm{p}}} \circ \mathcal{W}^{N_{\mathrm{p}}}. \tag{4.3}$$

It is a tightened set that considers all disturbances. It represents the sets that need to be reached by every state in the prediction horizon. However, the farther the predicted state in prediction horizon is, the more constrained it becomes. An example is depicted in Fig. 4.1 that shows the probabilistic set $\mathcal{X}_{\mathrm{p}}$ where states $0 \leq \boldsymbol{x}_{1,2} \leq 4$ and horizon of CVPM is $N_{\mathrm{p}} = 10$. The set $\mathcal{X}_{\mathrm{p}}$ is the external set in the figure. The next tightened set is the set $\mathcal{X}_{\boldsymbol{x}_1}$ which represents the terminal set for the first predicted state $\boldsymbol{x}_1$ and the same applies for further predicted states with more tightened terminal set. The inner most set $\mathcal{X}_{\boldsymbol{x}_{10}}$ is the set for the last predicted state $\boldsymbol{x}_{10}$. Setting the target set as defined in (4.3), the set $\mathcal{U}_{\mathrm{opt}}$ computed by CVPM minimizes at every iteration the probability of violation and ensures that the final predicted state $\boldsymbol{x}_{10}$ is as close to tightened set $\mathcal{X}_{\boldsymbol{x}_{10}}$ as possible. A solution for the probability optimization therefore exists, when the target set $\mathcal{X}_T$ and $\mathcal{U}_{x,N_p}$ are non non-empty sets. Therefore, it is important to consider that the probabilistic set $\mathcal{X}_{\mathrm{p}}$ is large enough relatively to the disturbance set $\mathcal{W}$ in order to ensure the optimization of the states in the direction of the target set.

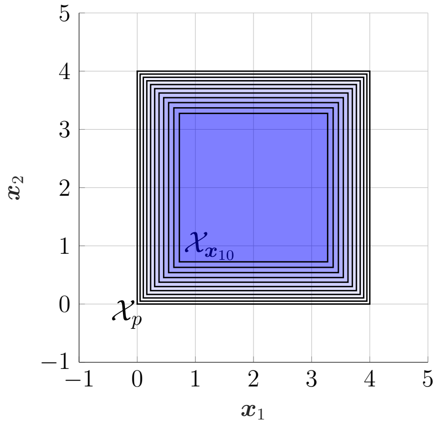


Figure 4.1: Constraint sets for predicted states in CVPM horizon N_{p} according to the tightened target set $\mathcal{X}_{\mathrm{T}}$

Assumption 3. *The underlying MPC method used for the CVPM-MPC approach is recursively feasible for all feasible initial states.*

Under Ass. 3, a feasible initial state $\boldsymbol{x}_0$ leads to a non-empty admissible input set $\mathcal{U}_x$. Recall that the admissible input set $\mathcal{U}_x$ of the underlying MPC used in Sec. 3.2 is $\mathcal{U}^N$ and it does not consider the state constraints. Therefore, $\mathcal{U}_x$ is a non-empty set. This is a necessary property to guarantee the feasibility of the overall approach.

Lemma 1. *The set $\mathcal{U}_{opt}$ used for the MPC optimization is always non empty.*

Proof. This can be proved based on the definition of the admissible input set $\mathcal{U}_{x,N_p}$ of CVPM and the admissible input set $\mathcal{U}_x$ of underlying MPC. Since $\mathcal{U}_{x,N_p} = \mathcal{U}^{N_\mathrm{p}}$ and $N_\mathrm{p} \leq N$, it is guaranteed that

$$\mathcal{U}_{x,N_p} \subseteq \mathcal{U}_x. \tag{4.4}$$

Therefore, the set $\mathcal{U}_{x,N_p}$ on which CVPM optimizes, is a non empty set. Furthermore, the computed set $\mathcal{U}_\mathrm{opt}$ must also be a non-empty because it is needed for MPC optimization. Since $\mathcal{U}_\mathrm{opt} \subseteq \mathcal{U}_{x,N_p}$, it follows that $\mathcal{U}_\mathrm{opt} \subseteq \mathcal{U}_x$. Additionally, if $\mathcal{U}_{x,N_p}$ is non-empty, the set $\mathcal{U}_\mathrm{opt}$ is always non-empty. The reason is that even if case 1 is not possible, case 2 always selects an input sequence of $\mathcal{U}_{x,N_p}$ to minimize the violation probability. □

Theorem 1. *The overall CVPM-MPC approach is recursively feasible.*

Proof. Ass. 3 is necessary to ensure that the set $\mathcal{U}_{x,N_p}$ on which CVPM optimizes is a non-empty set. Therefore, according to Lem. 1, $\mathcal{U}_\mathrm{opt}$ is also non-empty and the optimization of MPC remains feasible. Both Ass. 1 and Ass. 2 are necessary for the existence of a solution when case 2 is applied. Even if case 1 is not possible, case 2 always has a solution under these assumptions since it is a minimization problem. □

4.2 Recursive Feasibility of CVPM with underlying RMPC

This section discusses the recursive feasibility for the CVPM-RMPC approach. Same as the recursive feasibility of CVPM-MPC approach, the CVPM with underlying RMPC method is recursively feasible if the underlying RMPC is recursively feasible and the optimization problem for case 2 has a solution.

Assumption 4. *The robust input set* $\mathcal{U}_{rob} = \mathcal{U} \ominus \boldsymbol{K} \circ \mathcal{Z}$ *is a non-empty set.*

Assumption 5. *The robust state set* $\mathcal{X}_{rob} = \mathcal{X} \ominus \mathcal{Z}$ *and the robust terminal state set* $\mathcal{X}_{rob,f} = \mathcal{X}_f \ominus \mathcal{Z}$ *are a non-empty sets.*

If Ass. 4 and Ass. 5 are given, the admissible input set $\mathcal{U}_x$ of the underlying RMPC is a non-empty set. Therefore, the admissible set $\mathcal{U}_{x,N_p}$ is non-empty and the CVPM optimization is possible . The explanation for this is the same as for the CVPM-MPC approach in Sec. 4.1, since $\mathcal{U}_\mathrm{opt} \subseteq \mathcal{U}_x$. Furthermore, in order to guarantee the robustness of the underlying RMPC controller, only modelled disturbances $\boldsymbol{w} \in \mathcal{W}$ are considered. Otherwise, the real system would have a big mismatch with respect to the nominal system. Besides, it is important that the disturbance invariant set $\mathcal{Z}$ is converged to guarantee that the uncertain system does not violate the probabilistic set $\mathcal{X}_p$ even if the worst case disturbance occurs. Once it is guaranteed that the admissible input set $\mathcal{U}_{x,N_p}$ is non-empty, the assumptions for the recursive feasibility of the probability minimization problem need to be considered.

Assumption 6. *The target set $\mathcal{X}_T = (\mathcal{X}_p \ominus \mathcal{Z})^{N_p}$ is a non-empty set.*

Assumption 7. *The covariance matrix of the state sequence $\boldsymbol{\Sigma}_{\boldsymbol{X}_{N_p}}$ has full rank.*

Assumption 6 and Ass. 7 are necessary to guarantee that the optimization problem when case 2 is applied has a solution. The target set represents equal sets $(\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})$ that need to be reached by all the states of the state sequence $\boldsymbol{X}_{N_p}$.
In the CVPM-RMPC approach, it is not necessary that $\boldsymbol{G}$ has full row rank because the covariance matrix (3.26) is computed differently than the one for CVPM-MPC approach. The matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}_{N_p}}$ has an added term needed for the error correction. Therefore, it is not necessary that $\boldsymbol{G}$ has linearly independent rows as discussed in Sec. 4.1.

Theorem 2. *The CVPM-RMPC method is recursively feasible.*

Proof. Ass. 6 and Ass. 7 are necessary for the minimization of the probability violation problem when case 2 is applied. The reason for this is that according to the optimization problem of case 2 discussed in (3.28), it is necessary to consider a point $\boldsymbol{\xi} \in \mathcal{X}_\mathrm{T}$ and also to guarantee that the matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}_{N_p}}$ is invertible. Assumption 1 and Ass. 4 guarantee that the admissible set $\mathcal{U}_x$ of underlying RMPC is non-empty when considering the hard constraints. Therefore, it follows that $\mathcal{U}_\mathrm{opt} \neq \emptyset$ and the overall method remains feasible. □

Chapter 5

Simulation

The simulation of the discussed methods is shown in this chapter. The robot is a 2-revolute joints robot manipulator. The length of the segments is $L_1 = L_2 = 4$ and the mass $m_1 = m_2 = 15$. The simulation time T is 10 s and the sampling time T_s is 0.1 s. The prediction horizon N is equal to 10. The discrete model used in order to estimate the states at every optimization problem is a double integrator discrete model

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{5.1}$$

The weights for the states and inputs are defined as

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{5.2}$$

The weight $\boldsymbol{Q_f}$ of the terminal cost is computed using the function *idare* that computes the stabilizing solution of the discrete-time algebraic Riccati equation. The inputs must be within the input set $\boldsymbol{u}_k \in \mathcal{U}$ and the states must be within the state set $\boldsymbol{x}_k \in \mathcal{X}$ for all time steps k. Both sets are compact, convex and contain at least the origin. The input set is defined as convex polyhedral set, i.e.,

$$\mathcal{U} = \left\{ \boldsymbol{u} \;\middle|\; \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u} \leq \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} \right\} \tag{5.3}$$

and the state set is defined as

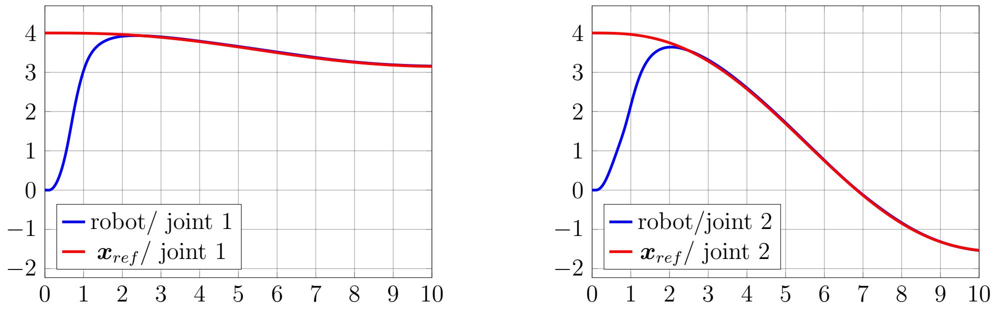
$$\mathcal{X} = \left\{ \boldsymbol{x} \;\middle|\; \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x} \leq \begin{bmatrix} 2\pi \\ 2\pi \\ 2\pi \\ 2\pi \end{bmatrix} \right\}. \tag{5.4}$$

The Inverse Dynamics are computed with the functions *itorque*, *coriolis* and *gravload* of the Serial-Link robot class of the Robotics Toolbox [Cor02]. The computation of the polyhedral sets is done with MPT3 toolbox introduced in [HKJM13].

This chapter is structured as follows. In Sec. 5.1, the simulation of linear MPC is shown for a system without uncertainties. The next Sec. 5.2 presents the results for the simulation of CVPM with underlying MPC. The simulations demonstrate how the system behaves after a sudden change of probabilistic constraints. Also, it is shown how system behaves under different disturbance sets $\mathcal{W}$ and how the discretization method can influence the computations of the approach. Section 5.3 presents the simulation results in the workspace as well as in the corresponding configuration space when the robot is controlled under CVPM-RMPC and CVPM-MPC.

Remark: Both Sec. 5.1 and Sec. 5.2 focus on the behaviour of the robot in its configuration space only. The unity of the joints angles is radians. For Sec. 5.3, the workspace including an obstacle $\mathcal{O}$ is transformed to the configuration space. We use for the simulations the degree unit for the joints angles of the robot since it allows a better presentation of the configuration space.

5.1 Simulation of Linear MPC



(a) time evolution of the first joint of the plant and the reference

(b) time evolution of the second joint of the plant and the reference

Figure 5.1: time evolution of the joints under linear MPC

The reference trajectory starts from the initial state $\boldsymbol{x}_{\text{ref},0} = [4, 4, 0, 0]^\top$ and reaches the finale state $\boldsymbol{x}_{\text{ref},100} = [\pi, -\frac{\pi}{2}, 0, 0]^\top$. The robot manipulator starts at the initial state $\boldsymbol{x}_0 = [0,0,0,0]^\top$. Figure 5.1 presents two subfigures where the blue lines describe the evolution of the first joint and second joint of the robot manipulator over time and the red lines describe the evolution of the reference trajectory. Here, only the values of the joint angles are shown. The robot reaches the reference trajectory at time step $t = 3$ s and follows it for the rest of the simulation. The simulation of this part is run in Simulink.

5.2 CVPM with underlying MPC

We simulate the CVPM with underlying MPC in this section and demonstrate how the system behaves when a sudden change of probabilistic constraint occurs. The simulation of this part is run in Matlab. The initial state of the reference trajectory as well as the initial state of the robot start at initial state $\boldsymbol{x}_{\text{ref},0} = \boldsymbol{x}_0 = [0,0,0,0]^\top$. The final state reached by the reference trajectory is $\boldsymbol{x}_{\text{ref},100} = [\frac{\pi}{4}, \frac{\pi}{4}, 0, 0]^\top$. We set the prediction horizon N_p of CVPM as well as the prediction horizon N of MPC to 10. The change of the chance constraint occurs at time $t_p = 1s$. For the time steps $0 \le t < t_p$, the probabilistic set is $\mathcal{X}_\text{p} = \mathcal{X}$ defined in (5.4) and for $t_p \le t \le 10$, the probabilistic set changes to

$$\mathcal{X}_\text{p} = \left\{ \boldsymbol{x} \;\middle|\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \boldsymbol{x} \le \begin{bmatrix} 0.5 \\ 1 \\ 2 \\ -0.5 \end{bmatrix} \right\}. \tag{5.5}$$

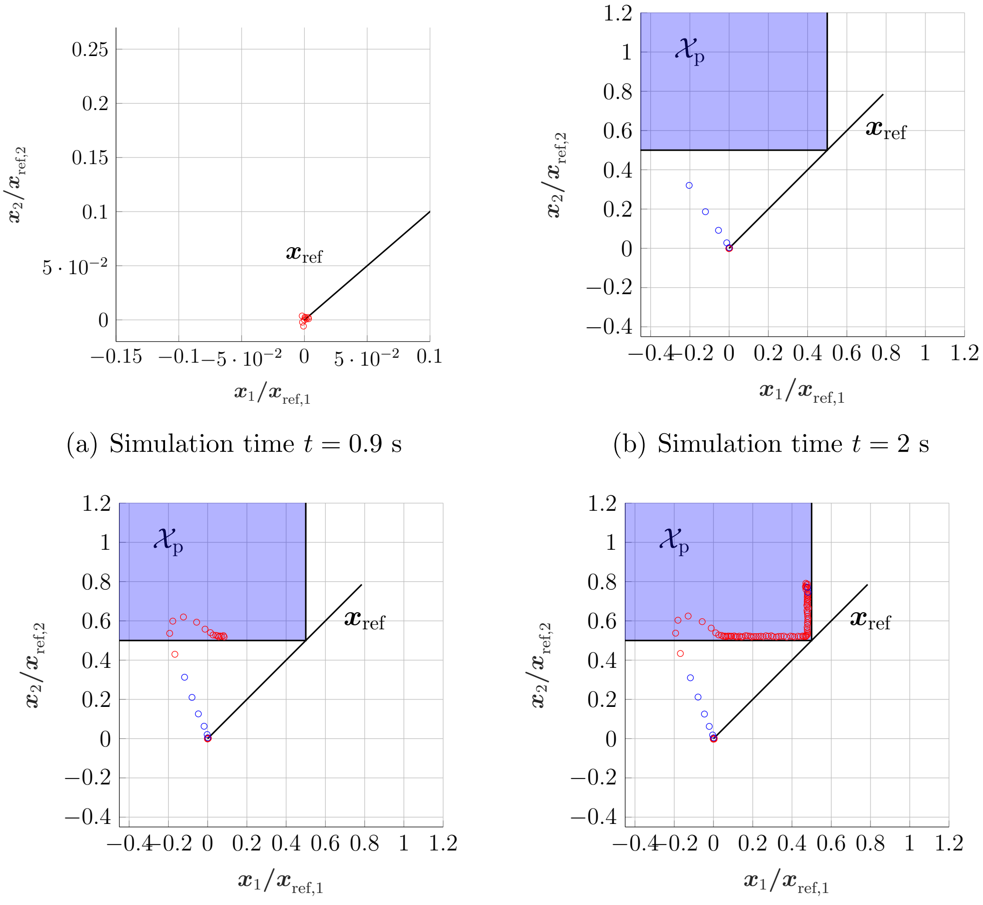
For the disturbance, we define the disturbance matrix and the covariance matrix of the disturbance as

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma}_w = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}. \tag{5.6}$$

The disturbance set $\mathcal{W}$ contains random values of a constraint box with a maximum value of 0.01,i.e.,

$$\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}^4 \mid |w_i| \le 0.01, \forall \quad i \in \{1, \dots, 4\}\}. \tag{5.7}$$

The simulation is shown in Fig. 5.2. The Fig. 5.2(a) represents the simulation at time step $t = 0.9$ s. Since the probabilistic set here is the initial set $\mathcal{X}$, all the states satisfy the probabilistic constraint and therefore case 1 is applied. The use of case 1 is denoted with red circles and the reference trajectory is a black line. At time step $t_p = 1s$, the probabilistic set changes to $\mathcal{X}_\text{p}$ as defined in (5.5). The first two dimensions of this set are shown in Fig. 5.2(b) with a blue box. The robot leaves the reference trajectory $\boldsymbol{x}_\text{ref}$ in order to reach the probabilistic set. Since the states



(a) Simulation time $t = 0.9$ s

(b) Simulation time $t = 2$ s

(c) Simulation time $t = 4$ s

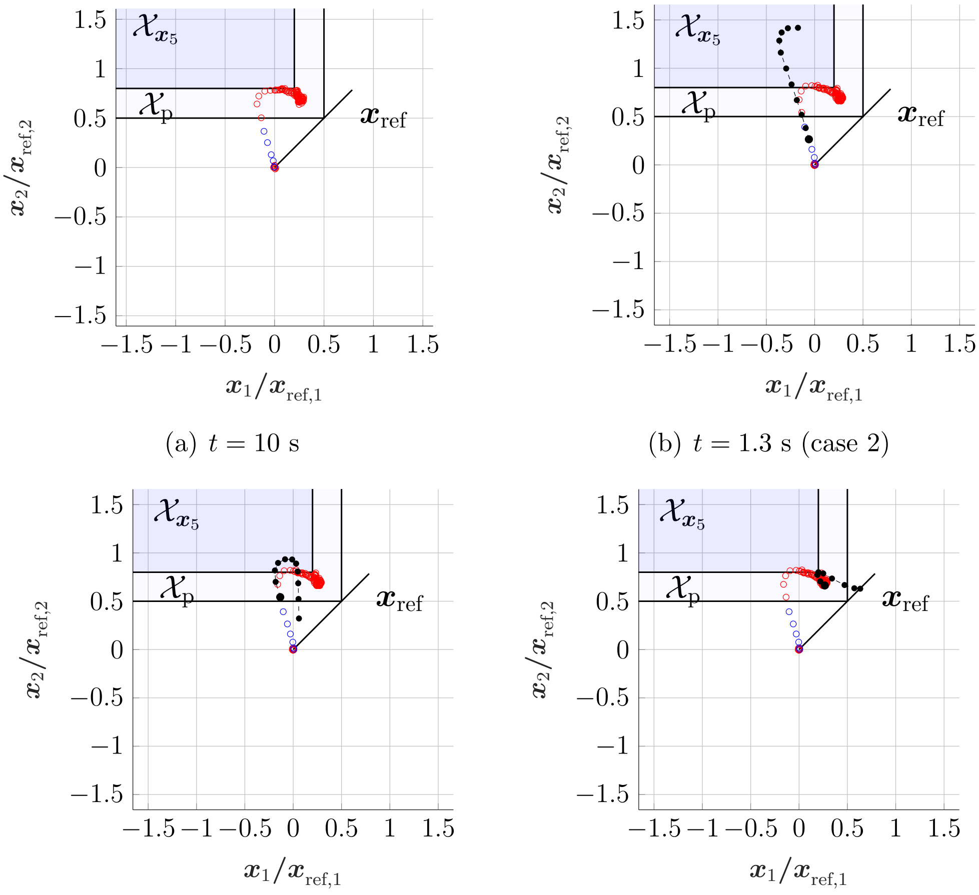
(d) Simulation time $t = 10$ s

Figure 5.2: Simulation of CVPM with underlying MPC (red: Case 1, blue : Case 2)

of the robot are not in $\mathcal{X}_{\mathrm{p}}$, case 2 is applied and it is denoted with the blue circles. When the robot reaches $\mathcal{X}_{\mathrm{p}}$, the system switches back to case 1. This is represented in Fig. 5.2(c). The states of the robot lie at the edge of the probabilistic set. The system reaches the safe set $\mathcal{X}_{\mathrm{p}}$, and then MPC computes the optimal control inputs that keep the robot as close to the reference trajectory as possible without violating the constraints again. Figure. 5.2(d) shows the simulation at the final time step. The final state of the plant is $\boldsymbol{x}_{100} = [0.5, \frac{\pi}{4}, 0.0979, 0.0079]$. The second state of the robot is equal to the second state of the reference, and the first state of the robot stays at 0.5 in order not to cross the probabilistic set edge.

5.2.1 Visualization of MPC predicted states

This part illustrates how the MPC optimization is computed depending on the used case by CVPM. Figure 5.3 represent 4 subfigures in which the future N predicted states in the MPC optimization are shown at different time instants when case 1 is applied or case 2 is applied. In this simulation, the prediction horizon of CVPM is $N_\mathrm{p} = 5$ and prediction horizon of MPC is $N = 10$. First, the probabilistic constraint set $\mathcal{X}$ of the states is a polyhedron box of bounds 2π and the disturbance set is a box of bounds 0.05. The probabilistic set changes at time $t = 1$ s to the set $\mathcal{X}_\mathrm{p}$ defined in (5.5). Figure 5.3(a) represents the states of the robot at final time $t = 10$ s. The black line is the reference trajectory $\boldsymbol{x}_\mathrm{ref}$. Case 1 is denoted with red circles and case 2 with blue circles. The external box in purple is the projection of the first two dimensions of the probabilistic set $\mathcal{X}_\mathrm{p}$. The inner most box $\mathcal{X}_{\boldsymbol{x}_5}$ in light purple is the tightened set for the last predicted state in the CVPM horizon $N_p = 5$, since the target set is also tightened and defined as $\mathcal{X}_T = \mathcal{X}_\mathrm{p}^{N_\mathrm{p}} \ominus \tilde{\boldsymbol{G}}_{N_\mathrm{p}} \circ \mathcal{W}^{N_\mathrm{p}}$. In order to apply case 1, each component of the state sequence in the CVPM horizon N_p must lie within each according tightened set. As discussed in Sec. 4.1, the further the states are in the future, the more tightened is the set they should reach. If this is not possible for all states $\boldsymbol{x}_i$ in the state sequence $\boldsymbol{X}_{N_\mathrm{p}}$, case 2 is applied



(a) $t = 10$ s

(b) $t = 1.3$ s (case 2)

(c) $t = 1.6$ s (case 1)

(d) $t = 7.3$ s (case 1)

Figure 5.3: Visualization of N predicted states of MPC depending on the applied case in CVPM ($N_\mathrm{p} = 5$, $N = 10$)

in order to optimize in the direction of the target set $\mathcal{X}_\mathrm{T}$. As mentioned in Sec. 3.2, case 2 minimizes the probability of violation and therefore the solution to this optimization in CVPM is a unique input sequence $\boldsymbol{U}^*_{N_\mathrm{p}}$. Therefore, $\mathcal{U}_\mathrm{opt} = \{\boldsymbol{U}^*_{N_\mathrm{p}}\}$ and in the next part, there is no optimization in MPC for the first N_p states, since only one input sequence exits as a candidate. Figure. 5.3(b) represents the simulation at time step $t = 1.3$ s, where case 2 is applied since the state sequence is outside $\mathcal{X}_\mathrm{p}$. It illustrates the future predicted states in the horizon $N = 10$ depicted in black circles. These estimated states lean toward the target set $\mathcal{X}_T$ to minimize probability of violation. Once the probabilistic set $\mathcal{X}_\mathrm{p}$ is reached at time $t = 1.5$ s, case 1 is applied. The set computed by CVPM does not contain one solution, but it is a set of different input sequences that guarantee zero violation of the disturbance. When zero-violation is guaranteed, MPC selects the input sequence that leans toward the reference trajectory $\boldsymbol{x}_\mathrm{ref}$. Figure. 5.3(c) represents the future predicted states in the horizon N at time $t = 1.6$ s. The Fig. 5.3(c) shows that the first 5 predicted states of MPC reach the target set. The reason is that $N_p = 5$ and this implies that the first 5 inputs of the input sequence $\boldsymbol{U}_N$ of MPC must be included in the set $\mathcal{U}_\mathrm{opt}$ that guarantees the non-violation of constraints. The other 5 inputs of the input sequence are selected such that the states are as close as possible to the reference trajectory. Figure. 5.3(c) shows that the last 5 predicted states are outside of the target set and optimized in the direction of $\boldsymbol{x}_\mathrm{ref}$. The Fig. 5.3(d) is another visualization of the estimated states at time $t = 7.3$ s.

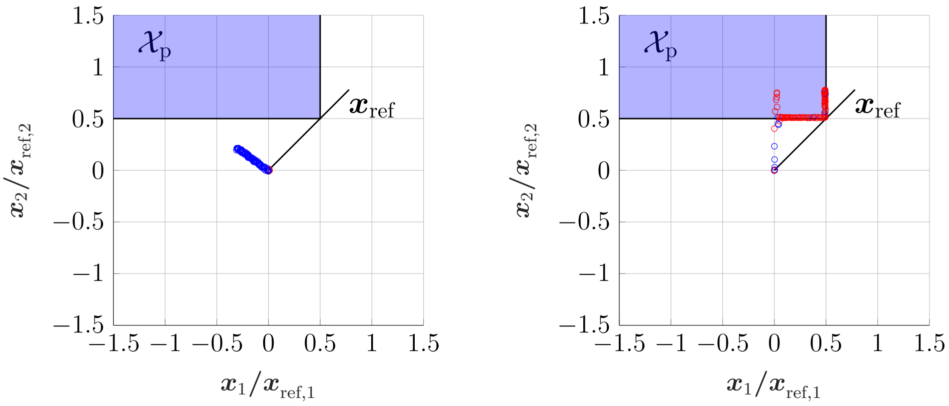
5.2.2 Discretization Method

The discretization of the double integrator model used in the previous parts is the Euler-Method discretization, as mentioned in Sec. 3.1. The discretized matrices of the system are equal to the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ presented in (5.1). This is an approximation of the discrete system. Another discretization method that can be used is Zero-Order Hold (ZOH). The discrete system matrices obtained with ZOH for a sampling time of $0.1s$ and a 2-joint robot manipulator is

$$\boldsymbol{A}_{zoh} = \begin{bmatrix} 1.1052 & 0 & 0.0111 & 0 \\ 0 & 1.1052 & 0 & 0.0111 \\ 0 & 0 & 1.1052 & 0 \\ 0 & 0 & 0 & 1.1052 \end{bmatrix} \tag{5.8a}$$

$$\boldsymbol{B}_{zoh} = \begin{bmatrix} 0.0005 & 0 \\ 0 & 0.0005 \\ 0.1052 & 0 \\ 0 & 0.1052 \end{bmatrix}. \tag{5.8b}$$

The Figure. 5.4 represents two plots at the end of simulation time $t = 10$ s using a prediction horizon of CVPM as $N_\mathrm{p} = 1$. The left Fig. 5.4(a) is the simulation using the Euler-Method discretization. After change of probabilistic set to $\mathcal{X}_\mathrm{p}$, only case 2 is applied and it is denoted with blue circles. The set $\mathcal{X}_\mathrm{p}$ is not reached, because the prediction horizon is small. This is due to a delay in the double integration using the Euler-Method. In this method, it is assumed that in a first step the acceleration changes the velocity and in the next step, the velocity changes the position of the robot. The right Fig. 5.4(b) is the simulation using ZOH discretization. The plot shows that the performance is better, since $\mathcal{X}_\mathrm{p}$ is reached after few steps. The ZOH is a more precise method, since it is assumed that the acceleration can directly have a small effect on the position of the robot. However, the exact discretization ZOH results in a slower performance. For a CVPM horizon $N_\mathrm{p} = 1$ and MPC horizon $N = 10$, the simulation duration of Euler-Method is $80s$, whereas the simulation duration using ZOH is $87s$. For a larger horizon N_p, the simulation is much slower due to the complex computations of high dimensional polyhedra. For example, for $N_\mathrm{p} = 10$, the duration of simulation with Euler-Method is $110s$ and the duration with ZOH discretization is $171s$.

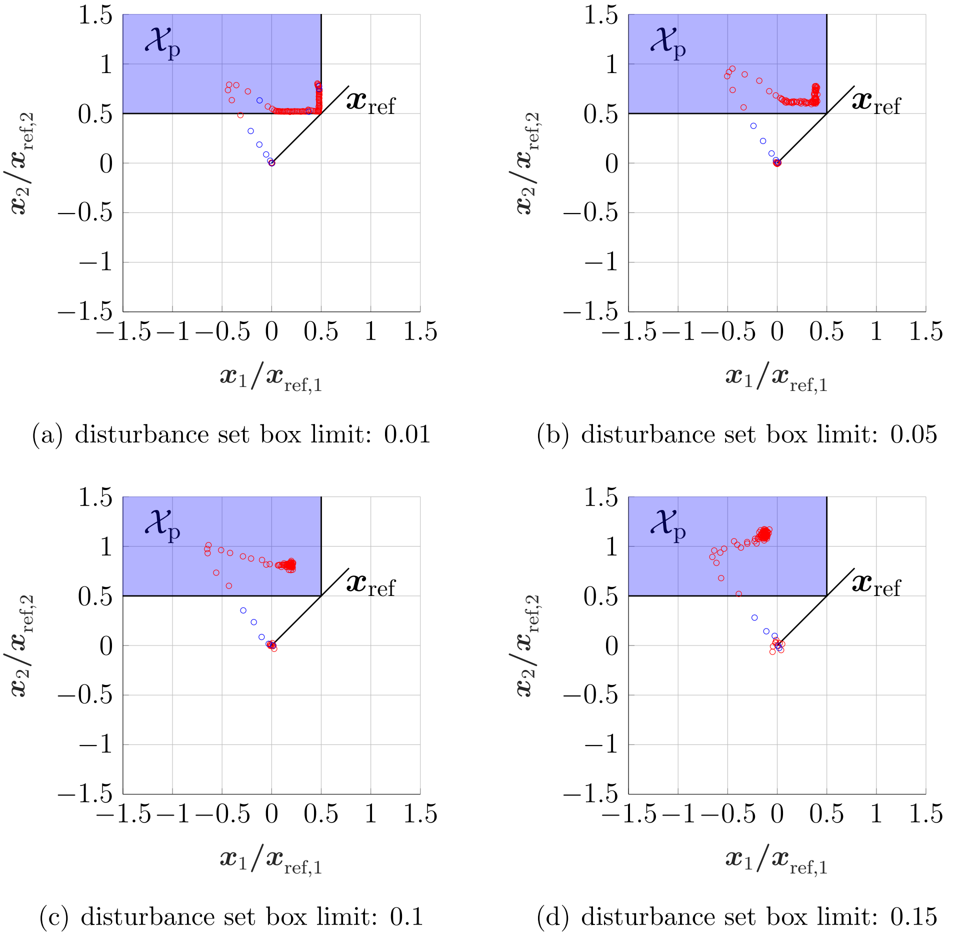


(a) Simulation at $t = 10$ s (Euler-Method) (b) Simulation at $t = 10$ s (ZOH)

Figure 5.4: Simulation of CVPM- MPC using Euler-Method and ZOH discretization methods ($N_\mathrm{p} = 1$)

5.2.3 Conservativeness of Approach

In this part, the computed solution is discussed for different disturbance sets $\mathcal{W}$. A bigger probabilistic set $\mathcal{X}_\mathrm{p}$ than the one defined in (5.5) is used in order to allow bigger disturbances and to ensure that the target set $\mathcal{X}_T = \mathcal{X}_\mathrm{p}^{N_\mathrm{p}} \ominus \tilde{\boldsymbol{G}}_{N_\mathrm{p}} \circ \mathcal{W}^{N_\mathrm{p}}$ for case 2 is non-empty. The simulation is shown in Fig. 5.5. The four simulations are presented at final time $t = 10$ s. The simulation parameters are the same for the different examples, except for the size of the disturbance set $\mathcal{W}$. The Fig. 5.5(a) is for the case where the disturbance is bounded by a constraint box with a maximum value of 0.01. The robot joints lie close to the edge of the probabilistic set $\mathcal{X}_\mathrm{p}$. For the Fig. 5.5(b), Fig. 5.5(c) and Fig. 5.5(d), the limits of the disturbance box constraint are 0.05, 0.1 and 0.15 respectively. Case 1 is only applicable if the whole



(a) disturbance set box limit: 0.01

(b) disturbance set box limit: 0.05

(c) disturbance set box limit: 0.1

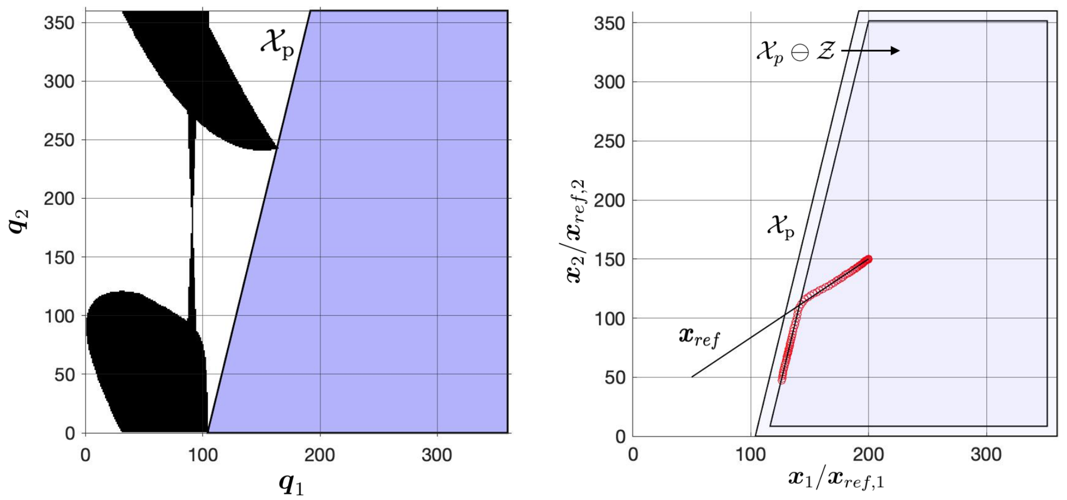
(d) disturbance set box limit: 0.15

Figure 5.5: Simulation of CVPM with underlying MPC for different disturbance sets $\mathcal{W}$ at time $t = 10$ s (red: Case 1, blue : Case 2)

state sequence $\boldsymbol{X}_{\mathrm{p}}$ is included in the set $\mathcal{X}_{\mathrm{p}}^{N_{\mathrm{p}}} \ominus \tilde{\boldsymbol{G}}_{N_{\mathrm{p}}} \circ \mathcal{W}^{N_{\mathrm{p}}}$. Therefore, the bigger the disturbance is, the more tightened the admissible input set is and the more conservative the behaviour of the robot is. The figures demonstrate that the solution of the approach is more conservative for bigger disturbances.

5.3 Workspace and Configuration Space Simulations

This section represents the simulation results of the CVPM with underlying RMPC method as well as CVPM with underlying MPC method. Additionally, it is shown how the robot moves in the configuration space as well as in the planar workspace. We use in this section an obstacle set $\mathcal{O}$ where $x \in [-1, 7]$ and $y \in [4, 12]$. The corresponding configuration space is shown in Fig. 5.6(a) where the probabilistic set $\mathcal{X}_\text{p}$ is represented with a blue set. This is the largest polyhedron constructed in the collision-free space $\mathcal{X}_\text{safe}$ as mentioned in Sec. 3.4. We consider in the following simulation that the angles of the robot have the units in degrees. The joint angles vector $\boldsymbol{q} = [\boldsymbol{q}_1 \; \boldsymbol{q}_2]^\top$ corresponds to the first two components $[\boldsymbol{x}_1, \boldsymbol{x}_2]^\top$ of the state vector. The reference trajectory in the configuration space starts at time $t = 0$ at position $\boldsymbol{x}_{\text{ref},0} = [200° \quad 150° \quad 0 \quad 0]^\top$ and ends at the final position $\boldsymbol{x}_{\text{ref},100} = [50° \quad 50° \quad 0 \quad 0]^\top$ which corresponds to a point that collides with the obstacle in the configuration space. The disturbance set $\mathcal{W}$ has 0.5 bounds and the covariance matrix of the disturbance has diagonal entries of value 0.2. In this simulation, the obstacle lies in the same position over the whole simulation time. The prediction horizons are $N = N_\text{p} = 10$.

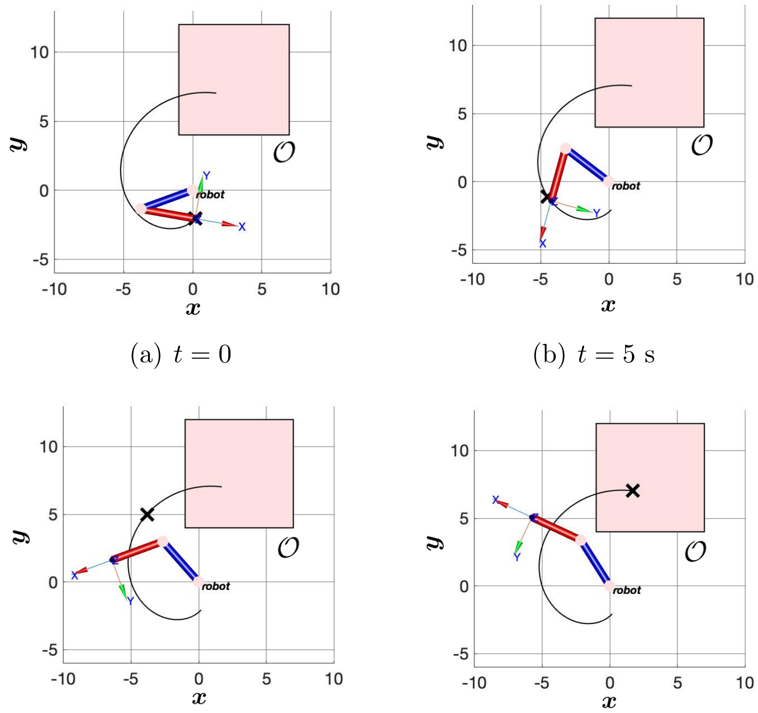


(a) Configuration space

(b) Simulation of CVPM-RMPC, $t = 10$ s

Figure 5.6: Simulation of CVPM-RMPC in the configuration space of the robot

CVPM-RMPC in configuration and work-space The behaviour of the robot in the configuration space at $t = 10$ s is shown in Fig. 5.6(b). The robot follows the reference trajectory $\boldsymbol{x}_\text{ref}$ until $t = 5$ s. Since the reference trajectory leaves the collision-free set $\mathcal{X}_\text{p}$, the robot leaves this reference and continues on the edge of the set $(\mathcal{X}_\text{p} \ominus \mathcal{Z})$. This is the target set discussed in Sec. 3.3 and it is the same set for the whole predicted states of the state sequence $\boldsymbol{X}_{N_\text{p}}$. Therefore, it is guaranteed that the robot does not collide with the obstacle. This can be demonstrated in the workspace as shown in Fig. 5.7. The four subfigures are shown in the $x - y$ plane that include the obstacle $\mathcal{O}$. The reference trajectory in the workspace is drawn with a black line. The cross point represents the current position of the reference trajectory in the workspace. Figure 5.7(a) shows the robot at the initial time step. The following Fig. 5.7(b) is the position of the robot at $t = 5$ s where the robot continues to follow the reference trajectory in a collision-free motion. Figure 5.7(c) represents the position of the robot at $t = 7$ s. It is shown that the end-point of the first joint of the robot is close to the obstacle and for further motion following the reference trajectory, a collision can occur. Therefore, for the rest of the simulation, the robot stays within the set $(\mathcal{X}_\mathrm{p} \ominus \mathcal{Z})$ in order to follow the closest configuration to the reference $\boldsymbol{x}_\mathrm{ref}$ without violating the probabilistic set $\mathcal{X}_\mathrm{p}$. The simulation at final time step is shown in Fig. 5.7(d) where the end-effector of the robot lies outside the reference trajectory and the robot remains in a collision-free position.



(a) $t = 0$

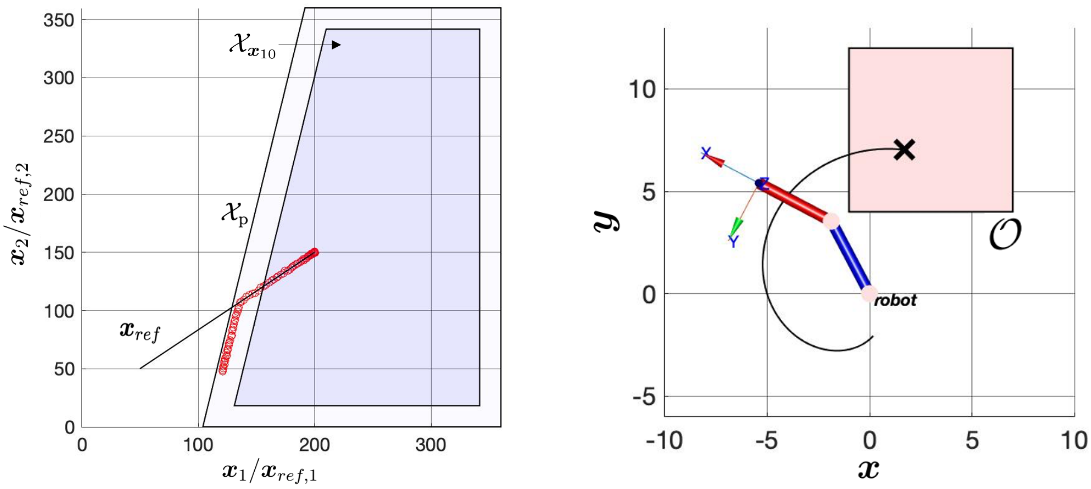
(b) $t = 5$ s

(c) $t = 7$ s

(d) $t = 10$ s

Figure 5.7: Simulation of CVPM-RMPC in the workspace of the robot

CVPM-MPC in configuration and work-space A simulation is run under CVPM-MPC with the same workspace obstacle $\mathcal{O}$, reference trajectory $\boldsymbol{x}_\mathrm{ref}$ and the same parameters defined for the controller. Figure 5.8(a) represents the configuration space of the robot at time $t = 10$ s. The robot follows the reference trajectory until it reaches the tightened target set. Therefore, the states of the robot lie closer to the bounds of the probabilistic set. This is a less robust performance compared to the CVPM-RMPC approach. The Fig. 5.8(b) shows the position of the robot at the final time step in the $x-y$ plane. The cross point represents the current position of the reference trajectory in the workspace. The simulation results show that the joints are slightly closer to the obstacle $\mathcal{O}$.



(a) configuration space

(b) Workspace

Figure 5.8: Simulation of CVPM-MPC in configuration and work-space ($t = 10$ s)

Chapter 6

Discussion

The linear MPC method uses a linear discrete model in order to make predictions for future N states. Therefore, the robot manipulator is linearized using the Inverse Dynamics approach and is reduced to n-linear, decoupled SISO systems. The plant is modelled using a double integrator discrete system of a sampling time 0.1. For a given reference trajectory $\boldsymbol{x}_{\mathrm{ref}}$, the robot states follow this trajectory under linear MPC controller. In the presence of uncertainties, MPC is extended to the CVPM-MPC approach as discussed in Chap. 3. The proposed CVPM-MPC approach deals with systems in the presence of uncertainties and also handles the change of the probabilistic set. If it is guaranteed that the whole state sequence lies within the probabilistic set $\mathcal{X}_{\mathrm{p}}$ for the next N_{p} time steps, CVPM computes the set of input sequences that ensure zero probability of violation, even if the worst case appears. Otherwise, CVPM optimizes the probability of violation for the next N_{p} states and the solution for this case is a unique input sequence that reaches the nearest point of the target set $\mathcal{X}_{\mathrm{T}}$. When case 1 is applied and the non-violation is guaranteed, MPC selects the optimal input sequence that optimizes the cost function in the direction of the reference trajectory $\boldsymbol{x}_{\mathrm{ref}}$. When case 2 is applied, the predicted states of CVPM are optimized in the direction of the target set. It is necessary that the tightened target set $\mathcal{X}_{\mathrm{T}}$ is a non-empty set, so that the optimization for case 2 is feasible. This is a drawback of the approach, because a relatively large probabilistic set $\mathcal{X}_p$ and a relatively small disturbance set $\mathcal{W}$ are necessary conditions so that the target set is not empty even for a large prediction horizon N_{p}. The discretization method used for the simulation is the Euler-Method and it is an approximation of the discrete system. This method does not perform well for a CVPM prediction horizon $N_{\mathrm{p}} = \{1, 2\}$, because the resulting discrete system has a delay that is not compensated. For $N_{\mathrm{p}} \geq 3$, the simulation is similar to the simulation using ZOH discrete model and has a smaller computation duration. The ZOH is a more precise discretization and shows better performance for $N_{\mathrm{p}} \geq 1$, but it is notably slower. Another discussed approach is the CVPM-RMPC method. In this approach, the CVPM optimization is similar to the CVPM-MPC method, except that it uses the same target set for all predicted states in the state sequence $\boldsymbol{X}_{N_{\mathrm{p}}}$ and a different covariance matrix for the states. Afterwards, depending on the tightened computed set $\mathcal{U}_{\mathrm{opt}}$, RMPC controls a nominal system under standard MPC such that the uncertain system approaches it. It is also necessary that the target set is non-empty and that the robust input set $\mathcal{U}_{\mathrm{rob}}$ and robust state set $\mathcal{X}_{\mathrm{rob}}$ are also non-empty to guarantee a robust performance under modelled disturbance. It is demonstrated how the problem of reference tracking is reformulated from the workspace to the configuration space. The obstacle set $\mathcal{O}$ is transformed to a configuration obstacle $\mathcal{O}_{\mathrm{c}}$ in which all joint angles points that cause a collision are enclosed in a convex polyhedron. It is necessary to define a convex polyhedron as a probabilistic set $\mathcal{X}_{\mathrm{p}}$ in the collision-free area. This is the largest convex polyhedron that can be constructed in the free space $\mathcal{X}_{\mathrm{safe}}$. However, building the configuration space based on a grid-decomposition is computationally expensive, especially for a robot with more degrees of freedom.

Chapter 7

Conclusion

The CVPM-MPC algorithm proposed in [BGWL20] is used in this thesis for a robot manipulator that is aimed to track a reference trajectory. The robot has constraints that are presented as polyhedral sets. The state constraints are introduced as probabilistic constraints and are only handled in the CVPM optimization. These probabilistic constraints represent a set in which a collision-free motion of the robot is guaranteed. The CVPM part is a preprocessing step that fulfils the minimization of the constraints violation. The prediction horizon of CVPM denotes the number of future predicted states for which the constraint violation is minimized. In the next step, MPC optimizes an objective function in order to get as close as possible to the reference trajectory. The method can handle a sudden change of probabilistic constraint and the approach is more conservative for big disturbances. A limitation of the method is that the probabilistic set must be large enough compared to the disturbance set. If this is not the case, the admissible input set for CVPM is empty and also the target set is empty. Therefore, a solution for the algorithm cannot be found. The CVPM is also introduced as an extension of RMPC where both probabilistic and hard constraints can be considered. In future work, the method can be investigated for a moving probabilistic constraint. This can be interpreted as a dynamic environment in which the robot moves and the obstacle changes its position. For example, when a human approaches the robot, the robot needs to avoid this time-variant obstacle and ensure a collision-free path. Besides, the approach can also be addressed for a robot with more degrees of freedom.

The CVPM-MPC and CVPM-RMPC approaches are important for safety-critical systems. The system is handled in a robust way, if it is guaranteed that there is no violation of constraints in the next step. Otherwise, it is handled in a probabilistic way in which the probability of collision is minimized.

List of Figures

Bibliography

[BBM17] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[BGWL20] Tim Brüdigam, Victor Gaßmann, Dirk Wollherr, and Marion Leibold. Minimization of constraint violation probability in model predictive control. *arXiv preprint arXiv:2006.02337*, 2020.

[CLH$^+$05] Howie M Choset, Kevin M Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, Sebastian Thrun, and Ronald C Arkin. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[Cor02] Peter Corke. Robotics toolbox. *Obtained from Peter O. Corke site: http://www. petercorke. com/Robotics% 20Toolbox. html*, 2002.

[Fin20] Michael Fink. Constraint violation probability minimization in model predictive control for linear systems and constraints. 2020.

[HKJM13] Martin Herceg, Michal Kvasnica, Colin N Jones, and Manfred Morari. Multi-parametric toolbox 3.0. In *2013 European control conference (ECC)*, pages 502–510. IEEE, 2013.

[IFM17] Gian Paolo Incremona, Antonella Ferrara, and Lalo Magni. Mpc for robot manipulators with integral sliding modes generation. *IEEE/ASME Transactions on Mechatronics*, 22(3):1299–1307, 2017.

[Mes16] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[MRRS00] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[MSR05] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[RMD17] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[SSVO10] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

License