Project: Rest-to-Rest Control of a 2-DoF Robot

Mehmet Batu Özmeteler

Abstract— **This project aims to formulate multiple model predictive control schemes for different control tasks and solve the optimal control problem of rest-to-rest control of a 2-DoF robot. The utilized dynamic model is derived from the equations of motion of the manipulator in joint space using Lagrangian formulation where the generalized coordinates are the joint angles. The dynamic model only considers inertia, Coriolis, gravity and the torques exerted by actuators as the forces acting on the robot. One attempts to demonstrate and compare a number of distinct ways to approach the MPC problem to show how certain design aspects can influence the overall performance of the controller. Furthermore, disturbances are mathematically devised to inspect both open-loop and closed-loop control while stability issues are addressed with some standing assumptions.**

I. Introduction

A. Optimal Control Theory and MPC

Optimization is an act, process, or methodology of making something (such as a design, system, or decision) as fully perfect, functional, or effective as possible. Optimization algorithms are used in many application areas like allocation of resources in logistics, investments in business, estimation and fitting of models to measurement data from experiments in science, design and operation of technical systems in engineering et. al. When optimization is discussed in engineering, model predictive control is an effective method commonly used in various industries like automotive, energy, industrial manufacturing and-so-forth.

Model predictive control (MPC) is a control algorithm that uses a model to make predictions about the future behaviour of a system in order to compute an optimal control input which optimizes an a priori defined performance criteria. It can handle constraints, has preview capability and can easily incorporate future reference information into the control problem to improve controller performance. Additionally, it can deal with multi-input multi-output (MIMO) systems that might have interactions between their inputs and outputs which is particularly difficult to achieve for PID controllers. However, it requires a powerful, fast processor with a large memory since it solves an online optimization problem at each time-step to select the optimal control action from a sequence that drives the predicted output as close to the desired reference as possible. This optimization problem is also called an optimal control problem (OCP) in literature.

The goal of optimal control is to determine an array of control signals that ensures satisfaction of physical constraints while minimizing a defined cost-functional. For the sake of clarity, one needs to provide the characterization of an OCP.

An OCP consists of four main ingredients:

- the dynamics with initial conditions $\dot{x} = f(t, x(t), u(t)),\ x(t_0) = x_0$
- the objective functional $J(t, x(\cdot), u(\cdot))$
- the class of state and input signals $x(\cdot) \in \mathbb{X},\ u(\cdot) \in \mathbb{U}$
- the equality/inequality constraints $h(x) = 0,\ g(x) \leq 0$

Solving an OCP is basically computing an infinite dimensional object $u(\cdot)$ where the constraints must be satisfied for all times. This challenge can be overcome with the help of discretization by some approaches namely: indirect or direct methods. When working with OCPs, the essential step of discretization of the continuous time representation of a system allows NLP solvers to find solutions at specified points in a time horizon. Solvers use iterative algorithms which start with initial guesses, generate further iterates that improve the performance criteria and stop when the stopping criteria is met. For unconstrained problems, the main idea behind most approaches is to choose certain directions in which the cost function decreases when moved in to seek for a local minimum as in gradient descent. Equality constrained problems can be solved via utilizing the KKT optimality conditions however, inequality constrained problems require some other techniques to be employed such as active-set methods, projection-based methods or interior-point methods.

This project solves the OCPs using direct multiple shooting which is a direct method that focuses on converting the OCP into a finite-dimensional NLP. The problem is first discretized and then optimized numerically. The general idea for multiple shooting is to divide the horizon into N control intervals where the input is parametrized, e. g. piecewise continuous. By having finitely many points, constraints are enforced at discretization points only. The system ODEs are integrated separately on N intervals from shooting nodes using the relevant input and continuity is enforced upon convergence thanks to continuity constraints. Using multiple shooting, the problem becomes larger due to the increase in the number of decision variables but sparser. The implementation is straightforward, the integration can be parallelized and also the non-linearity in the dynamics can be handled quite well which are the reasons why multiple shooting was the preferred discretization method for this project. The formulated OCPs will be discussed in the upcoming chapters thoroughly.

B. The Importance of the Dynamic Model

In mechanics, kinematics is concerned with the motion of objects without reference to the forces which cause the motion. It is purely related to the geometry of the motion while dynamics on the other hand, is concerned with the motion of bodies under the action of forces.

The forces that act on the robot are exerted by inertia, Coriolis, frictions, gravitational acceleration and the contact with the environment. The model predictive control schemes designed for the optimal control of a two-link planar arm requires an accurate model that reflects the dynamic behaviour of the robot under the mentioned varying forces. When a dynamic model is present, simulating manipulator motion allows control strategies and motion planning techniques to be tested without relying on a physically available system [1].

The analysis of the dynamic model can also be helpful for mechanical design of prototype arms. Last but not least, calculating the forces and torques that are required for the execution of a typical motion provides valuable information in designing joints, transmissions, and actuators. Hence, the derivation of a dynamic model is essential because it can be used for simulation of motion, for the analysis of the manipulator structure and especially for the design of control algorithms.

C. The Derivation of the Dynamic Model

The dynamic model of a manipulator provides a description of the relationship between the joint actuator torques and the motion of the structure. To relate joint velocities, positions, accelerations with forces and torques in mechanics, the equations of motion of the manipulator in the joint space are required. There exists 2 different methods to derive the equations of motion of a manipulator in joint space. The first method that comes to mind is the Lagrange formulation. It is a simple theoretical formulation that is easy to intuitively understand. The second method is based on Newton-Euler formulations and yields the model in a recursive format. Since it exploits the usually open structure of manipulator kinematic chains, it is computationally more efficient.

This project uses a model derived with Lagrange formulation for a number of reasons. The Newton-Euler formulation derives directly from Newton's Second Law of Motion, one that describes dynamic systems through force and momentum. The equations obtained from this method include the constraint forces acting between adjacent links. Thus, the formulation doesn't contain direct explicit relations between the joint torques and the resultant motion in terms of joint displacements.

In the Lagrangian formulation, on the other hand, the system's dynamic behaviour is described in terms of work and energy. The equations of motion can be derived in a methodical way independently of the reference coordinate frame using generalized coordinates. The resultant equations are generally compact and provide a closed-form expression. Furthermore, the derivation is simpler and more systematic than in the Newton-Euler method.

D. The Lagrange Formulation

Lagrange formulation is a reformulation of classical mechanics where the trajectory of a body is derived from solving Lagrangian equations. The main idea is to transform the mechanical system onto a set of generalized coordinates which coincide with the degrees of freedom of the mechanical structure.

$$q = [q_1, ..., q_n], \quad \dot{q} = [\dot{q_1}, ..., \dot{q_n}] \tag{1}$$

Degrees of freedom of a kinematic chain is the number of parameters (joint variables) that define the configuration of the chain. Then, the kinetic and potential energy is expressed as a function of the generalized coordinates.

$$T(q_i, \dot{q_i}), \quad U(q_i, \dot{q_i}) \tag{2}$$

These expressions form the Lagrangian equations which can be solved to obtain the trajectory of the moving body.

$$\begin{gathered} L(q_i, \dot{q_i}) = T(q_i, \dot{q_i}) - U(q_i, \dot{q_i}) \\ \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L(q_i, \dot{q_i})}{\partial \dot{q_i}} - \frac{\partial L(q_i, \dot{q_i})}{\partial q_i} = Q_i, \quad i = 1, ..., n \end{gathered} \tag{3}$$

Following this reformulation, after some arithmetic and algebraic manipulation, the dynamic model can be obtained in a closed-form. In this project, the dynamic model only considers inertia, Coriolis, gravity and the torques exerted by actuators as the forces acting on the robot.

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u \tag{4}$$

E. 2 DoF Robot

Computing optimal input trajectories for robots to reach a goal position while spending minimal energy or time is a frequently encountered control task in the industry. In this project, the considered model is a two link planar arm. The optimal control of this manipulator will be done by a model predictive control scheme which requires a dynamic model to calculate optimal input trajectories for the actuators. The actuators are servo drives which are located on the joints. The motors apply a certain commanded torque that rotates the joint which results in the motion of the end-effector.

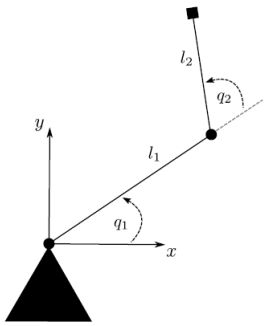

Fig. 1. 2 DoF Robot

In robotics, the position and orientation of the end-effector frame is expressed by spatial transformations which are mathematical transformations to indicate the relative position and orientation of a certain frame w.r.t. the reference frame. The transformation consists of 2 main elements: translation and rotation. There are multiple representations of this transformation however, the most commonly used one where both these elements are represented in a single form is the homogeneous transformation with homogeneous matrices.

The forward kinematics aims to compute the pose and orientation of the end-effector as a function of the joint variables. The utilized dynamic model is derived from the equations of motion of the manipulator in joint space where the generalized coordinates are the joint angles. The homogeneous transformation matrix for the 2-DoF Robot is formulated as:

$$\mathrm{T}_e^b = \begin{bmatrix} n_e^b & s_e^b & a_e^b & p_e^b \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & s_{12} & c_{12} & (l_1 \cdot c_1 + l_2 \cdot c_{12}) \\ 0 & -c_{12} & s_{12} & (l_1 \cdot s_1 + l_2 \cdot s_{12}) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

By using the homogeneous transformation matrix to convert an initial frame into a final frame, one can extract the final pose of the end-effector frame in x-y coordinates by inputting joint angles. Since this transformation aids to shift from one coordinate system to the other, choosing joint angles as the generalized coordinates suits well. In other words, moving in x-y-plane is also simply rotating in z plane for some degrees of q_1 and q_2.

II. Mathematical Preliminaries

A. Reformulation of the Dynamic Model

The dynamic model is an implicit system of second-order differential equations. In order to apply control theory and have a standard notation, one needs to reformulate the dynamic model as an explicit first-order system in state-space form $\dot{x} = f(x,u)$.

$$\ddot{q} = B^{-1}(q)(u - C(q,\dot{q})\dot{q} - G(q)) \tag{6}$$

The system of equations is expressed as an explicit first-order system by including 2 extra states for the derivatives to eliminate the double derivative. An appropriate state and input vector are selected as follows:

$$\begin{aligned} x \in \mathbb{X} \subset \mathbb{R}^4 : [q_1, q_2, \dot{q}_1, \dot{q}_2] \\ u \in \mathbb{U} \subset \mathbb{R}^2 : [\tau_1, \tau_2] \end{aligned} \tag{7}$$

The system matrices are:

$$\begin{aligned} B(q) &= \begin{bmatrix} b_1 + b_2 \cdot cos(q_2) & b_3 + b_4 \cdot cos(q_2) \\ b_3 + b_4 \cdot cos(q_2) & b_5 \end{bmatrix} \\ C(q,\dot{q}) &= -c_1 \cdot sin(q_2) \begin{bmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix} \\ G(q) &= \begin{bmatrix} g_1 \cdot cos(q_1) + g_2 \cdot cos(q_1 + q_2) \\ g_2 \cdot cos(q_1 + q_2) \end{bmatrix} \end{aligned} \tag{8}$$

The model parameters are illustrated in Table I.

TABLE I
DYNAMIC MODEL PARAMETERS

Parameter	Value	Unit
b_1	200.0	[kg m^2 / rad]
b_2	50.0	[kg m^2 / rad]
b_3	23.5	[kg m^2 / rad]
b_4	25.0	[kg m^2 / rad]
b_5	122.5	[kg m^2 / rad]
c_1	-25.0	[Nms^{-2}]
g_1	784.8	[Nm]
g_2	245.3	[Nm]
l_1	0.5	[m]
l_2	0.5	[m]

The system of ODEs for the derivatives of each state becomes:

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \mathrm{B}^{-1}(x_2) \left(\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - C(x_2, x_3, x_4) \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} - G(x_1, x_2) \right) \end{aligned} \tag{9}$$

III. Problem Statement and Implementation

The problems considered in this work search for the optimal control input trajectories $u(\cdot)$ for a robot with a closed kinematic chain of two rigid bodies connected with two revolute joints, also known as the two-link planar arm. Multiple OCP formulations that achieve setpoint stabilization while spending fixed-time, minimal time or consuming minimal control energy are presented in the next subsections. Nominal NMPC setting was assumed for the numerical simulations. The analysis of the design aspects regarding stability, recursive feasibility and optimality will be addressed subsequently.

A. Fixed-End-Time OCP Formulations

The usual methodology followed to formulate an OCP which in turn will achieve setpoint stabilization while keeping the NMPC scheme stable is based on a terminal cost function and a terminal constraint region. In this subsection, two distinct OCPs are formulated for the control task at hand where one aims to stabilize the NMPC scheme using a zero-terminal constraint. The latter accomplishes stabilization using a scaled terminal penalty and avoids terminal constraints. The continuous-time OCP formulation is presented below as a starting point to build upon for both OCPs.

$$\begin{aligned} \min_{u(\cdot)} & \int_{t=t_0}^{t_f} \ell(x(t), u(t))\, dt + V_f(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(x(t),\ u(t)), \quad x(t_0) = x_0 \\ & x \in \mathbb{X}, \quad u \in \mathbb{U} \\ & t \in [t_0,\ t_f] \end{aligned} \tag{10}$$

where $\ell(x(t), u(t))$ is the stage cost and $V_f(x(t_f))$ is the terminal cost function.

The goal for the robot is to reach the upper end position $(\frac{\pi}{2},0,0,0)$ when starting from the point $(-5,-4,0,0)$ in a fixed-time horizon. Optimal control trajectories are obtained by minimizing a quadratic stage cost ℓ. In order to steer the system as close as possible to its destination x_{goal} during the control horizon $[t_0,t_f]$, the robot's objective is formulated as:

$$\ell(x,u) = \sum_{k=0}^{N-1} (x(k)-x_{goal})^\top Q(x(k)-x_{goal}) + u(k)^\top Ru(k) \tag{11}$$

where $Q \succ 0,\ R \succ 0$. This convex form for the stage cost is commonly used for MPC applications. By including the $u^T Ru$ term, one aims to avoid singular arcs on the optimal control trajectories. The dynamics are non-linear hence, the optimization problem is non-convex due to equality constraints not being affine. As mentioned before, the OCP is discretized using direct multiple-shooting and the finite-horizon OCP formulation with a zero-terminal constraint is obtained as such:

$$\begin{aligned}
\min_{u(\cdot),\,s(\cdot)} \ & \sum_{k=0}^{N-1} (x(k)-x_{goal})^\top Q(x(k)-x_{goal}) + u(k)^\top Ru(k) \\
\text{s.t.} \quad & x(k+1) = s(k) + \int_{t_k}^{t_{k+1}} f(s(t),u(t))\ dt, \quad x(0) = x_0 \\
& x(k) = s(k), \quad k \in [1\ldots,\ N-1] \\
& x_{min} \le x(k) \le x_{max} \\
& u_{min} \le u(k) \le u_{max} \\
& x(N) = x_{goal} \\
& Q \succ 0,\ R \succ 0 \\
& k \in [0,1\ldots,\ N]
\end{aligned} \tag{12}$$

where N is the length of the prediction horizon, s is the additional decision variable for the continuity constraints and Q,R and P are the tuning matrices. The box constraints are determined by the lower and upper bounds for the states and inputs x_{min}, x_{max}, u_{min}, u_{max} which are visualized below.

$$\vec{x}_{min} = \begin{bmatrix} -\infty \\ -\infty \\ -3\pi/2 \\ -3\pi/2 \end{bmatrix}, \quad \vec{x}_{max} = \begin{bmatrix} \infty \\ \infty \\ 3\pi/2 \\ 3\pi/2 \end{bmatrix} \tag{13}$$

$$\vec{u}_{min} = \begin{bmatrix} -1000 \\ -1000 \end{bmatrix}, \quad \vec{u}_{max} = \begin{bmatrix} 1000 \\ 1000 \end{bmatrix} \tag{14}$$

For the following OCP formulation, one chooses a terminal penalty of the quadratic form to be a local Lyapunov function in the terminal region in order to stabilize the NMPC scheme.

$$V_f(x(t_f)) = (x(t_f)-x_{goal})^\top P(x(t_f)-x_{goal}) \tag{15}$$

Taking this into account, one comes up with the consequent OCP formulation.

$$\begin{aligned}
\min_{u(\cdot),\,s(\cdot)} \ & \sum_{k=0}^{N-1} (x(k)-x_{goal})^\top Q(x(k)-x_{goal}) + u(k)^\top Ru(k) \\
& + (x(N)-x_{goal})^\top P(x(N)-x_{goal}) \\
\text{s.t.} \quad & x(k+1) = s(k) + \int_{t_k}^{t_{k+1}} f(s(t),u(t))\ dt, \quad x(0) = x_0 \\
& x(k) = s(k), \quad k \in [1\ldots,\ N-1] \\
& x_{min} \le x(k) \le x_{max} \\
& u_{min} \le u(k) \le u_{max} \\
& Q \succ 0,\ R \succ 0,\ P \succ 0 \\
& k \in [0,1\ldots,\ N]
\end{aligned} \tag{16}$$

Table II contains the values of parameters used for the numerical simulation of both OCPs.

TABLE II
FIXED-TIME OCP FORMULATION PARAMETERS

Parameter	Values	Description
t_0	0	start time [s]
t_f	3	end time [s]
wQ	10	weight of Q matrix
wR	0.1	weight of R matrix
wP	10^8	weight of P matrix
h	0.05 or 0.10	integration step size [s]

Both OCPs [12], [16] were solved numerically with different number of sampling points and integrators using CasADi toolbox. The resulting trajectories from the open-loop control of the robot are illustrated in the ensuing graphs.

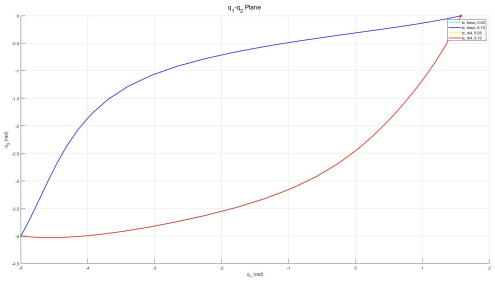

Fig. 2. OCP [12] in q_1-q_2-plane (Δt, integrator type)

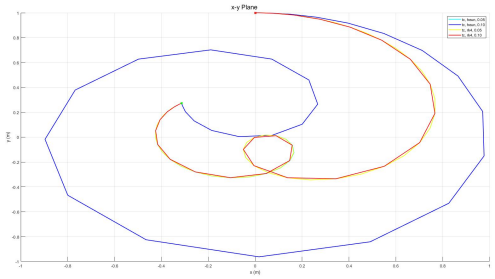

Fig. 3. OCP [12] in x-y-plane (Δt, integrator type)

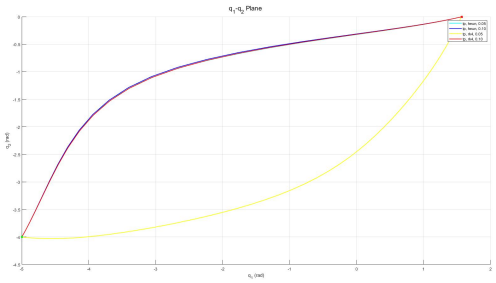

Fig. 4. OCP [16] in q_1-q_2-plane (Δt, integrator type)

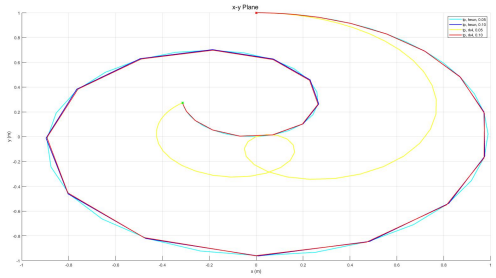

Fig. 5. OCP [16] in x-y-plane (Δt, integrator type)

From the plots, one can clearly see that both methods achieve setpoint stabilization with quite similar trajectories. However, using different integration schemes e.g. Runge-Kutta, Heun yields a different result for both OCPs. With a larger number of sampling points, the trajectories become smoother and numerically more accurate. Overall, the accuracy for the RK4 integration scheme is better and yields decent outcomes. But one can argue that Heun integrator outperforms RK4 because the computation time is significantly smaller while the accuracy is relatively comparable when it is employed at smaller sampling times.

B. Free-end-time OCP Formulations

In this subsection, a free-end-time OCP formulation of the same control task will be provided. One begins with introducing a new time variable τ to transform the time horizon.

$$t = \tau\, t_f : t \in [0, t_f] \to \tau = \frac{t}{t_f} \in [0,1] \tag{17}$$

For the time transformation, one denotes

$$\begin{aligned} x(t) &= x(t(\tau)) \\ \dot{x} = \frac{dx(t)}{dt} &= f(t, x(t), u(t)) \\ \frac{dx(t(\tau))}{d\tau} &= \frac{\partial x(t)}{\partial t} \cdot \frac{\partial t}{\partial \tau} \end{aligned} \tag{18}$$

In τ-time, the dynamics become

$$\frac{dx(\tau)}{d\tau} = t_f\, f(\tau, x(\tau), u(\tau)), \quad x(0) = x_0 \tag{19}$$

By including the free-end-time t_f as an extra decision variable, one arrives to the succeeding OCP formulation.

$$\begin{aligned} &\min_{u(\cdot),\, s(\cdot),\, t_f} \quad t_f \\ \text{s. t.} \quad & x(k+1) = s(k) + \int_{t_k}^{t_{k+1}} f(s(t), u(t))\; dt, \quad x(0) = x_0 \\ & x(k) = s(k), \quad k \in [1\ldots,\ N-1] \\ & x_{min} \le x(k) \le x_{max} \\ & u_{min} \le u(k) \le u_{max} \\ & x(t_f) = x_{goal} \\ & k \in [0, 1\ldots,\ N] \end{aligned} \tag{20}$$

Besides, the objective function for the OCP [20] can be chosen as the following to have a compromise between minimal time and minimal control energy in the performance criteria.

$$\ell(x,u) = \sum_{k=0}^{N-1} u(k)^\top R u(k) + w_T t_f \tag{21}$$

where $w_T = 10^6$ and $w_R = 1$. This results in the OCP formulation below.

$$\begin{aligned} &\min_{u(\cdot),\, s(\cdot),\, t_f} \sum_{k=0}^{N-1} u(k)^\top R u(k) + w_T t_f \\ \text{s. t.} \quad & x(k+1) = s(k) + \int_{t_k}^{t_{k+1}} f(s(t), u(t))\; dt, \quad x(0) = x_0 \\ & x(k) = s(k), \quad k \in [1\ldots,\ N-1] \\ & x_{min} \le x(k) \le x_{max} \\ & u_{min} \le u(k) \le u_{max} \\ & x(t_f) = x_{goal} \\ & k \in [0, 1\ldots,\ N] \end{aligned} \tag{22}$$

A numerical simulation of the OCP [12], [20] and [22] with Heun integrator using 100 sampling points over the control horizon was performed. The resulting trajectories from the open-loop control of the robot are shown in the upcoming graphs.

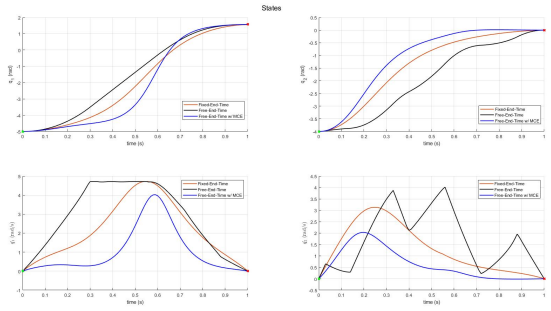

Fig. 6. States for OCP [12] vs. OCP [20] vs. OCP [22]

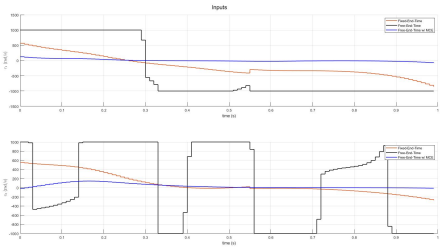

Fig. 7. Inputs for OCP [12] vs. OCP [20] vs. OCP [22]

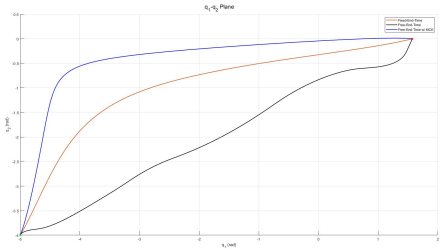

Fig. 8. OCP [12] vs. OCP [20] vs. OCP [22] in q_1-q_2-plane

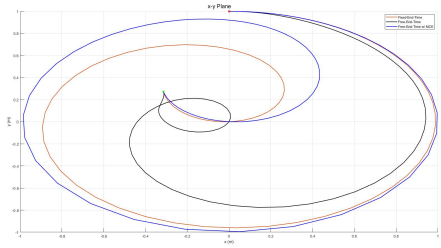

Fig. 9. OCP [12] vs. OCP [20] vs. OCP [22] in x-y-plane

The plots compare the states, inputs and the trajectories in q_1-q_2 and x-y planes. The goal positions are successfully reached for all OCPs and the minimal time for the robot to arrive at the upper end position for OCP [20] is 1.506 seconds. The time when the robot reaches the goal position for OCP [22] is 2.502 seconds. Applied inputs are mainly the extreme values (upper and lower bounds) for OCP [20] which suggests an aggressive control behaviour to arrive at the goal position as soon as possible. Meanwhile for OCP [22], the input trajectory almost always stays around the origin when applying inputs. The weights for the final optimal time and the control energy can be adjusted to prioritize the minimization of one over the other.

C. Closed-loop NMPC

As demonstrated in the previous section, the open-loop optimal control of the two-link planar arm was able to successfully steer the system towards the goal position in a nominal NMPC setting. Nominal NMPC schemes assume the plant states to be exactly known at sampling times and also assumes no plant-model mismatch. In addition, it is further assumed that for all $k \in N$, an optimal solution to the OCP exists and is attained. This arrangement requires strong assumptions which may not be possible to satisfy in real-life situations. Thus, in order to obtain practically applicable results, realistic scenarios must be implemented to verify the viability of the implementation. In reality, system models are never perfect due to inexact parameters which happen because of inaccuracies in parameter estimation, approximation and numerical errors. Similarly, state information isn't always available due to environmental conditions and delays. Mainly, states need to be measured or estimated during which additive disturbances act on the process. If the model isn't exact or measurement/estimation isn't performed accurately; violation of constraints, a decrease in performance or instability could occur. Hence, the analysis of such scenarios will be beneficial for the design of a robust MPC controller. Usually, two types of uncertainty must be considered:

- Additive Disturbance: $x_{k+1} = f(x_k, u_k) + w_k, \quad w_k \in \mathbb{W}$
- Parametric Uncertainty: $x_{k+1} = f(x_k, u_k, d_k), \quad d_k \in \mathbb{D}$

This work considers white Gaussian noise for the additive disturbance and an incorrect model with 2 inexact parameters for the parametric uncertainty. A closed-loop simulation of the system under the influence of different disturbances were performed with the parameter values depicted in Table III.

TABLE III
CLOSED-LOOP SIMULATION PARAMETERS OF THE FIXED-END-TIME NMPC SCHEME

Parameter	Values	Description
N	60	prediction horizon
σ	1e-3	variance of the measurement noise
wQ	10	weight of Q matrix
wR	0.1	weight of R matrix
wP	10^8	weight of P matrix
h	0.05	integration step size [s]
Δt	0.10	sampling time [s]

The OCP [16] was employed for the closed-loop simulation of the NMPC scheme with RK4 integrator. NMPC loop was operated until convergence or until a number of maximum iterations were reached. The number of maximum iterations allowed was 150 and the convergence tolerance was decided to be $1e-3$. The resulting trajectories from various scenarios for the closed-loop control of the robot are illustrated in the following graphs.

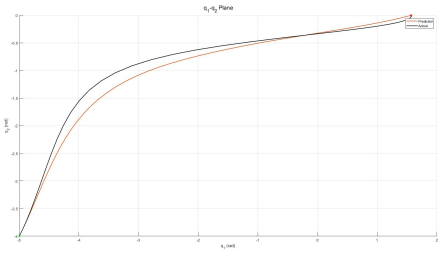

Fig. 10. Closed-Loop Simulation of OCP [16] in q_1-q_2-plane

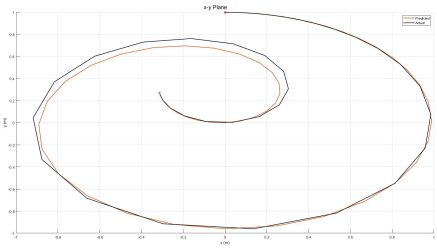

Fig. 11. Closed-Loop Simulation of OCP [16] in x-y-plane

Fig. 10 and Fig. 11 depict a nominal closed-loop NMPC simulation. Convergence occurs at 8 seconds and the closed-loop trajectory is almost identical to the open-loop trajectory since the model is exact and there exists no measurement noise.

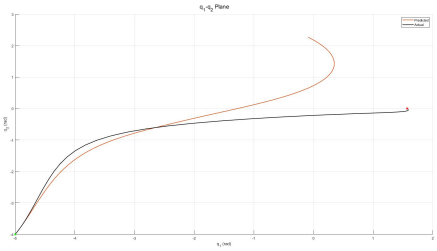

Fig. 12. Closed-Loop Simulation with Parametric Uncertainty of OCP [16] in q_1-q_2-plane

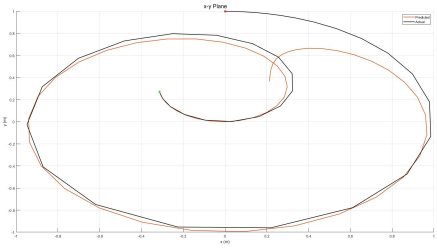

Fig. 13. Closed-Loop Simulation with Parametric Uncertainty of OCP [16] in x-y-plane

Fig. 12 and Fig. 13 show a closed-loop NMPC simulation with an inaccurate model. The inexact model has incorrect values for $b_1 = 180$ and $b_2 = 45$ instead of 200 and 45 respectively. The OCP [16] was formulated utilizing the inexact model but the inputs were applied to the exact model for both the open-loop and closed-loop simulations. Consequently, the predicted trajectory does not coincide with the actual one and diverges. Convergence occurs at 8.1 seconds for the closed-loop trajectory.

Fig. 14 and Fig. 15 illustrate a closed-loop NMPC simulation with a measurement noise. A Gaussian white noise with a reasonable variance value of $1e-3$ was mathematically devised and integrated. Even though the predicted and the actual trajectory seem to be approximately the same, convergence doesn't occur and the maximum number of NMPC

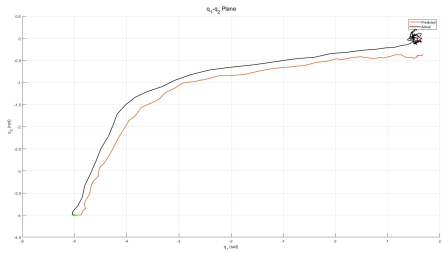

Fig. 14. Closed-Loop Simulation with Measurement Noise of OCP [16] in q_1-q_2-plane

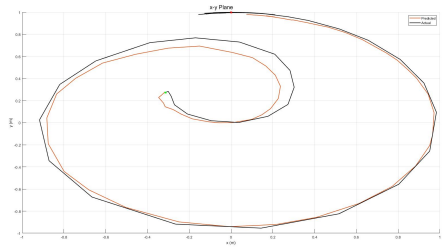

Fig. 15. Closed-Loop Simulation with Measurement Noise of OCP [16] in x-y-plane

iterations is reached due to oscillatory behaviour caused by this additive disturbance. To cope with this issue, one approach could be to reduce the sampling time to deal with overshoots or to tune the matrices according to the specific situation. Nevertheless, the goal position is reached fairly quickly and the magnitude of the final error is around $2e-3$.

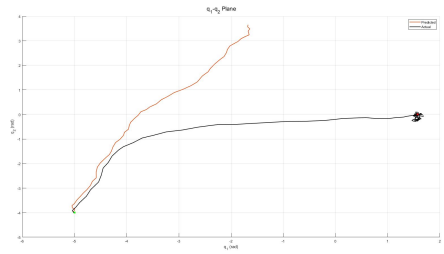

Fig. 16. Closed-Loop Simulation with Parametric Uncertainty and Measurement Noise of OCP [16] in q_1-q_2-plane

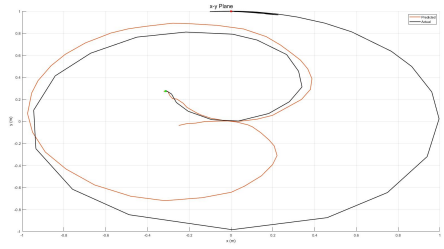

Fig. 17. Closed-Loop Simulation with Parametric Uncertainty and Measurement Noise of OCP [16] in x-y-plane

Fig. 16 and Fig. 17 demonstrate a closed-loop NMPC simulation with both parametric uncertainty and measurement noise. Again, convergence doesn't occur and the maximum number of NMPC iterations is reached due to oscillatory behaviour caused by measurement noise. Open-loop trajectory still diverges due to parametric uncertainty yet the desired state is reached for the closed-loop one.

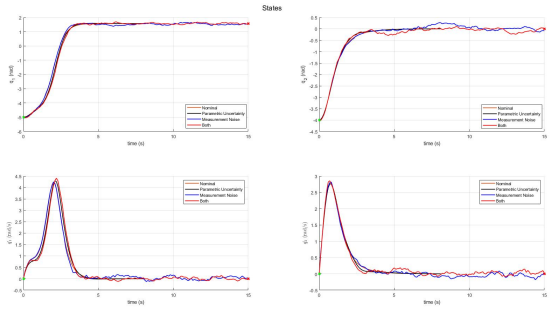

Fig. 18. States for Closed-Loop Simulation of OCP [16]

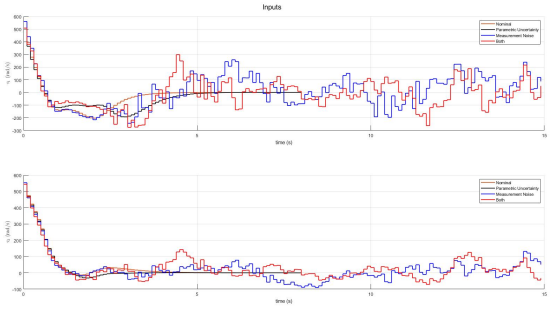

Fig. 19. Inputs for Closed-Loop Simulation of OCP [16]

Lastly, Fig. 18 and Fig. 19 are provided for a better visualization of the comparison of states and inputs for all the implemented scenarios for the closed-loop NMPC simulation.

IV. Stability Analysis of the NMPC Scheme

In this section, one analyzes the stability property of the presented NMPC schemes. Aforementioned, the system model, state and input constraints are provided for the 2-DoF robot and the control task is to stabilize upon a setpoint first under nominal NMPC setting (exact model, state feedback and availability of solutions at every timestep) then with disturbances for the closed-loop optimal control. The variants of NMPC are characterized by the choice of design aspects which are the stage cost, terminal penalty, terminal constraint, terminal region, prediction horizon and sampling time.

Throughout the report, two main approaches were discussed to stabilize the NMPC scheme. Firstly, a zero-terminal constraint with no terminal penalty $(=0)$ was used in combination with a sufficiently long prediction horizon N_{min} which ensures every open-loop trajectory reaches the terminal region $X_f = x_{goal}$ that only spans the desired state. Next in order, a scaled terminal penalty with no terminal constraint and no terminal region were employed where by weighting the terminal cost, the domain of attraction of the MPC controller without terminal constraint is enlarged reaching (practically) the same domain of attraction of the MPC with terminal constraint to stabilize the closed-loop system. With this, the terminal cost becomes an upper bound on the cost to go when it is chosen appropriately to be a local Lyapunov function that satisfies the stability criteria.

These variants are proven to achieve asymptotic stability in [2], [4] and [5] under the presented standing assumptions for nominal conditions.

Assumption 1 (Steady-State) Given the steady-state $\bar{x}$, there exists $\bar{u} \in \mathbb{U}$, such that $f(\bar{x},\bar{u}) = 0$ and $(\bar{x},\bar{u}) \in int(X \times U)$. Without loss of generality, one supposes $(\bar{x},\bar{u}) = (0,0)$.

Assumption 2 (Continuity) The functions $f : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$, $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_0^+$ and $V_f : \mathbb{X}_f \to \mathbb{R}_0^+$ are continuous and $f(0,0) = 0$, $\ell(0,0) = 0$, $V_f(0) = 0$. For all $x_0 \in \mathbb{X}$, and any $u(\cdot) \in \hat{\mathbb{C}}([0,T],\mathbb{U})$, the solution $x(\cdot,x_0,u(\cdot))$ exists on $[0,T]$ and is absolutely continuous.

Assumption 3 (Constraint Sets) The constraint set $\mathbb{X}$ is closed, $\mathbb{U}$ and $\mathbb{X}_f$ is compact and all sets contain the origin in its interior.

Assumption 4 (Bounds on the Stage and Terminal Cost) There exists class-K functions $\alpha_1, \alpha_2, \alpha_3 : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, such that

$$\alpha_1(\|x-\bar{x}\|) \leq \ell(x,u) \leq \alpha_2(\|x-\bar{x}\|)$$
$$V_f(x) \leq \alpha_3(\|x-\bar{x}\|)$$

Assumption 5 (Descent Property of the Terminal Cost) For all $x \in \mathbb{X}_f \subseteq \mathbb{X}$, there exists a local feedback control law $u = k_f(x)$ for which the following inequality holds

$$V_f(f(x,k_f(x))) - V_f(x) \leq -\ell(x,k_f(x)), \quad \forall x \in \mathbb{X}_f$$

Assumption 6 (Terminal Set is Positive Control Invariant) For all $x \in \mathbb{X}_f$, there exists $u = k_f(x) \in \mathbb{U}$ such that $f(x,k_f(x)) \in \mathbb{X}_f$.

For future work, another OCP formulation where both a terminal cost and a terminal inequality constraint are employed, can be considered for asymptotic stability. For instance, another quasi-infinite horizon NMPC scheme with guaranteed stability is presented [5] where the stage cost is quadratic, the terminal state penalty matrix of the terminal cost term is a solution of an appropriate Lyapunov equation and the terminal inequality constraint forces the states at the end of the finite horizon to lie within a prescribed terminal region. It is concluded that if the Jacobian linearization of the non-linear system to be controlled is stabilizable, feasibility of the OCP at $t = 0$ implies asymptotic stability of the closed-loop system.

V. Summary

This project formulates multiple NMPC schemes specified for the control task of setpoint stabilization with OCP formulations on fixed-end-time, free-end-time and minimal control energy problems for the rest-to-rest control of a 2-DoF robot. At base, quasi-infinite horizon NMPC scheme was adopted and a number of distinct ways to approach the NMPC problem are explored by identifying the relevant design aspects while also justifying the preference for the situation. The dynamic model of the two-link planar arm is derived using Lagrangian formulation where the generalized coordinates are the joint angles. The viability of the design of the NMPC controller was validated via closed-loop and open-loop simulations. Numerous simulation scenarios with types of uncertainties were performed numerically with different integrators at varying sampling rates to observe how certain design parameters influence the behaviour of the controller. Convergence and robustness were discussed and stability issues were addressed with standing assumptions.

VI. Acknowledgement

This case study was done as a part of the NMPC course at Technische Universität Dortmund. The guidance and the support of the instructor and the teaching assistants are acknowledged.

References

[1] Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. (2010). "Robotics: Modelling, Planning and Control". Springer Science Business Media.

[2] Mayne, David Q., et al. "Constrained model predictive control: Stability and optimality." Automatica 36.6 (2000): 789-814.

[3] Limón, Daniel, et al. "On the stability of constrained MPC without terminal constraint." IEEE transactions on automatic control 51.5 (2006): 832-836.

[4] Fontes, Fernando ACC. "A general framework to design stabilizing nonlinear model predictive controllers." Systems and Control Letters 42.2 (2001): 127-143.

[5] Chen, Hong, and Frank Allgöwer. "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability." Automatica 34.10 (1998): 1205-1217.