Modelling, Kinematics, Dynamics and Control Design for Under-actuated Manipulators

vorgelegt von
M.Sc.
Ahmad Albalasie
geb. in Hebron

von der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
- Dr.-Ing. –
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. phil. Manfred Thüring
Berichter:	Prof. Dr.-Ing. Günther Seliger
Berichter:	Prof. Dr.-Ing. Ahmed Abu Hanieh
Berichter:	Prof. Dr.-Ing. Jörg Krüger

Tag der wissenschaftlichen Aussprache: 16. August 2016

Berlin 2016

Dedications

To

my parents,

my wife,

and

my daughter

Acknowledgments

This thesis is submitted in partial fulfilment of the requirements for the doctoral degree at the Technische Universität Berlin, faculty of mechanical engineering and transport systems, department of assembly technology and factory management.

I would like to thank Prof. Seliger for his powerful supervision, suggestions, and continuous encouragements in every stage of my work. He gave me full access to the facilities and laboratories owned by the department, which led to the successful experimental realization of the work. I am also very grateful to Dr. Ahmed Abu Hanieh and to Prof. Dr. Jörg Krüger for their efforts in co-referring and reviewing my work. In addition, I would like to thank Dr. Brett for his works and his contributions on the first phase of the project. I would like to give special thank for M.Sc. Arne Glodde for his valuable and excellent technical assistance during my research period. Furthermore, I would like to extend my thanks to M.Sc. Mohsen Afrough for his contribution on designing the robot. Special thanks to the Avempace II Erasmus Mundus Action 2 for their financial support during my research period in Germany.

Abstract

Robotic handling operations cover a diversity of applications. Pick and place, palletizing or depalletizing, loading on machines or unloading from machines, storage/retrieval, and feeding the production lines are some examples. A manifold of different applications inspires the development of industrial robot types. The advantages of industrial robots can be summarized in different aspects: worker's protection in dangerous working conditions, higher working quality, higher productivity rate, and cost saving. Due to concerns about resource efficiency, energy consumption has become an issue for robotic development.

The System Applying Momentum Transfer for Acceleration of an End Effector with the Redundant Axis (SAMARA) is a robotic prototype of an industrial robot for pick and place applications. This prototype uses redundant, under-actuated configurations and an evolutionary algorithm (EA) to minimize energy consumption. Enabling for applications with relatively large displacement tasks, higher than one meter and high payload of up to 5.5 kilograms, the effectiveness of handling can be increased. Energy saving in specified cycle time has been achieved for this robotic kinematics. Reducing the cycle time and energy consumption are conflicting goals. However, actually, the computation time for the trajectory planning is too long. PID (proportional–integral–derivative) control is not adequate for a robust under-actuated motion (UAM). The uncertainty of payload causes unacceptable effects on accuracy, repeatability, and precision of the under-actuated robot.

Using the Quasi-Linearization (QL) is an approach for trajectory planning with minimizing energy consumption and reducing computation time. The QL is focused on reducing the cycle time to increase the productivity of the handling operations to achieve an optimal performance for the robot to meet the industrial requirements. The suggested control scheme uses the adaptive model predictive control (AMPC). The AMPC is classified as an advance optimal control technique; it has the ability to minimize the input torque, and the error between the actually achieved response and the desired response of the manipulator. The model has the inherent ability to deal naturally with constraints on the inputs and has the capability of updating the linearized dynamic model at each current operating point, which solves the problem of nonlinearity in dynamic equations of the robot.

Evaluations for the control scheme and for the trajectory planning are tested for SAMARA prototype. The concepts have been verified using several criteria, e.g., by comparing the results between the simulation power consumption and the actual power consumption measured from the physical prototype, comparing the performance of the QL approach with EA as trajectory planning algorithm for the under-actuated motion, and comparing the performance of SAMARA with other industrial robots from several perspectives.

The applicability of under-actuated robotic kinematics for practical applications has been approved by examples from food industry, and press lines industry, with their respective requirements.

Kurzfassung

Roboterbasierte Handhabungsoperationen sind in unterschiedlichen Anwendungsbereichen relevant. Beispiele sind Pick-und-Place Operationen, Palettierungen, Be- und Entladen von Maschinen sowie die Bereitstellung von Fertigungslinien mit Werkstücken. Die Entwicklung von Industrierobotern wird von einer Vielfalt unterschiedlicher Anwendungsbereiche inspiriert. Vorteile von Industrierobotern sind hohe Arbeitsgenauigkeit, hohe Produktivität, Kostenreduktion und der mögliche Einsatz in gefährlichen Produktionsumgebungen. Vorbehalte gegenüber Industrierobotern ergeben sich aus ihrer oft geringen Ressourceneffizienz, weswegen der Energieverbrauch in den Fokus aktueller Entwicklungsarbeiten tritt.

Das „System Applying Momentum Transfer for Acceleration of an End Effector with the Redundant Axis“ (SAMARA) ist der Prototyp eines Industrieroboters für Pick-und-Place Operationen. Unter Verwendung redundanter, unteraktuierter Konfigurationen und Evolutionärer Algorithmen (EA) kann der Energieverbrauch reduziert werden. Für Anwendungen mit verhältnismäßig größeren Handhabungsweglängen und hohen Traglasten von bis zu 5,5kg kann die Effektivität der Handhabungsoperation verbessert werden. Für diese Roboterkinematik konnten bereits Energieeinsparungen erreicht werden. Es hat sich gezeigt, dass die Verringerung der Zykluszeit bei gleichzeitiger Reduktion des Energieverbrauches ein Zielkonflikt darstellt. Gleichermaßen ist die Berechnungszeit zur Planung der Trajektorie der Endeffektoren zu hoch. PID Control eignet sich nicht für robuste unteraktuierte Bewegungen (UAM). Unsicherheiten über die Höhe der Traglast beeinflussen die Ablagegenauigkeit sowie die Wiederholgenauigkeit bei der Handhabung.

Der Ansatz einer Quasi-Linearization (QL) dient der Planung von Trajektorien bei gleichzeitiger Reduktion von Energieverbrauch und Rechenzeit. QL ist auf die Reduktion der Zykluszeit und damit auf die Erhöhung der Produktivität von Handhabungsoperationen durch die Entwicklung eines verbesserten Controllers gerichtet, um die aktuellen industriellen Anforderungen zu erfüllen. Die Control-Strategie verwendet ein sog. Adaptive Model Predictive Control (AMPC). Das AMPC ist eine verbesserte Control-Strategie mit der Eigenschaft, das Eingansdrehmoment und den Fehler zwischen Soll- und Istposition des Manipulators zu minimieren. Das Modell hat darüber hinaus die Fähigkeiten mit Randbedingungen der Eingänge umzugehen und das

linearisierte dynamische Modell zu aktualisieren, wodurch das Problem der Nichtlinearität der dynamischen Gleichungen des Roboters umgangen werden kann.

Eine Bewertung des Control-Vorganges und der Trajektorieplanung wurde für SAMARA durchgeführt. Die Konzepte wurden unter Berücksichtigung mehrerer Kriterien verifiziert. Hierzu zählen der Vergleich der Ergebnisse von simuliertem Energieverbrauch mit dem tatsächlichen im Betrieb gemessenen, der Leistung des QL-Ansatzes mit EA als Algoritmus zur Planung von Trajektorien und der Leistung von SAMARA mit anderen Industrierobotern.

Sowohl in der Lebensmittel- als auch in der Druckindustrie konnten unteraktuierte Roboter ihre Anwendbarkeit unter Beweis stellen.

Table of contents

List of Figures

List of Tables

Index of Abbreviations

A	Ampere.
AC	Alternative current.
AM	Actuated motion.
AMPC	Adaptive model predictive control.
CF	Cost function for the optimization problem.
cm	Centimeter.
DC	Direct current.
DOF	Degree of freedom of the mechanical system.
e.g.	For example.
E1	First experiment.
E2	Second experiment.
E3	Third experiment.
E4	Fourth experiment.
E5	Fifth experiment.
E6	Sixth experiment.
EA	Evolutionary algorithm.
EM	End motion phase.
FE	The first experiment to measure the power consumption for the 1st prototype of SAMARA when the used controller from Bosch company.
G	Gear ratio.
GA	Genetic algorithm.
GE1	The first group of experiments.
GE2	The second group of experiments.
GE3	The third group of experiments.
GE4	The fourth group of experiments.
i.e.	In other words.
kg	Kilogram.
kg/m^2	Kilogram per square meter.
kg/m^3	Kilogram per cubic meter.

kN	Kilonewton.
kW	Kilowatt.
LTI	Linear time invariant.
m	Meter.
M1	The first trajectory planning algorithm uses the QL algorithm for the UAM and the analytical method for the NSM.
M2	The second trajectory planning algorithm uses the PTP synchronized trajectory planning algorithm based on the fifth order polynomial method for the AM and the analytical method for the NSM.
m^2	Square meter.
MAS	Main axes system.
Min.	Minimize.
mm	Millimeter.
MPC	Model predictive control.
ms	Millisecond.
N	Newton.
NM	Newton meter.
NMPC	Nonlinear model predictive control.
NSM	Null space motion.
PID	Proportional–integral–derivative.
ppm	Picks per minute.
PTP	Point to point.
QL	Quasi-Linearization.
QP	Quadratic programming.
rad	Radian.
rad/s	Radian per second.
rad/s^2	Radian per second squared.
s	Second.
SAMARA	System Applying Momentum Transfer for Acceleration of an End-effector with Redundant Axis.
SAS	Secondary axes system.

SC1	Adept one MV SCARA.
SC2	Turboscara SR 60 SCARA.
SCARA	Selective Compliance Assembly Robot Arm.
SE	The second experiment to measure the power consumption for the 1st prototype of SAMARA when the used controller from Beckhoff company.
ST	Start motion phase.
T€	Thousand €.
TPBVP	Two-point boundary value problem.
UAM	Under-actuated motion.
V	Volt.
W	Watt.

Index of Formula Symbols

a	Number of active axes in the MAS.
$\mathrm{A_k} \in \mathrm{R}^{\mathrm{r*r}}$	The dynamic matrix for the LTI model.
$\mathrm{A} \in \mathrm{R}^{\mathrm{2r*2r}}$	The calculated A matrix into the QL algorithm.
β	The angle of the orientation of the end effector in the base frame.
$\mathrm{B_k} \in \mathrm{R}^{\mathrm{r*z}}$	The input matrix for the LTI model.
C	Control horizon.
$\mathrm{c_a(q(t), \dot{q}(t))} \in \mathrm{R^a}$	The Coriolis and centrifugal forces for the active axes in the MAS.
$\mathrm{C_k} \in R^{o*r}$	The output matrix for the LTI model.
$\mathrm{c_p(q(t), \dot{q}(t))} \in \mathrm{R^p}$	The Coriolis and centrifugal forces for the passive axis in the MAS.
$C\big(q(t), \dot{q}(t)\big) \in R^n$	The Coriolis and centrifugal forces for the MAS.
$\mathrm{c} \in \mathrm{R^r}$	Unknown vector.
$\mathrm{d_a(q(t), \dot{q}(t))} \in \mathrm{R^a}$	The damping and friction moments for the active axes in the MAS.
$\vartriangle u(i)_{max} \in \mathrm{R^z}$	The maximum allowable change of rates in the inputs at the prediction i.
$\vartriangle u(i)_{min} \in \mathrm{R^z}$	The minimum allowable change of rates in the inputs at the prediction i.
d_i	The offset distance along previous z_{i-1} to the common normal.
$\mathrm{D_k} \in R^{o*z}$	The feedback matrix for the LTI model.
$\mathrm{d_p(q(t), \dot{q}(t))} \in \mathrm{R^p}$	The damping and friction moments for the passive axis in the MAS.
$D\big(\dot{q}(t)\big) \in R^n$	The damping and the friction moments for the MAS.
$e_1(t) \in \mathrm{R^r}$	The difference between the particular solution and the homogenous solution for the states.
$e_2(t) \in \mathrm{R^r}$	The difference between the particular solution and the homogenous solution for the co-states.
H	Hamiltonian equation.
$h_p(q, \dot{q})$	Represents the Coriolis, the centrifugal, the gravity, and the damping terms for the passive axis/axes equation/s.
I_{xx}	Represents the second moment of area with respect to the x-axis.

I_{yy}	Represents the second moment of area with respect to the y-axis.
I_{zz}	Represents the second moment of area with respect to the z-axis.
J	Jacobian matrix.
$J^{\#}$	pseudo-inverse of the Jacobian matrix.
$J_{payload}$	The inertia of the payload.
k	Current control interval.
$\lambda^{Hi}(t_0) \in \mathrm{R}^{\mathrm{r}}$	The initial condition of the co-states for the i-th homogenous solution.
$\lambda^{p}(t_0) \in \mathrm{R}^{\mathrm{r}}$	The initial condition of the co-states for the particular solution.
$\lambda \in \mathrm{R}^{\mathrm{r}}$	The Lagrange multipliers functions, also called the co-state variables.
$\dot{\lambda}^{*} \in \mathrm{R}^{\mathrm{r}}$	Represents the optimal derivatives for the co-states.
l_{i}	Length of the link i.
$\mathrm{M(q)} \in \mathrm{R}^{\mathrm{n*n}}$	The inertia matrix for the MAS.
n	Number of axes in the MAS.
N_b	Number of teeth of the output gear (payload side).
N_m	Number of teeth of the input gear (motor side).
o	Number of the outputs for the LTI model.
p	Number of passive axes in the MAS.
P	Predication horizon.
$\dot{p}$	Represents the generalized velocities in the task space.
$\emptyset$	The standard cost function of the QP.
$\emptyset_1$	The cost function for minimizing the position error for the QP.
$\emptyset_2$	The cost function for minimizing the inputs or to become near the nominal target inputs for the QP.
$\emptyset_3$	The cost function for the change rates of the inputs for QP.
p_i	Represents the generalized coordinates in the task space in the direction of i-axis.
$\mathrm{q_a(t)} \in \mathrm{R}^{\mathrm{a}}$	The generalized coordinates for the active axes in the MAS.
$\dot{q}$	Represents the generalized velocities in the joint space.
q_i	Represents the generalized coordinates for i-axis in the joint space.

$\mathrm{q_p(t)} \in \mathrm{R^p}$	The generalized coordinate for the passive axis in the MAS.
$\mathrm{Q} \in \mathrm{R^{r*r}}$	Weighting matrix, where it is positive semi-definite matrix.
q_{SAS}	The angular positions for the SAS into the payload side.
$\dot{q}_{SAS}$	The angular velocities for the SAS into the payload side.
$\ddot{q}_{SAS}$	The angular accelerations for the SAS into the payload side.
$\mathrm{q(t)} \in \mathrm{R^n}$	The generalized coordinates for the MAS.
r	Number of states for the LTI model, also it is the number of the states for the MAS.
$ref \in R^r$	Reference trajectory.
$\mathrm{r(k+i\|k)} \ \in \mathrm{R^o}$	Represents the reference vector at the i-th prediction horizon in the current sample k.
ρ	Damping factor.
$\mathrm{R, S} \in \mathrm{R^{z*z}}$	Weighting matrix, where they are positive definite matrices.
$\mathrm{S_u} \in \mathrm{R^z}$	Scaling factors for the inputs of the QP.
$\mathrm{S_y} \in \mathrm{R^o}$	Scaling factors for the outputs of the QP.
t_f	Final time.
$\mathrm{t_{f\_UAM1}}$	The final time for the $\mathrm{UAM_1}$ when the robot does not carry a payload.
$\mathrm{t_{f\_UAM2}}$	The final time for the $\mathrm{UAM_1}$ when the robot carries a payload.
$\mathrm{t_{NSM}}$	The time for each NSM.
t_0	Initial time.
T_p	Peak torque.
$\mathrm{T} \in \mathrm{R^a}$	Vector containing active motor torques in the MAS.
T_{rt}	Rated torque.
T_s	Sampling time.
T_{ss}	Standstill torque.
$\mathrm{\overline{u}_k} \in \mathrm{R^z}$	It is associated to control input for MAS only.
$\mathrm{u_i}$	The motor i for axis i in the MAS.
$u(i)_{max} \ \in \mathrm{R^z}$	The maximum allowable torque inputs at the prediction i.

$u(i)_{min} \in \mathrm{R}^{\mathrm{z}}$	The minimum allowable torque inputs at the prediction i.
$U_{max} \in \mathrm{R}^{\mathrm{n}}$	The maximum allowable torque inputs.
$U_{min} \in \mathrm{R}^{\mathrm{n}}$	The minimum allowable torque inputs.
$\mathrm{U} \in \mathrm{R}^{\mathrm{n}}$	The torque vector in the dynamic equation of the MAS.
$u^{*} \in \mathrm{R}^{\mathrm{a}}$	Represents the optimal torque inputs for the MAS in the UAM phase.
$u_{target} \in \mathrm{R}^{\mathrm{z}}$	Nominal target inputs.
v	Number of all states for the robot (including the states of the MAS and the SAS).
φ	Preselected as a small number close to zero.
V_{rs}	Rated speed.
$\bar{\mathrm{x}}_{\mathrm{k}} \in \mathrm{R}^{\mathrm{r}}$	Represents the error between the actual states and the reference for MAS only.
$\dot{x}_{model\,1}$	The first state space model for the robot in the phase of the UAM when the robot is not gripping a payload.
$\dot{x}_{model\,2}$	The second state space model for the robot in the phase of the UAM when the robot is gripping a payload.
$\dot{x}^{*} \in \mathrm{R}^{\mathrm{r}}$	Represents the optimal derivatives for the states.
$x^{Hi}(t) \in \mathrm{R}^{\mathrm{r}}$	The i-th homogeneous solution.
$x^{Hi}(t_0) \in \mathrm{R}^{\mathrm{r}}$	The initial condition of the states for the i-homogenous solution.
$x^{p}(t)$	The particular solution.
$x^{p}(t_0) \in \mathrm{R}^{\mathrm{r}}$	The initial condition of the states for the particular solution.
$X \in R^{r}$	Denotes to the state vector.
$x(t_f), x_f \in \mathrm{R}^{\mathrm{r}}$	The final states for the TPBVP.
$x(t_0), x_0 \in \mathrm{R}^{\mathrm{r}}$	The initial states for the TPBVP.
$y(i)_{max} \in \mathrm{R}^{\mathrm{o}}$	The maximum allowable outputs at the prediction i.
$y(i)_{min} \in \mathrm{R}^{\mathrm{o}}$	The minimum allowable outputs at the prediction i.
$y(\mathrm{k}+\mathrm{i}\vert\mathrm{k}) \in \mathrm{R}^{\mathrm{o}}$	Represents the predicted vector for the response at the i-th prediction horizon in the current sample k.
z	Number of inputs for the LTI model.

1 Introduction

1.1 Motivation

The development of efficient handling robot is a challenging problem for researchers and engineers. This emerges from the fact that there are different requirements to improve the performance of the handling robot. Several of these requirements are in conflict with each other, such as the necessity to minimize the power consumption and minimize the cycle time for the robot [Kir-04, p.247 ff.]. Consequently, engineers focus on improving the most important features of the robot while neglecting others depending on the requirements of the particular industry. However, this approach has disadvantages, because the robot can evolve better if the engineers can take all requirements into consideration, or at least, they can reduce the effect of such conflicts if they add them as requirements in the design phase.

This research concentrates on two important goals for developing an efficient material handling robots. Firstly, to increase power saving to reduce costs. Secondly, to increase the productivity rate of the material handling lines. The last goal is achieved partly by integrating and using industrial robots in the handling operations instead of using labourers to perform the handling tasks. Subsequently, optimizing the performance of the under-actuated robot to maximize energy saving with the capability of decreasing the cycle time at the same time is the scope of this research.

According to [Kir-04, p.247 ff.], the minimum time response for the robot is equal to using the maximum level of the control input during the execution period; whereas, minimizing the energy consumption is equal to using the minimum level of control. Thus, in general, there is a conflict between these two goals from a control point of view. This dissertation introduces and verifies an approach to reduce the effect of the previous problem by solving the optimization problem to minimize the power consumption while reducing the cycle time capability for the under-actuated manipulator prototype.

1.2 Project background

In the first phase of this research, the 1st prototype of the SAMARA robot was built and designed at TU Berlin to test the conceptual design, evaluate the performance of the motion planning, and develop the control scheme for the under-actuated robot.

The first prototype has three degrees of freedom (DOFs) to execute the UAM and the null space motion (NSM) without the existence of the end effector; see Figure 1:1. Therefore, the robot at that time cannot perform a physical pick and place task. Furthermore, this design has redundant and under-actuated configurations in the phase of the UAM to apply the idea of the effective use of the robot dynamics in the trajectory planning. In addition, EA has been used in the motion planning process in the phase of UAM to minimize the energy consumption [Bre-10], [Bre-13, p. 105].

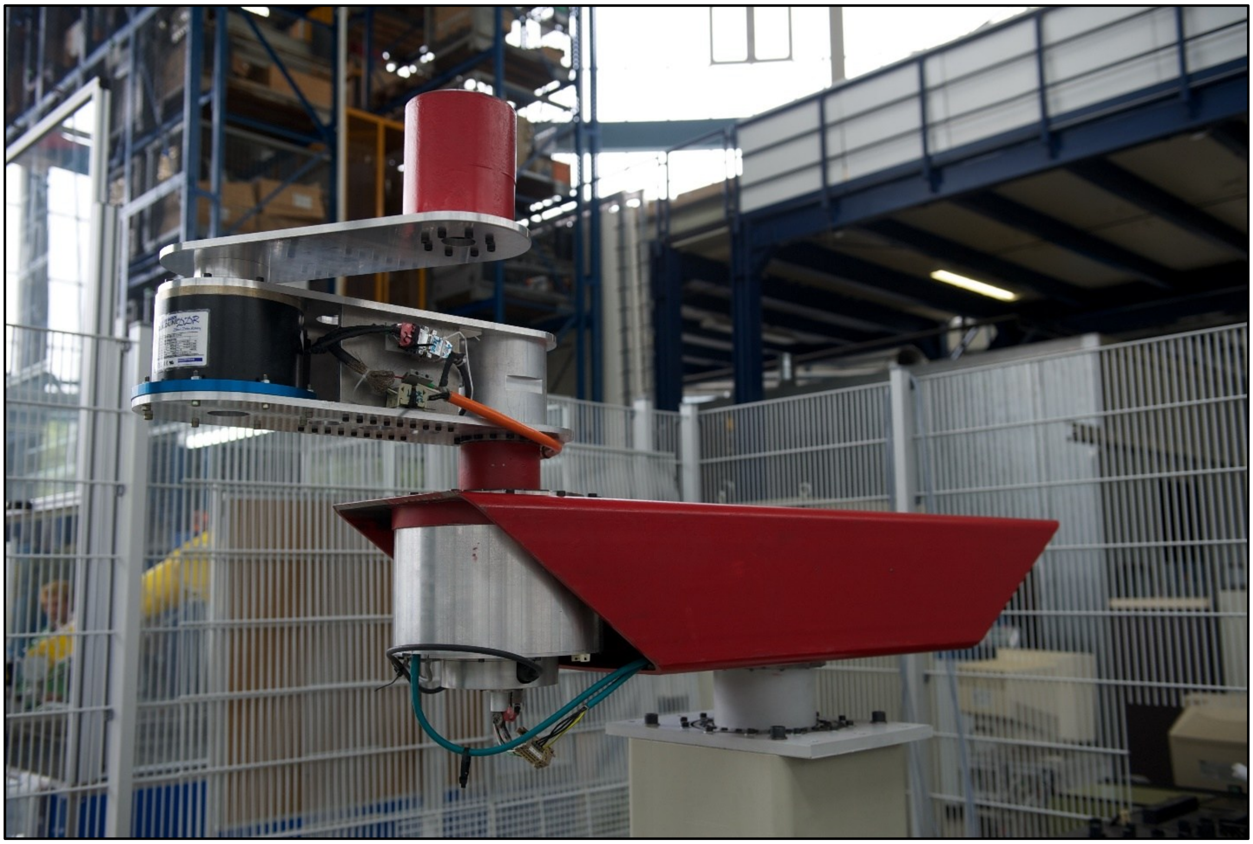


Figure 1:1: The 1st prototype of SAMARA robot.

The essential advantages of using these concepts can be summarized by the ability to integrate two types of motions to perform a pick and place task, the ability of reducing the energy consumption on the basis of simulation, and on experimental basis in several cases. The 1st prototype of SAMARA covers a wide range of motions; also it is designed to carry up to 15 kg including the mass of the end effector, but the end effector is not integrated into the structure of the robot [Bre-13, p. 117]. In addition, one of the main goals of the previous phase of research is to decrease the ratio of the robots accelerated masses by applying the idea of effective use of robot dynamics in trajectory planning [Bre-08]. While the project faced critical obstacles based on the author point of view which can be summarized in the following points: the computation time for solving the optimization problem by using the EA consumed around two hours in the case of low error [Bre-13, p. 108]. This result means the EA is not suitable to work in a real-time environment without storing the pre-calculated trajectories in a database, but in this case, any modification will lead to a simulation of the motion again using the same approach. Furthermore, there is a poor matching between the simulated and the actual power consumption [Bre-13, p. 134]. Therefore, at best, the actual power consumption is higher than the simulation results by three to five times [Bre-13, p. 134]. Moreover, there is no clear methodology to describe how to reduce the cycle time to increase the handling rate of the robot. Another challenging problem was that the achieved positioning accuracy was in the range of several centimetres and this is not appropriate for any typical applications [Bre-10]. In addition, there is a possibility for the divergence of the optimization solution, i.e., there is no solution found if the initial guess was poor.

1.3 Objectives

This work focuses on developing a trajectory planning algorithm and on improving the control scheme for the 2nd prototype of SAMARA, which will aim to minimize the power consumption with a capability of reducing the cycle time, as much as possible, to increase the handling rate. In addition, this prototype must be able to execute physical pick and place tasks with better position accuracy. In addition, this prototype shall be able to carry up to 5.5 kg for long distances with a short cycle time.

Another challenging problem that has been taken as a goal in this research is to decrease the computation time, to become more reliable, if any modifications are required for any reason. This can be achieved by replacing the code, which solves the optimization algorithm, by another code that uses another method in calculations to improve the computation time. Finally, in this work, the gap between the simulation and the actual results for the power consumption will be reduced to increase the accuracy of the simulation model in comparison with the physical behaviour for the robot.

1.4 Outline of the thesis

Chapter 1 contains four sections, briefly. The first section discusses the motivation of this research field showing why it is important to develop such kinematics and such robots from several perspectives. The second section discusses the background for this research. The objectives of this dissertation have been clarified in section 3. Finally, the outline of this dissertation is given in section 4.

Chapter 2 has six sections. Firstly, an overview of the industrial robots is discussed in section 1. Secondly, the kinematics of the under-actuated robot has been clarified in section 2. Next, the non-holonomic constraints and the motion phases with their dynamics equations of the under-actuated robot are shown in section 3 and 4 respectively. After that the trajectory planning for the robot is shown in section 5. Finally, an introduction about model predictive control (MPC) is shown in section 6.

Chapter 3 discusses the requirements of the efficient control and trajectory planning for the under-actuated robot.

Chapter 4 includes several essential topics in this dissertation, such as minimizing the energy consumption, approach of decreasing the cycle time, the QL algorithm as an approach for trajectory planning, position control scheme, and optimal control scheme.

Chapter 5 evaluates the results for the trajectory planning by using QL, in addition to the results for using optimal control scheme. After that the reduction in the power consumption and the performance of the robot are discussed in this chapter as well. Finally, the comparison between the 2nd prototype of SAMARA and other modern industrial robots is presented in this chapter to evaluate the benefits.

Chapter 6 presents the usage of the under-actuated robot in diverse industrial applications.

Finally, chapter 7 discusses the summary, the outlook, and the recommendations based on the achieved results.

2 Motion Phases and Trajectory Planning for Under-actuated Mechanisms

2.1 Industrial robots

The international organization for standardization defines the industrial robot as *"an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications"* [ISO-12]. The first application field for using the industrial robots is the automotive industry in 1961 by General Motors. At that time, General Motors used the robot "UNIMATE" for die-casting at an automobile factory in USA [Nof-09, p. 17]. Later on, the industrial robots invaded several fields of applications because of their high capability in performing tasks in limited time, and due to their accuracy, their precision, and their capability for repeatability.

Robotic operations cover diversity of operations. The major operations are handling operations, welding operations, assembly operations, processing operations, and dispensing operations. This wide usage of the industrial robots causes a dynamic process to develop the industrial robots continuously. A manifold of different applications inspired developing different types of industrial robots. This led to a dramatic development in the manufacturing and production processes during the previous years. Due to concerns about resource efficiency, energy consumption becomes a major challenge for industrial robots' development. For this reason, evolving and improving industrial robots that consume less energy is an essential task for engineers and researchers.

There are five essential types of industrial robots classified based on the mechanical structure design. These types are articulated robots, cylindrical robots, linear robots (including Cartesian and gantry robots), parallel robots, and Selective Compliance Assembly Robots Arm (SCARAs); see Figure 2:1. Each of these robots types has its specific features to perform the required tasks.

Usually, the decision makers select the suitable type of the industrial robot depending on several criteria, such as the DOFs in the mechanical structure, the covered operational space, the maximum payload, the accuracy, the repeatability, the precision, the cycle time, the mass of the robot, the inertia of each axes, the man-machine interfacing ability, the programming flexibility, the service quality, the reliability, etc. Consequently, it is difficult for the decision maker to select the suitable type of the robot and the suitable model as well, due to the diversity of products, and the diversity of specifications for each robot. Anyway, there is a scientific approach suggested by Chatterjee to select the suitable industrial robot depending on the requirements of each application field [Cha-10].

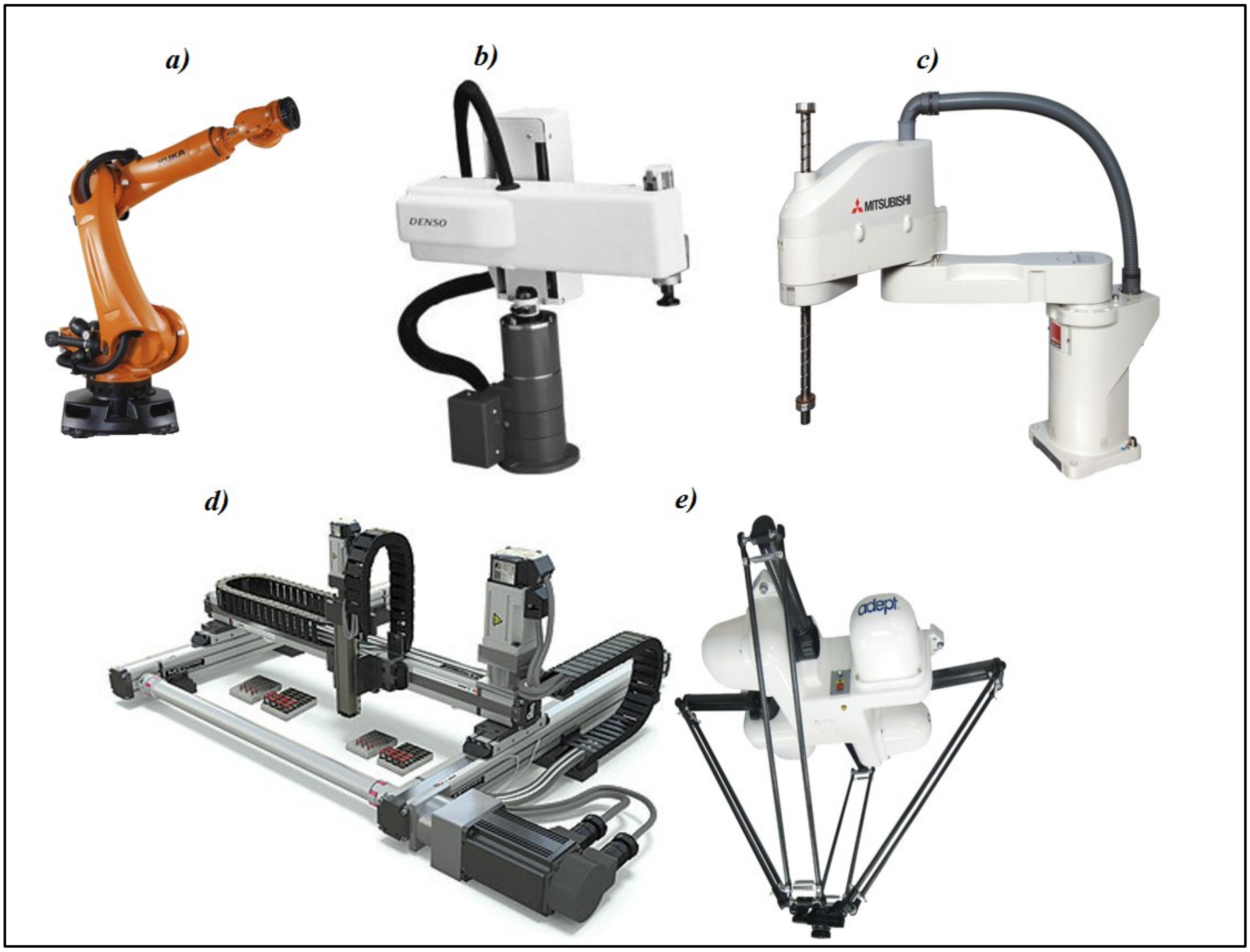


Figure 2:1: Types of the industrial robots from mechanical structure point of view: a) articulated [Kuk-16a] b) cylindrical [Den-16] c) SCARA [Mit-16] d) gantry [Pbc-16] e) parallel [ADE-16].

This study provides a wide discussion for the modelling, kinematics, dynamics, trajectory planning, and it suggests two schemes for controlling the 2nd prototype of the under-actuated robot called SAMARA for use in pick and place applications.

The 2nd prototype of SAMARA is designed to carry payloads up to 5.5 kg; it can reach up to 0.963 m in 2 s as the minimum achieved cycle time, and it can save power consumption up to 65% with comparison to other industrial robots, for more details see section 5.3. All these features will be discussed in detail in the following chapters and sections.

2.2 Kinematics

This section discusses various concepts and classifications used in developing the mechanical design of SAMARA robot besides clarifying to the direct and inverse kinematics used to study the motion of the robot.

Two classes of kinematics chains are used for the mechanical structure base on the topological viewpoint: open and close chains. As reported in [Sci-00, p. 39], the open chain for the manipulator formed by a series of links connects the base frame to the end effector. While the closed chain uses a loop to connect the base frame to the end effector. A famous example of the bad effect of using an open serial chain is found in the performance of SCARA robot, e.g., an increase in the distance between the wrist and the shoulder axis decreases the position accuracy of the SCARA robot [Sci-00, p. 9]. This is the reason SCARA robot is not adequate for long distances with high payloads without modifying the structure. On the contrary, the accuracy increases if the closed loop chain is used in the mechanical structure of the robot when the payloads are large. Occasionally, this is used in the structure of the anthropomorphic robot by using the parallelogram mechanism between the shoulder and elbow joints [Sci-00, p. 10]. In this regard, studying other concepts from the kinematics point of view is substantial to evaluate and predict the performance of the robot in a proper way.

Now, let us focus on the definitions of redundancy. There are different types of redundancy[1] such as sensor redundancy, kinematic redundancy, and actuation redundancy [Con-97]. According to [Sci-00, p. 63], *"a manipulator is termed kinematically redundant when it has a number of degrees of mobility greater than the number of variables that are necessary to describe a given task"*. Conkur defines the redundant mechanism as the mechanism that has more DOFs than the dimension of the task variables [Con-97]. There are different examples of the redundant applications but the most famous application is the human arm, because it has seven DOFs with neglecting the DOFs in the fingers [Sci-00, p. 64]. The advantage of using the redundancy as a configuration in the robot is to provide dexterity during the motion. In addition, this configuration is useful for a robot that needs to avoid an obstacle.

[1] In this dissertation, redundancy means the kinematic redundancy.

The 1st prototype of SAMARA robot was designed and built. The primary research results are discussed briefly in section 1.2. The 2nd prototype of SAMARA robot was designed and built to solve the problems faced in the first research period. This prototype has five DOFs as shown in Figure 2:2. It utilizes redundant and under-actuated configurations. The 2nd prototype of SAMARA consists of three main axes system and their subsystems, which are called (MAS), and two secondary axes system and their subsystems, which are called (SAS), as shown in Figure 2:2 and Figure 2:7. The MAS determine the position of the end effector, while SAS has been used to orientate the payload and to lift the payloads upward or downward. Based on the previous discussion, this configuration is an example of an open serial chain.

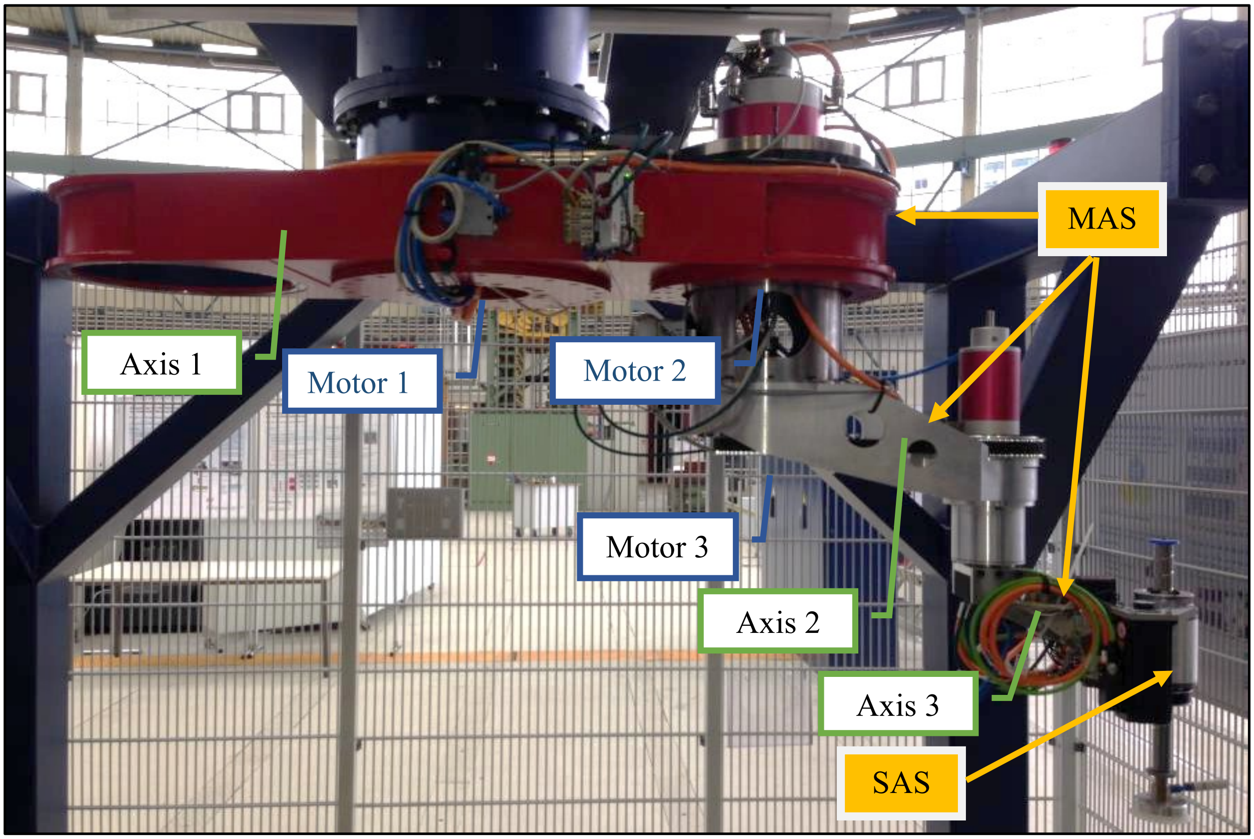


Figure 2:2: The 2nd prototype of SAMARA robot assembled in the research laboratory.

The shape of the reachable space for the 2nd prototype of SAMARA robot is a hollow cylinder. The maximum radius of the cylinder is 0.963 m while the minimum radius of the cylinder is 0.491 m. In addition, the maximum height of a cylinder is 16 cm. In Table 2:1, a comparison is made to clarify the major differences and the major similarities between the 1st and the 2nd prototypes of SAMARA from the mechanical point of view.

Table 2:1: The major differences and the major similarities between the 1st and the 2nd prototypes of SAMARA from the mechanical point of view.

-	Item	Reference	Unit	1st prototype of SAMARA	2nd prototype of SAMARA
Concepts	DOFs	[Bre-13, p. 97] [Alb-16]	[-]	Three	Five
	Existences of the end effector, i.e. the SAS.	[Bre-13, p. 117]		No	Yes
	The capability of the robot to perform a pick and place task.	[Bre-13, p. 117]		No	Yes
	Redundant configuration for the MAS.	[Bre-13], [Alb-15]		Yes	Yes
	Uses the under-actuated configuration into the path planning for the UAM.	[Bre-13], [Alb-15]		Yes	Yes
	The topological viewpoint for the MAS.	------		Open chain	Open chain
	The location of the third motor.	------		Second axis	Third axis
	Air suction cup system.	------		No	Yes
	The shape of the operational task	------		Hollow cylinder	Hollow cylinder
Features	Mass of the robot	[Bre-13, p. 117]	[kg]	197.91	228.37
	Maximum payload	-----		-----	5.5 kg
	Maximum reachable distance for the end effector	[Bre-13, p. 117]	[m]	1.297	0.963
	Minimum reachable distance for the end effector	[Bre-13, p. 117]		0.661	0.491
	Stroke	-----	[cm]	-----	16

According to [Bre-08], one of the problems for the industrial robot is related to the usage of the open serial chain; each actuator must accelerate the rest of the chain and this affects the performance of industrial robots. Therefore, a solution has been suggested to decrease the acceleration for each mass by using three principles as stated in [Bre-08]. The first principle[2] is that the robot maintains a uniform velocity for a special link and this is achieved by increasing the

[2] In the case of the 2nd prototype of SAMARA robot, the first link has been chosen to apply this principle.

inertia and the mass of the link. The second principle is the robot uses a redundant configuration to apply the self-motion or the NSM to provide opportunity for the end effector to execute the pick and place tasks from or on the processing points. The third principle is the robot using dynamic coupling between and through the axes and the links.

In the case of the 2nd prototype of SAMARA, the third axis is the passive axis in the phase of the UAM is shown in subsection 2.4.1. Therefore, the third axis is indirectly controlled by applying the third principle. Consequently, applying the third principle in controlling the third axis saves the power consumption during the UAM phases.

By using this design, the redundant robot axis constitutes an under-actuated configuration to realize the idea for the effective use of robot dynamics. The trajectory planning uses the centrifugal and the Coriolis forces to reduce the required torque for each axis. The coupling forces cause a highly nonlinear coupling because of the existence of a passive axis.

Now, let us focus on presenting the direct kinematics for the SAMARA prototype. The procedures to calculate the direct kinematics for any robot is summarized by calculating the Denavit-Hartenberg table, and then using the homogeneous transformation matrix principle to transform the coordinates from one frame to another, continuously. As a result, the position and the orientation of the end effector are determined in the base frame; see Figure 2:3.

The orientation and the position of the end effector relative to the base frame are determined in Eq. (2:1); more details are available in Appendix 9.1.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ \beta \end{bmatrix} = \begin{bmatrix} l_1 * \cos(q_1) + l_2 * \cos(q_{12}) + l_3 * \cos(q_{123}) \\ l_1 * \sin(q_1) + l_2 * \sin(q_{12}) + l_3 * \sin(q_{123}) \\ -d_2 - d_3 - d_4 \\ q_{1235} \end{bmatrix} \tag{2:1}$$

where:

$q_{1235} = q_1 + q_2 + q_3 + q_5$

$q_{123} = q_1 + q_2 + q_3$

$q_{12} = q_1 + q_2$

Where, p_i represents the generalized coordinates in the task space in the direction of i-axis, l_i is the length of the link i, q_i is the generalized coordinates for i-axis in the joint space, d_i is the offset distance along previous z_{i-1} to the common normal, and β is the angle of the orientation of the end effector in the base frame. Consequently, the inverse kinematics for SAMARA prototype is calculated using the standard approach. The Jacobian matrix uses the results to map the trajectory from the task space to the joint space or vice versa see Eq. (2:2).

$$\dot{q} = J^{-1}(q)\dot{p} \tag{2:2}$$

Where $\dot{p}$ represents the generalized velocities in the task space, J the Jacobian matrix, and $\dot{q}$ the generalized velocities in the joint space. Unfortunately, this method cannot work in the cases of singularity configurations, because the inverse of the Jacobian matrix becomes a singular matrix, i.e., the end-effector cannot reach a specific direction of motion [Bus-04]. As a result, two solutions have been suggested to solve this issue by approximating the Jacobian matrix.

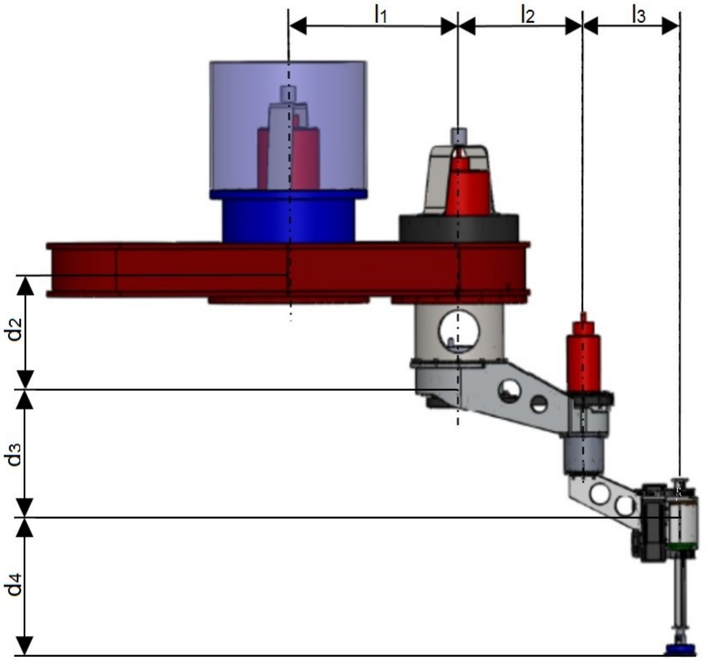


Figure 2:3: SolidWorks model for the 2nd prototype of SAMARA robot.

One solution suggests replacing the inverse of the Jacobian matrix by the pseudo-inverse of the Jacobian matrix. Nevertheless, there is a chance for discontinuity, especially in the cases of singular configurations see Eq. (2:3).

$$\dot{q} = J^{\#}\dot{p} \tag{2:3}$$

$$J^{\#} = J^T(JJ)^{-1}$$

The second solution suggests using the damped pseudo-inverse of the Jacobian matrix as shown in Eq. (2:4). This solution solved the problem of discontinuity.

$$\begin{gathered}\dot{q} = J^{\#}\dot{p} \\ J^{\#} = J^T(JJ + \rho^2 I)^{-1}\end{gathered} \tag{2:4}$$

Where ρ << 1 and it is called the damping factor.

In the case of SAMARA, the trajectory planning in the task space is used only in the NSM. The trajectory planning for the UAM is executed in the joints space to avoid these problems.

2.3 Non-holonomic constraints

The dynamic equations for the majority of industrial manipulators have nonlinear models. Usually, the nonlinear models have various sets of constraints. Moreover, the constraints can be classified as holonomic or non-holonomic constraints depending on their definitions. Satisfying the specification of each constraint is a critical issue from a control point of view.

The first class is the holonomic constraint and it has the form shown in Eq. (2:5).

$$h_i(q) = 0, \qquad i = 1, \dots, K < n \tag{2:5}$$

Where n is the number of generalized coordinates. The most famous sources for the holonomic constraints in the mechanical structure are the revolute and the prismatic joints [Luc-95, p. 278]. Moreover, this type of constraint reduces the DOFs for the mechanical system, if it is used.

The second class of constraints is the non-holonomic constraint. Being contrary to the holonomic constraint, this class of constraint reduces the control space without affecting the DOFs. As a result, the control problem is expected to be more difficult. The non-holonomic constraint is classified to first and second order non-holonomic constraints.

The first order non-holonomic constraint has a general form shown in Eq. (2:6). In addition, this class is a constraint on the generalized coordinates and their velocities. This type of constraint is non-integrable, i.e., it cannot be solved by integration.

The first order non-holonomic constraint is also called as velocity constraint. Usually, it is found in the dynamic equations for the mobile robot, and vehicles. Therefore, the bicycle model for a car-like robot is an example of the first order non-holonomic constraint see Figure 2:4 [Luc-95, p. 299]. The same model is used here to explain the meaning of this type of constraints.

$$h_i(q, \dot{q}) = 0, \qquad i = 1, \dots, K < n \tag{2:6}$$

The benefit of the bicycle model for a car is to simplify the dynamic model. The idea of this model is to replace the front two wheels into one front wheel located at the midpoint of the axis between the two wheels. The same idea is also applied to the two rear wheels.

The generalized variables for the system are $q = (x, y, \theta, \emptyset)$, where the front wheel has the ability of steering with angle $\emptyset$, while θ measures the orientation from the x-axis. In addition, (x, y) are the distances between the base frame to the position of the rear wheel in x and y directions respectively. Consequently, the coordinates for the front wheel is calculated in Eq. (2:7).

$$\begin{aligned} x_f &= x + l.\cos(\theta) \\ y_f &= x + l.\sin(\theta) \end{aligned} \tag{2:7}$$

By using Eq. (2:7), the first order non-holonomic constraints in the system are shown in Eq. (2:8).

$$\begin{aligned} &\dot{x}\sin(\emptyset + \theta) - \dot{y}\cos(\emptyset + \theta) - l\,\dot{\theta}\cos(\emptyset) = 0 \\ &\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0 \end{aligned} \tag{2:8}$$

Several control solutions based on different theories have been implemented to control different mechanical applications that have first order non-holonomic constraints, as shown in these references [Luc-95], [Sid-08], [Tok-14], and [Kal-16].

$$h_i(q, \dot{q}, \ddot{q}) = 0, \qquad i = 1, \dots, K < n \tag{2:9}$$

The second class of the non-holonomic constraint is the second order non-holonomic constraint. This class is called the acceleration constraint, because this class of constraints is a constraint on accelerations, velocities, and the generalized coordinates [Ane-03, p. 9]. It has a general formula shown in Eq. (2:9). In addition, the second order of non-holonomic constraint is a non-integrable constraint. This class exists in diverse applications, e.g., under-actuated robots, underwater vehicles, vertical take-off and landing aircrafts, and acrobat systems.

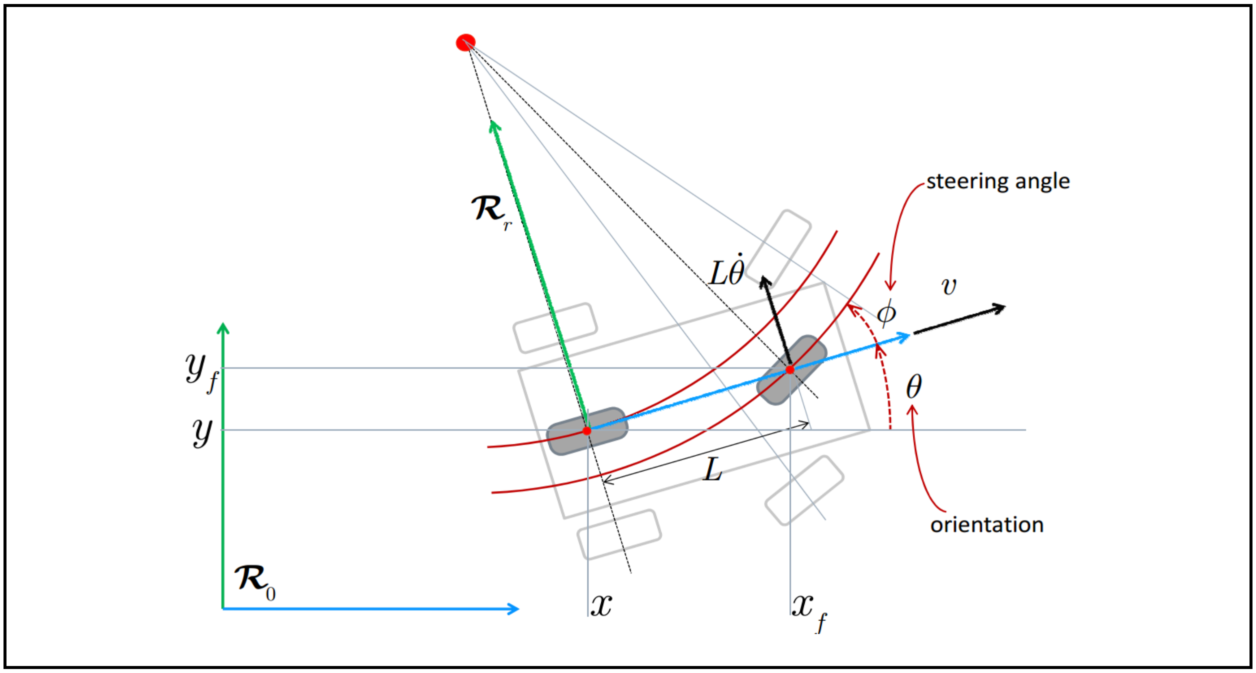


Figure 2:4: Car-like robot –gray wheels are the bicycle model [Bon-12, p. 38].

In the case of the under-actuated manipulator, the joint that utilizes the actuator is called the active joint. On the other hand, the joint that does not use an actuator is called the passive joint and it has a second order non-holonomic constraint. Generally, the form of the dynamic equation/s for this/these link/s is/are shown in Eq. (2:10).

$$m_{pa}(q)\ddot{q}_a + m_{pp}(q)\ddot{q}_p + h_p(q,\dot{q}) = 0 \tag{2:10}$$

Where $q_a(\mathrm{t}) \in \mathrm{R}^{\mathrm{a}}$ is the generalized coordinates for the 'a' active axes. While $q_p(\mathrm{t}) \in \mathrm{R}^{\mathrm{p}}$ is the generalized coordinates for the 'p' passive axis/es; therefore, $q = (q_a, q_p)$. Moreover, $h_p(q,\dot{q})$ is the Coriolis, centrifugal, gravity, and damping terms for the passive axis/axes equation/s.

Undoubtedly, the systems that have this type of constraint are more difficult to control. Different solutions have been suggested to solve several mechanical applications that have this type of constraint; some of these solutions are discussed in [Alb-16], [Alb-15], [Ane-03], [Mar-08], [Olf-01], [Spo-94], and [Ted-09].

2.4 Motion

In this section, three types of motions occurring in SAMARA are presented. These motions are the UAM, the self-motion known also as the NSM, and the end-effector motion. UAM and NSM are implemented in the MAS only, while the end-effector motion is the motion of SAS. This section clarifies how the SAMARA robot can execute the pick and place tasks by using these motions.

The trajectory of the robot is similar to a video tape. In each portion of this tape, there is a specific phase of motion to execute a specific task. Generally, there are four different types of phases used in this trajectory tape as shown in Figure 2:5. The first phase is the start motion (ST) and the final phase is the end motion (EM) in the trajectory tape. Both of them use the fifth order polynomial trajectory as a trajectory planning method. Usually, these phases of motion occur only once. SAMARA has two other phases of motion for pick and place cycle and each phase is divided into two separate motions consisting of the NSM and the UAM. Both motions use a different algorithm for trajectory planning. Usually, these two phases are repeated periodically depending on the number of pick and place tasks as shown in Figure 2:5.

2.4.1 Under-actuated

The under-actuated system is a system that has more DOFs than the number of control inputs [Olf-01, p. 15]. In the case of the under-actuated robot, the passive axes are not directly controllable by direct actuators. These passive joints utilize the coupling between the passive axes and the active axes to control their motions. This class of mechanisms is used in wide applications, because it saves cost and energy.

In the case of SAMARA during the UAM, the end effector moves from the initial point in the station A to the final point in the station B or vice versa. Station A and/or station B could be a conveyor belt. The robot usually grips the payload/object or releases it in the processing points (the points where the NSM occur). Minimizing the energy consumption during this phase is a feature of SAMARA. This is achieved by developing a trajectory planning algorithm to solve the optimization problem for minimizing the power consumption. During this phase, the third axis is treated like a passive axis. In addition, the boundary values for the UAM are known previously, because they are calculated in the NSM phase [Alb-15]. This problem is known in mathematics and is called the two-point boundary value problem (TPBVP).

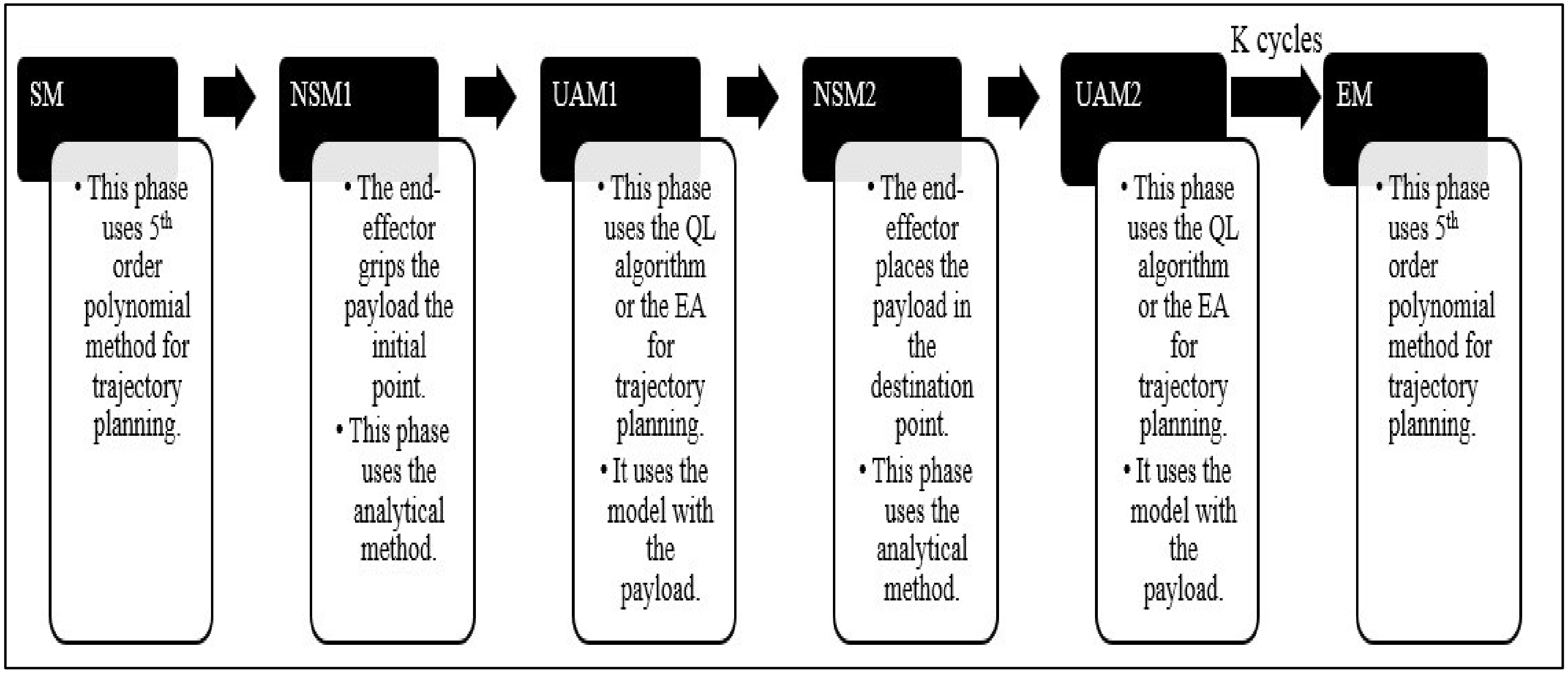

K cycles
SM
• This phase uses 5th order polynomial method for trajectory planning.
NSM1
• The end-effector grips the payload the initial point.
• This phase uses the analytical method.
UAM1
• This phase uses the QL algorithm or the EA for trajectory planning.
• It uses the model with the payload.
NSM2
• The end-effector places the payload in the destination point.
• This phase uses the analytical method.
UAM2
• This phase uses the QL algorithm or the EA for trajectory planning.
• It uses the model with the payload.
EM
• This phase uses 5th order polynomial method for trajectory planning.

Figure 2:5: Motion phases for k cycles of 2nd prototype of SAMARA robot (trajectory tape).

According to [Bre-10], the 1st prototype of SAMARA used the EA to increase energy saving. However, the disadvantages of using the EA are the computation time for solving the optimization problem by using EA, which lasted for two hours in the case of low error [Bre-13, p. 108], and the accuracy of the position of the NSM, which reached up to several centimetres [Bre-10]. There is a poor match between the simulation results and the experimental results [Bre-13, p. 134].

The model used in the robot is changed for each UAM phase and this depends on whether the end effector grips a payload or not. Consequently, two models have been developed for the 2nd prototype of SAMARA robot. The reason for using two models instead of one model for developing the trajectory planning is that in this case, the algorithm can take into account the inertia and the mass of a payload during the motion and the aim is to develop the achieved accuracy, the precision, and the repeatability for the robot.

A general overview of the derivation of the dynamic model for the robot when the end effector is not picking a payload is discussed here for MAS only. More details concerning this model are shown in Appendix 9.2.

The general dynamical model for the planar robot is shown in Eq. (2:11).

$$M\big(q(t)\big)\ddot{q}(t) + C\big(q(t),\dot{q}(t)\big) + D\big(\dot{q}(t)\big) = U(t) \tag{2:11}$$

Where n is the number of axes in the MAS, $q(\mathrm{t}) \in \mathrm{R}^{\mathrm{n}}$ is the generalized coordinates in the MAS. $M\big(q(t)\big) \in R^{n*n}$ is the inertia matrix, $C\big(q(t),\dot{q}(t)\big) \in R^{n}$ are the Coriolis and centrifugal forces, $D\big(\dot{q}(t)\big) \in R^{n}$ are the damping and the friction moments, and $U(t) \in R^{n}$ is the torque vector.

Based on the above (2:11), the dynamic model is expanded in Eq. (2:12) to recognize more details, especially to clarify the active and passive terms in Eq. (2:11). For simplification purpose, the notation t will be dropped in the writing process but the meaning still exists that all parameters are functions of time.

$$\begin{bmatrix} m_{aa}(q) & m_{ap}(q) \\ m_{pa}(q) & m_{pp}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_p \end{bmatrix} + \begin{bmatrix} C_a(q,\dot{q}) \\ C_p(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} D_a(\dot{q}) \\ D_p(\dot{q}) \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix} \tag{2:12}$$

Where a is the number of active axes in the MAS; p is the number of passive axes in the MAS. Consequently, $\mathrm{c_a}(\mathrm{q(t)},\dot{q}(t)) \in \mathrm{R}^{\mathrm{a}}$ and $c_p(q(t),\dot{q}(t)) \in R^{p}$ are the Coriolis and centrifugal forces for the active and passive axes in the MAS respectively. In addition, $d_a(\mathrm{q(t)},\dot{q}(t)) \in \mathrm{R}^{\mathrm{a}}$

and $d_p(q(t), \dot{q}(t)) \in R^p$ are the damping and friction moments for the active and passive axes in the MAS respectively. Finally, $\mathrm{T} \in \mathrm{R}^{\mathrm{a}}$ denotes the vector containing motor torques in the MAS.

The second row in Eq. (2:12) describes the dynamical equation of the passive axis and it is clear this type of constraint is a second order of non-holonomic constraint. Additionally, the sub-blocks in Eq. (2:12) are described in Eq. (2:13).

$$q_a = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, q_p = q_3 \qquad (2:13)$$

$$m_{aa}(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, m_{ap}(q) = \begin{bmatrix} m_{13} \\ m_{23} \end{bmatrix}, m_{pa}(q) = [m_{31} \quad m_{32}], m_{pp}(q) = [m_{33}]$$

$$C_a(q, \dot{q}) = \begin{bmatrix} C_1(q, \dot{q}) \\ C_2(q, \dot{q}) \end{bmatrix}, C_p(q, \dot{q}) = C_3(q, \dot{q})$$

$$D_a(\dot{q}) = \begin{bmatrix} D_1(\dot{q}) \\ D_2(\dot{q}) \end{bmatrix}, D_p(\dot{q}) = D_3(\dot{q}),$$

$$T_a = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Consequently, the state space representation of the first model is shown in Eq. (2:14).

$$x_1 = q \quad , \quad x_2 = \dot{x}_1 = \dot{q} \quad , \quad \dot{x}_2 = \ddot{q} \qquad (2:14)$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ M^{-1}(x_1) * (T - C(x_1, x_2) - D(x_2) \end{bmatrix} = f_1(x, u, t)$$

Various control theories, algorithms, and methods to execute the control on different mechanical systems are shown in Table 2:2.

2.4.2 Null space

The null space motion occurs when the orientation and the position of the end effector are fixed while the joints of the robot are changing their configurations [Bur-89].

The redundant robot, e.g., SAMARA has the ability to execute the null space motion and this improves the dexterous ability for the end effector [Shi-08, p. 413]. Besides, it improves the obstacle avoidance due to the fact that redundant robot has two spaces of motion: the task-space and the self-space.

Hollerbach et al. introduced the trajectory planning for the NSM by using four algorithms in [Hol-87]. Hollerbach et al. succeeded in minimizing the joint torque during the NSM. In addition, they compared the unweighted pseudo-inverse algorithm, inertia-weighted pseudo-inverse algorithm, unweighted null space algorithm, and weighted null space algorithm from different perspectives, such as power consumption and the stability of the motion in different ranges. According to [Hol-87], in the results for small and mid-range of trajectories, the null space algorithms are better than pseudo-inverse algorithms based on the power saving. However, for a long trajectory, the only method that succeeded and superseded to avoid the instability problem is an unweighted pseudo-inverse method; the reason is that this method is a kinematic method. This leads to reducing power consumption as a feature of unweighted/weighted null space algorithms, but the stability is a crucial problem. In addition, Brett analyzed the weighted null space algorithm to check if it is possible to use it as trajectory planning algorithm for the NSM in SAMARA robot. It is concluded that this algorithm cannot calculate the time of the NSM if the initial and final states are known, which is a critical problem [Bre-13, p. 104].

Table 2:2: Literature review of the under-actuated systems.

Branch of work	Reference	DOF	Model[3] :S/E	Control
Acrobat	[Spo-94]	2	E	Partial feedback linearization
Planar under-actuated biped robot	[Che-04]	7	S	Optimal reference trajectories generation and partial feedback linearization
Under-actuated underwater vehicle-manipulator	[Moh-12]	5	E	Null Space Control
Unmanned helicopter	[Cas-07]	6	S	MPC
Under-actuated mechanisms	[Ted-09]	-	S	Several methods
Under-actuated marine surface vessels	[Oh-10]	3	S	MPC
Groups of autonomous surface vessels	[Fah-07]	3	-	Nonlinear model predictive control (NMPC)
Under-actuated robot (1st prototype of SAMARA robot)	[Bre-08],[Bre-10], [Bre-13], [Glo-14]	3	E	EA + PID position controller
Under-actuated robot (2nd prototype of SAMARA robot)	[Alb-15], [Alb-16]	5	E	QL+AMPC or QL+PID position controller

[3] E: denotes to experimental results. S: denotes to simulation results

As mentioned in subsection 2.2, one of the essential principles to preserve the velocity for one axis must be uniform (the first axis in the case of SAMARA) at least for one axis [Bre-08]. Hence, one of the requirements is that the velocity of the first axis must be uniform during the null space motion and the computation for trajectory planning in this phase can be realized in a real time framework. Therefore, the analytical solution for calculating the NSM has been developed as an algorithm for trajectory planning in this phase [Bre-13, p.101 ff.].

In the first phase of this research project, Brett compared the weighted null space algorithms with analytical solution depending on several criteria, such as a uniform velocity for the first axis, online capability, the possibility to know the NSM period, and power consumption. In all criteria, the analytical solution is better than weighted null space algorithms except the power consumption [Bre-13, p. 104]. However, Brett suggested the time for the NSM as 30 ms to reduce the power consumption during this phase [Bre-13, p. 106]. More information concerning the model and the equations of the analytical solution are shown in [Bre-13, p.101 ff.].

The duration time of the NSM is an input parameter for the NSM algorithm and it can be selected in the range of [30 – 200] ms depending on the weight of the payload. Experimental tests have verified that the execution of pick or place tasks are possible within this period. As an input for the algorithm, the pick-and-place points have been selected before and a traditional fully actuated approach was used to calculate the trajectory for this portion; see Figure 2:6. Afterward, the angular configurations at the end and at the beginning of the NSM together with their velocities are used as the inputs for the UAM.

During this phase of motion, the end effector grips the payload/object from the initial point and it releases the payload/object in the final point. This process is a periodical process, but with different coordinates for the initial and final points if necessary. Usually, these points can be stored in a database or can be calculated in a real time environment as shown in Figure 2:6.

Finally, several publications discussed the control in the phase of NSM like Shibata et al. who proved a position controller for the task space and a PID controller for a null space that has the same capability as null space observer to stabilize the system [Shi-08, p.414 ff.]. Another reference is Mohan et al. who evolve the null space controller and extended Kalman filter disturbance observer for autonomous underwater-manipulator system to execute the target tasks [Moh-12].

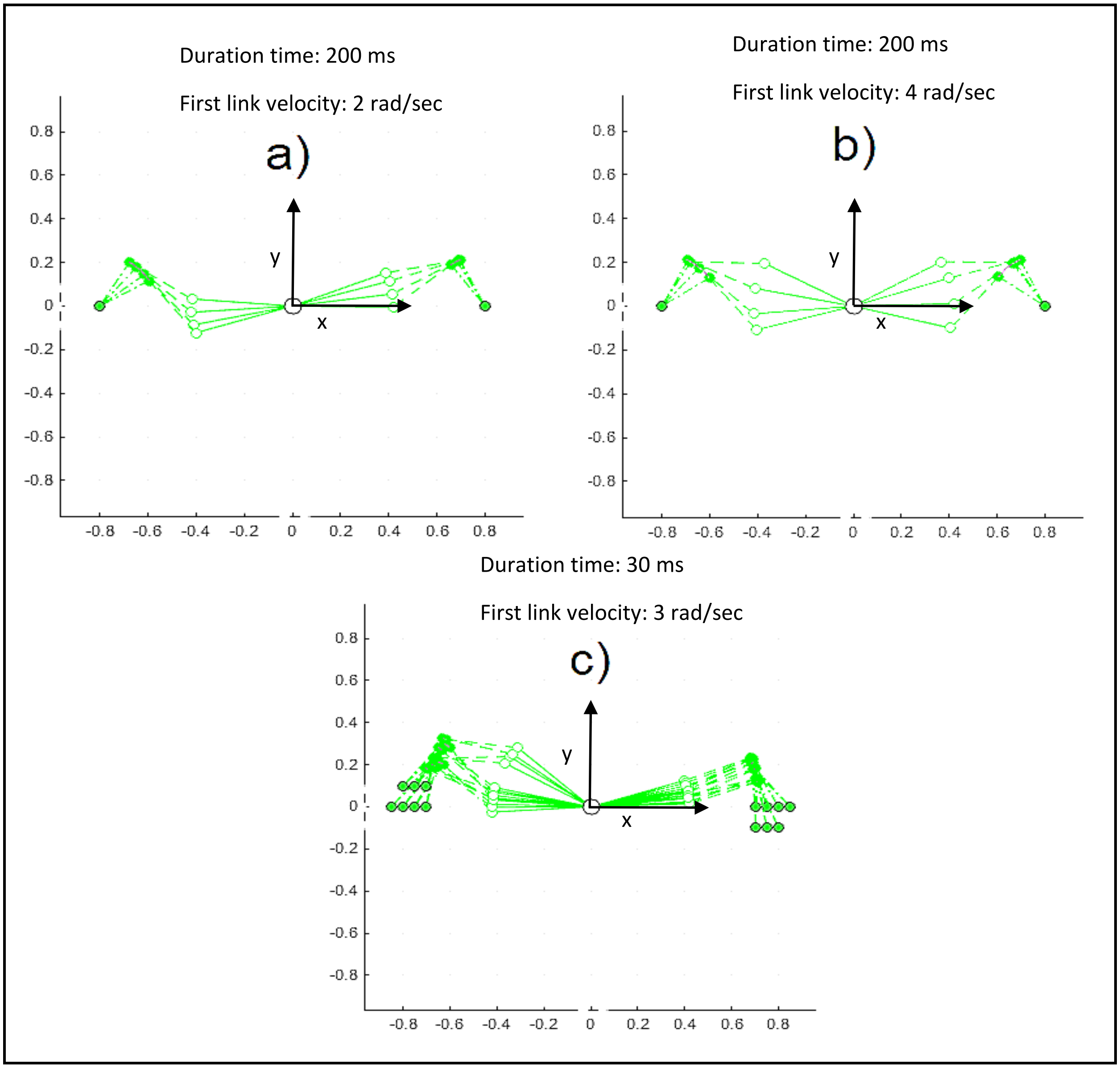


Figure 2:6: MAS configurations of SAMARA during several NSMs.

2.4.3 End effector

One of our goals is to provide SAMARA prototype the capability to carry payloads up to 5.5 kg. As a result, the end-effector is designed to pick the payload from the initial point; then it must place it in the destination point. This process is a periodic process to complete the entire pick and place tasks.

The end effector has two DOFs and a suction gripper system to carry the payload during the NSM. The first DOF is similar to the functionality of the prismatic joint to lift the payload upward or downward. This motion has been achieved by using the ball screw mechanism. The second DOF is similar to the functionality of the revolute joint. The second DOF is used to compensate the error in the orientation of the payload during the motion and this is achieved by using ball spline mechanism as shown in Figure 2:7. Two motors have been used to achieve the required motions. The ball screw mechanism uses only one motor to perform the upward motion or the downward motion depending on the direction of rotating of the motor, while the ball spline mechanism uses two motors to compensate the orientation error.

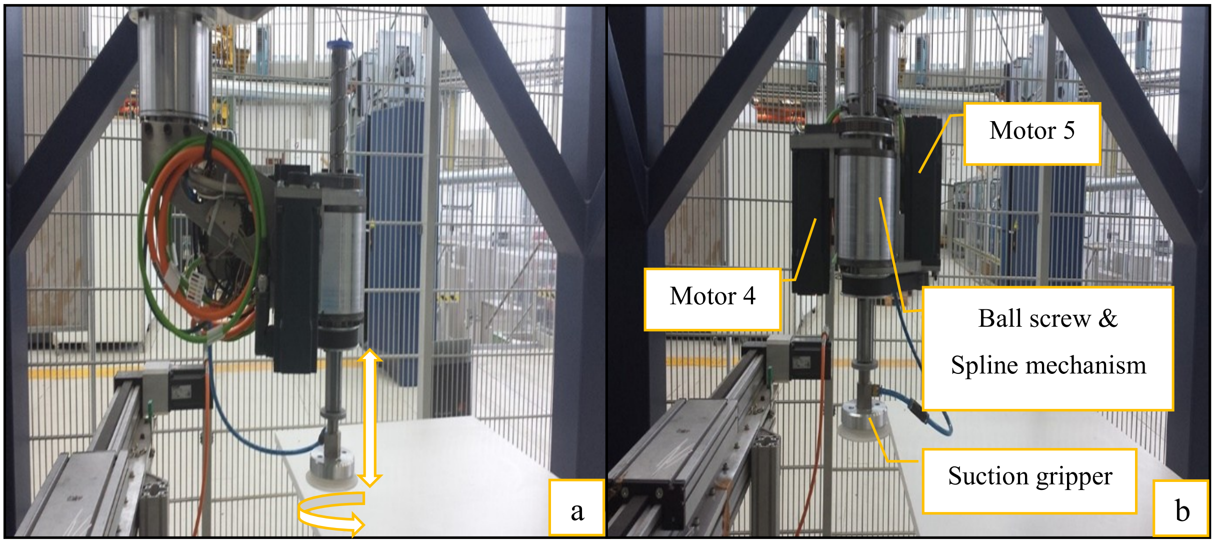


Figure 2:7: The end effector description a) directions b) the major components

Design of the end effector is a challenging problem in the case of SAMARA. The reason for that is related to the following criteria, firstly, the links of the robot are moving continuously and it is difficult to pick and place payloads while the links of the robot are rotating continuously. Therefore, to solve this problem, the null space motion integrated into the trajectory tape to permit the end effector giving the opportunity to execute pick and place tasks. The main problem is the maximum suitable time for the NSM. The main question here is, how the end effector can perform the pick and place tasks in this small period? The available solution is to perform the upward motion or downward motion during the phase of UAM. While the orientation for the payload has been corrected continuously. Otherwise, the end effector cannot satisfy the requirements for lifting the payload upward or downward and compensating the error in the orientation without using strong actuators. This solution decreases the power saving and maximizes the allowable payload.

The point to point (PTP) synchronized path planning algorithm has been developed based on the third order polynomial planning method to perform the required motions for each axis independently in the joint space; more details concerning this method are shown in [Jaz-07, p. 569 f.]. After that, these trajectories have been mapped to the actuator space taking into account the gear ratios and other parameters. Next, the superposition principle is used to calculate the trajectory of each motor; see Figure 2:8. Finally, both motors can track these trajectories by using the PID controller for each of them.

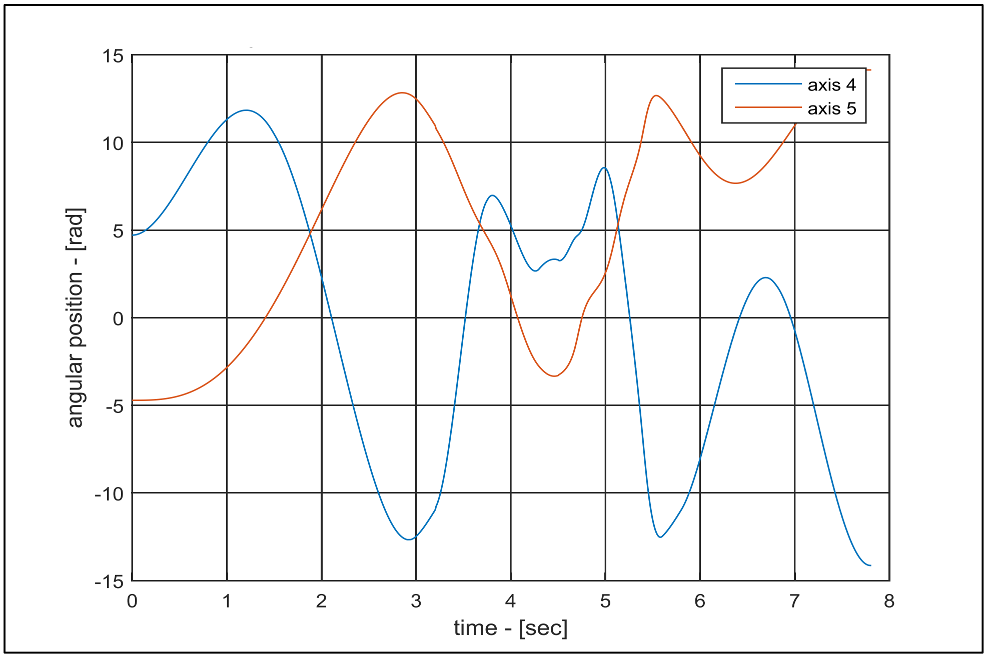


Figure 2:8: Position trajectories for the end effector in the actuator space.

2.5 Trajectory planning

One of the primary steps for implementing a successful motion planning for robotics is to select the adequate space for the trajectory planning, because each space has different requirements, advantages, and disadvantages; e.g., if the trajectory is planned in the task space, it is important to avoid the singular configurations or at least to use Eq. (2:3) or Eq. (2:4) to solve the problem of Jacobian singularity.

This section is a survey of the trajectory planning methods for different mechanical applications based on minimizing the cycle time and/or minimizing the energy consumption. In addition, this section formulates the problem precisely, for uses in chapter 4 to solve the problem. Moreover, this section distinguishes between the deterministic methods and the stochastic methods to identify which one is adequate to be employed in the case of trajectory planning for the UAM for the MAS links, only in the 2nd prototype of SAMARA robot.

2.5.1 Problem formulation

The problem formulation depends on the goals of this research. In the case of the 2nd prototype of SAMARA robot, two substantial goals must acquire an efficient motion performance. These goals are to minimize the power consumption for the robot and reduce the cycle time to increase the efficiency of the handling robot.

Usually, the cost function (CF) also called the performance index is used to measure the achieved goals in the optimization problem. The idea is to use the optimization algorithm to generate optimal trajectories for SAMARA. These trajectories satisfy the central goals of this research.

The performance index for minimizing the cycle time is shown in Eq. (2:15), while the performance index for minimizing the inputs, i.e., the power consumption, is shown in Eq. (2:16). Furthermore, another performance index can be used to minimize the inputs and the error between the desired reference and the actual response in the tracking problem is shown in Eq. (2:17).

$$CF_1 = \int_{t_0}^{t_f} dt \tag{2:15}$$

$$CF_2 = \int_{t_0}^{t_f} U^T R U dt \tag{2:16}$$

$$CF_3 = \int_{t_0}^{t_f} ((X - ref)^T Q (X - ref) + U^T R U) dt \tag{2:17}$$

Where, r is the number of states, n is the number of inputs, t_f is the final time, $X \in R^r$ denotes to the state vector, R $\in R^{n*n}$ is the positive definite matrix, Q $\in R^{r*r}$ is the positive semi-definite matrix, and $ref \in R^r$ is the reference command.

The notation CF_2 called in the rest of this dissertation as CF. The second performance index is used in the case of SAMARA.

Two fundamental points must be clarified before starting to solve the optimization problem. These points are:

- Are the final states $x(t_f)$ fixed states or free states?
- Is the final time t_f specified or free?

These two points are essential points to solve the optimization problem, because the previous points determine if the transversality conditions must add to the optimization problem or not. The initial states $x(t_0)$ and the final states $x(t_f)$ for SAMARA are known from the start/end NSM phase; see Figure 2:5. Furthermore, the final time for the UAM is known also, subsequently in section 4.2 will be clarified how the cycle time can be reduced. Thus, both $t_f$and $x(t_f)$ are specified. As a result, it is not necessary to satisfy the transversality conditions in this case [Kir-04, p. 200].

The trajectory planning for the UAM occurs in the joint space to avoid the problems of singularity. Moreover, the optimization problem is shown in Eq. (2:18) and it uses two models to describe the dynamic equation for the robot as mentioned in section 2.4.1. The first model is shown in Eq. (2:14) when the end effector is not gripping a payload. While the second the model applies when the end effector grips a payload.

$$J = \min \frac{1}{2} \int_{t_0}^{t_f} (U^T R U) dt \qquad (2{:}18)$$

$$\text{sub to: } \dot{x}_{model\,1} = \mathrm{f}_1(\mathrm{x}, \mathrm{u}, \mathrm{t}) \qquad \text{or} \qquad \dot{x}_{model\,2} = \mathrm{f}_2(\mathrm{x}, \mathrm{u}, \mathrm{t})$$

$$U_{min} < U < U_{max}$$

Let us focus on methods used to solve the optimization problem for several mechanical applications in the following two subsections. These methods are classified in general to deterministic methods and stochastic methods. Deterministic methods calculate the optimal solution based on a calculus of variations and the fundamental theory, while the stochastic methods find the optimal point based on probabilistic science.

2.5.2 Deterministic methods

The deterministic methods use the calculus of variations to find the optimal solution. The optimization problems are classified as a constraint or an unconstraint optimization problem. The last class is solved by applying the necessary conditions and the sufficient condition to solve the Hamiltonian equation discussed in [Kir-04, p. 188], whereas the constraint optimization problem is solved by the Pontryagin minimum principle, which is an adaptation of the previous solution in the case, if there is a constraint on the inputs or in the states [Kir-04, p. 233].

Generally, these methods find the optimal solution by calculating the gradient vector or the Hessian matrix. Therefore, deterministic methods to solve the constraint optimization problem like steepest decent method, variation of externals method, QL algorithm, and gradient projections method have been investigated to select the most adequate method based on different criteria [Kir-04, p.329 ff.]. Consequently, Albalasie et al. used the QL algorithm as trajectory planning algorithm to minimize the power consumption for the under-actuated robot [Alb-15]. Finally, Tokekar et al. present how to solve a constraint and unconstrained optimization problem to minimize the energy consumption for a car-like robot, which is similar to our case [Tok-14].

2.5.3 Stochastic methods

This section presents the literature review for the two famous stochastic algorithms. The first algorithm is the EA and the genetic algorithm (GA).

The EA algorithm is developed to minimize the power consumption for the 1st prototype of SAMARA robot during the UAM with the support of using a benchmark-indicator to analyze the consumed power [Bre-10]. The benchmark-indicator used the payload as single body system motion between the two processing points for the minimum path. The reason for using this type of motion is because it is familiar in the handling applications and it has constant acceleration and deceleration phase only. However, the experimental results show the power consumption for 1st prototype of SAMARA is three to five times higher than the simulation results in the best case [Bre-13, p. 133 f.]. Brett argued this result by clarifying that his model ignores any model of frictions in all joints and the simulation model used the CAD parameters as a rough estimation of the dynamic parameters; moreover, he argued other reasons for more details see [Bre-13, p. 134]. Subsequently, as a preliminary work for this dissertation, the friction model has been added to the simulation model but in the majority of cases the EA diverged, i.e., the algorithm cannot find the optimal solution for the optimization problem.

Another stochastic algorithm, which is used widely in different applications, is the GA; e.g., it is used as a path planning algorithm for space manipulator where the aim is to minimize the joints torque [Hua-06]. Nevertheless, the GA is more suitable for the nonlinear models [Hua-06]. In addition, GA can be used in path planning for the redundant manipulators to minimize the joint velocities, joint accelerations, largest joint displacement, joints torque, and total power joint consumption as stated in [Mar-09].

Finally, to achieve the goals of the research, it is desirable to use one of the deterministic methods, because it is faster than the stochastic methods in computation.

2.6 Model predictive control

MPC is an advanced control technique, which uses quadratic programming (QP) problem to calculate the optimal solution. It has the ability to minimize the cost function. In the majority of the cases for the robots, the cost function is the inputs torque/force, the error between the reference and the actual responses, as well as the change rates of the inputs.

According to [Cam-07, p. 3], MPC applies three fundamental concepts to calculate the optimal solution. Firstly, it employs the mathematical equations for the system to predict the future behaviour of a system on the prediction horizon (P), in contrast to other controllers. Secondly, MPC computes the control action (C) and solves the optimization problem to minimize the cost function as shown in Figure 2:9. Thirdly, MPC utilizes only the current values of inputs C(k) as a control signal for the current sample to use them, and this process is repeated continuously after rejecting the future control signal.

In agreement with [Cas-07], several desirable properties from the control point of view have been achieved by using MPC, e.g., this controller can take into consideration the physical constraints such as the maximum torque, maximum change rates of the inputs, and other constraints in the states. It is applied for multiple systems without sophisticated modifications on the control structure as shown in [Fah-07]. Also, it has the ability to handle the uncertainty or the inaccuracy in a system model, i.e., the controller has the capability to become a robust controller [Cam-07, p.217 ff.].

The selected form for the cost function effects on the performance of the mechanical application is shown in Table 2:3. Because the select form can minimize one or more of several parameters, such as the inputs, the change rates of the inputs, the error between the desired reference and the actual response, etc.

Two critical problems are challenging problems in the area of MPC. Firstly, the stability of MPC is not guaranteed, and the reason for that is because it uses the finite horizon to calculate the solution. However, increases in the prediction horizon more than the settling time for the system is an approach that satisfies the stability requirements in the majority cases as shown in [Oh-10], [May-14]. Secondly, the feasibility of MPC is not guaranteed due to significant reasons such as the case where the constraints of the optimization problem are restrictedly hard constraints. However, this problem can be solved by relaxing the constraints if it is possible or redesign the system itself, e.g., if the solution of MPC is infeasible due to the maximum limit torque constraints, i.e., the actuators cannot provide sufficient torques for the robot; therefore, the solution is to replace the current actuators by stronger ones.

The majority of the physical systems have nonlinear models in nature such as robots in general. Robots have fast dynamics. As a result, MPC is not commonly employed in these applications in general, in order to consume more time than other controllers do; the reason being that because MPC, as mentioned before, solves the optimization problem and it predicts the future response that consumes time. Anyway, Duchaine et al. suggested position and velocity control schemes by using MPC considering the suggestions of other researchers to use only the prediction horizon term after neglecting the other terms to reduce the computation time [Duc-08, p.139 ff.].

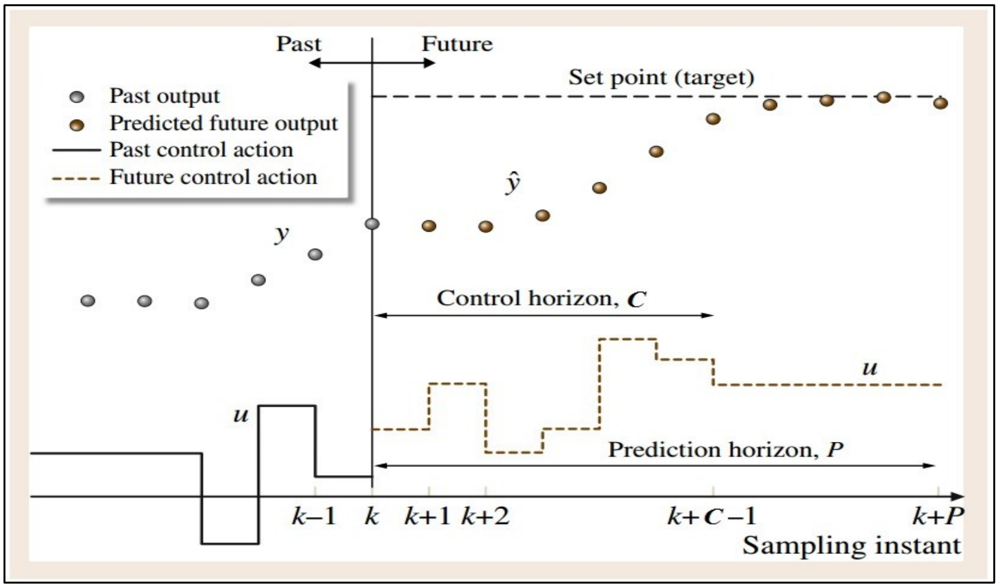


Figure 2:9: Model predictive control concept [Edg-09, p. 537][4].

MPC uses the linear model for describing the nonlinear dynamical model for a system. However, occasionally, MPC cannot work accurately due to high nonlinearity of the system. Therefore, it can be modified to employ the nonlinear model, and it is called in this case NMPC. The computational time for MPC is lower than NMPC. A suggested solution to solve the computation

[4] The picture has been Updated

time problem is to simulate the system in the off-line environment; subsequently the state space is divided into specific regions and all these regions are memorized on the database. Then, in the real time environment, an affine function with constant parameters has been used for each region to approximate the solution as demonstrated in [Fia-06].

Table 2:3 clarifies the literature review for the selected performance indexes with their effects on several mechanical systems. These optimization problems are solved by MPC, AMPC, QL and other methods.

Through this survey, MPC is used to control several applications, e.g., it is used to control the unmanned helicopter [Cas-07], and to control the under-actuated marine surface vessels as demonstrated in [Oh-10], while NMPC is used to control multiple autonomous surface vessels as shown in [Fah-07]. Furthermore, explicit MPC is used to control a linear motor in [Fia-06]. Finally, Albalasie et al. developed a control scheme for the under-actuated robot by using the AMPC [Alb-16].

Table 2:3: Literature review for the selected performance indexes and their effects.

Selected performance index	Reference(s)	Effect(s)
Minimize the inputs	[Alb-15], [Bre-10], [Bre-13], [Kir-04, p.259 ff.], [Tok-14], [Hua-06], [Glo-14], [Pae-14].	save more energy
Minimize the time	[Kir-04, p. 240], [Jaz-07, p. 605], [LI-90], [Pae-14].	faster response
Minimize the error, the change rates of inputs, and the inputs	[Alb-16], [Cas-07].	save more energy, and reduce the errors with reference signal taking into account the maximum allowable change rates of inputs
Minimize the error, and the inputs with the capability of obstacle avoidance algorithm	[Fah-07].	avoid obstacles if there is, saves more energy, and reduces error with the reference signal.
Minimize the error, and the inputs	[Oh-10], [Fia-06].	save more energy, and reduces error with the reference signal

3 Requirements of the Efficient Trajectory Planning for the Under-actuated Robot

In the last decades, researchers focused on control and developed trajectory planning algorithms for the under-actuated mechanisms. They focused on this class of mechanisms, because it can save more cost and energy, which is an essential issue for several applications. The fabulous example that clarifies the previous point is represented in the case of the autonomous surface vessel and unmanned aerial vehicles, because both of them are suffering from the capacity of the electrical battery. Therefore, it is desirable to use a class of mechanical structures that can reduce the energy by increasing the timetable for these applications. Therefore, engineers and researchers prefer to use the under-actuated configuration for these applications.

Usually, engineers prefer to use the fully actuated configuration instead of using the under-actuated configuration (according to the knowledge of the author) for developing the industrial robots, because the control and the trajectory planning for the robots using the fully actuated configuration are easier than the control and the trajectory planning for the under-actuated robots. Consequently, under-actuated industrial robots are not utilized for industry, until now.

During the last ten years, the 1st prototype of SAMARA robot has been developed as a prototype of the under-actuated robot; see Figure 1:1. That prototype had not included an end effector to perform the physical pick and place tasks. The goal was to evaluate if it is possible to realize the under-actuated robot in the industry or not and what are the benefits in comparison with other options. That prototype had two phases of motions; the first phase is the NSM and the second phase is the UAM. The trajectory planning during the UAM used the EA to minimize the energy consumption [Bre-13, p. 105].

Unfortunately, the research project faced several obstacles which need to be improved; for instance, the EA typically consumes up to two hours to compute each trajectory [Bre-13, p. 108]. In addition, there is a poor matching between the simulation results and the experimental results. In the best cases, the actual power consumption is three to five times higher than the simulation results [Bre-13, p. 134]. Furthermore, there is no clear methodology to reduce the cycle time for the robot while minimizing the energy. Finally, the achieved accuracy in the 1st prototype of SAMARA was several centimetres for the point where the NSM occurred [Bre-10].

During this research, the 2nd prototype of SAMARA built and designed to solve the problems mentioned before to improve the performance of the robot.

Based on the previous points, the requirements for the efficient trajectory-planning algorithm are focused on reducing the computation time by replacing the EA with another deterministic trajectory-planning algorithm. This improvement satisfies the previous point if it can work in a real time framework or if the computation time still works in the offline environment, but with a small computation time. On the other hand, the new trajectory-planning algorithm must satisfy the positive features for the EA, e.g., the ability of minimizing the energy consumption in the same level approximately. Moreover, another enhancement can be achieved for the trajectory-planning algorithm by introducing a clear approach for reducing the cycle time to increase the handling rate for the robot. Furthermore, the new trajectory-planning algorithm must reduce the error of the power consumption between the simulation and the experimental results to an acceptable range to increase the simulation resolution.

As a result, this dissertation focuses on replacing the EA by the QL algorithm for the trajectory planning to reduce the computation time. Therefore, a comparison between the EA and QL has been executed to evaluate the performance of each method based on different criteria.

This work also focuses on minimizing the power by solving the optimization problem. This problem is solved based on the theory of the calculus of variations to minimize the energy and this is achieved by using QL algorithm to compute the optimal trajectories. This method reduces the computation time dramatically. In addition, the dissertation suggests a clear approach to reducing the cycle time for the under-actuated robot to increase the handling rate and it is used to decrease the cycle time. The considerations also focus on reducing the relative error between the actual response and the theoretical response, as much as possible, to get results that are more accurate. This work develops the accuracy of the 2nd prototype of SAMARA robot to an acceptable range.

Other requirements for developing an efficient controller are summarized in the following points: the ability of the controller to minimize the backlash errors, i.e., the error between the desired trajectories and the actual behaviour of the robot to increase the position accuracy. In addition, the controller must track the torque inputs, which the trajectory planning algorithm calculates or it should have the ability to minimize the inputs to reduce the power consumption. Furthermore, it is desirable if the controller can take into consideration the saturation limits for the torque inputs and the change rates of the inputs. Consequently, it is expected to reduce the relative error between the simulation and experimental results for the power consumption. Therefore, this dissertation presents the optimal control scheme to control the robot based on previous requirements.

The optimal control scheme uses the AMPC to satisfy the requirements for the efficient controller. This controller can minimize the backlash errors, and the input torques if it is needed or it can track the torque inputs as required. In addition, it has the ability take into account the maximum limits for the torque inputs and the change rates of the inputs to achieve better performance.

The AMPC controller is more adequate than using the MPC or the NMPC. The reason is that MPC uses only one linear model for describing the nonlinear dynamics of the system. Therefore, the model is valid only in the domain of attraction around a nominal operating point. Because the NMPC uses a nonlinear model, it is more accurate. As a result, MPC is faster than NMPC concerning computation time. However, if the nonlinearity of the system is too high, the MPC performance becomes unacceptable, while NMPC calculates results that are more accurate. On this basis, AMPC solves the previous problem by using the linear time invariant (LTI) model structure. This model is updated continuously by linearizing the dynamical model of the robot at the current operating point and hence this point is changing continuously.

4 Concept of Control Techniques

This chapter presents solutions to satisfy the essential three goals for this research; minimizing the energy consumption, reducing the cycle time for the robot, and reducing the computation time. Furthermore, this chapter introduces two control schemes for controlling the robot. The first scheme is called the position control scheme. This control scheme has the capability to control the position of the robot and it has the ability to control both velocity and current. The second control scheme is called the optimal control scheme, because it uses the AMPC to control the robot but this scheme has higher capabilities as shown in section 4.5.

The dynamic parameters for the 2nd prototype of SAMARA robot have been identified rather than using the CAD parameters like the 1st prototype of SAMARA robot. Therefore, the friction model and the friction parameters have been added to the dynamic model for the robot to acquire results that are more accurate. The expected result of performing these steps is to reduce the relative error between the simulation and experimental results as verified in chapter 5.

4.1 Minimizing the energy consumption

Increasing the energy saving for the robot is a challenging problem. Several approaches have been suggested to solve this problem, by optimizing the response of the robot or by using other methods based on managerial concepts. This work uses the approach of optimizing the trajectories for the robot to minimize the energy consumption. Hence, there is an optimization problem that must be solved to satisfy the goal.

The optimization problem is discussed in section 2.5.1. This section clarifies the optimization problem in more detail. The selected formula of the optimization problem is shown in Eq. (2:18). This formula has been chosen, because it has the ability to minimize the inputs, i.e., maximize the energy saving. In addition, the 2nd prototype of SAMARA robot uses the three principles to reduce the energy as mentioned in section 2.2 and in [Bre-08]. These principles also apply the idea of the effective use of the robot dynamics.

The 2nd prototype of SAMARA robot uses the optimization problem to calculate the trajectories of the UAM. As mentioned in subsection 2.5.2 and in subsection 2.5.3, there are two classes of methods to solve the optimization problem. In this work, optimizing the trajectories for the robot by using one of the deterministic methods, has been chosen due to reducing the computation time.

Therefore, QL algorithm has been developed to achieve this major goal. This algorithm is one of the deterministic methods that uses the calculus of variations to solve the optimization problem by converting the problem to the Hamiltonian form.

As a preliminary work for solving the optimization problem, the following tasks are executed. The NSMs have been calculated on the pick and place points; see Figure 2:6 and Figure 4:1. Therefore, the values of the angular positions, the values of the angular velocities, and the values of angular accelerations are known in the start/end of the NSMs, and during the NSMs, while the angular positions and the angular velocities in the start/end NSMs are used as boundary conditions for calculating the UAM and this is called TPBVP. In addition, the most adequate model for the robot has been selected for use in the next procedures.

Let us present how the Eq. (2:18) is solved to calculate the optimal trajectories. To solve Eq. (2:18), the problem converts to the Hamiltonian equation as shown in Eq. (4:1) after selecting the proper dynamical model.

$$\mathcal{H} = \frac{1}{2} * [(U(t)^T * R * U(t)] + \sum_{i=1}^{r} \lambda(t)_i * f(x(t), u(t), t)_i \qquad (4:1)$$

Where r is the number of states in the MAS, R is the positive definite matrix, $\lambda(t)$ is the co-state vector, $U(t)$ is the torque vector in the dynamic equation for the MAS, and $f(x(t), u(t), t)$ is the state space representation for the MAS also. According to [Kir-04, p. 188], the fundamental theorem of the calculus of variations required to satisfy three necessary conditions and the sufficient condition for calculating the optimal solution. The three necessary conditions are shown in Eq. (4:2), Eq. (4:3), and Eq. (4:4).

$$\dot{x}^*(t) = \frac{\partial \mathcal{H}}{\partial \lambda} = f(x^*(t), u^*(t), t) \qquad \text{, for all } t \in [t_0, t_f] \qquad (4:2)$$

Eq. (4:3) calculates the optimal derivative for the co-state variables, while the optimal torque inputs minimize the performance index, i.e., minimize the energy consumption have been calculated from Eq. (4:4).

$$\dot{\lambda}^* = -\frac{\partial \mathcal{H}}{\partial x} \qquad \text{, for all } t \in [t_0, t_f] \qquad (4:3)$$

$$\frac{\partial \mathcal{H}}{\partial U} = 0 \quad \xrightarrow{yields} \quad u^* = opt.input \qquad (4:4)$$

The QL algorithm uses Eq. (4:2), Eq. (4:3), and Eq. (4:4) to find the optimal trajectories for the robot. Furthermore, H is a quadratic in $u(t)$. Therefore, the solution is a local minimum optimum solution, because the result of the sufficient condition is positive definite matrix. Their result is shown in Eq. (4:5) [Kir-04, p. 233 f.].

$$\frac{\partial^2 \mathcal{H}}{\partial U^2} = R > 0 \tag{4:5}$$

The optimal torque inputs must not exceed the maximum limit for the motors. Nevertheless, if this is according to the Pontryagin minimum principle used to solve this problem, the Pontryagin minimum principle replaces Eq. (4:4) by another necessary condition as shown in Eq. (4:6) to solve the constraint optimization problem. Eq. (4:4) included internally in the Eq. (4:6) in case the inputs are less than the maximum allowable inputs [Kir-04, p. 233].

$$H(x^*(t), u^*(t), \lambda^*(t), t) \leq H(x^*(t), u(t), \lambda^*(t), t) \quad \text{, for all admissible } u(t) \text{ and for all } t \in [t_0, t_f] \tag{4:6}$$

Finally, the optimal inputs for the robot, which are minimized in Eq. (2:18), have the form shown in Eq. (4:7).

$$u_1^* = \begin{cases} u_{\max\_1} \\ opt.\,input_1, \\ u_{\min\_1} \end{cases} , u_2^* = \begin{cases} u_{\max\_2} \\ opt.\,input_2 \\ u_{\min\_2} \end{cases} \tag{4:7}$$

Let us suggest a scenario to perform multi pick and place tasks to clarify the concept. The tasks focused on picking a series of payloads from different points A, C, and E, as shown in Figure 4:1. Then, the robot places these payloads in the following points: B, D, and F. Firstly, the NSMs have been calculated in the entire points, e.g., see the NSMs for point A and point B have been calculated in Figure 4:1. Secondly, choose the correct dynamical model for the required UAM, which is the second state space model ($\dot{x}_{model\,2}$) for the robot in case of movement from point A to point B, while the correct model for the motion from point B to point C is the first state space model ($\dot{x}_{model\,1}$). Thirdly, use the correct boundary conditions for each UAM. In the case of movement between point A and point B, use the boundary conditions of the end of NSM in the point A, which is the configuration in red colour as shown in Figure 4:1. In addition, the start configuration for the NSM is point B, which is the configuration in the light yellow colour in the same figure. Fourthly, select a suitable final time to perform the task; later in the next section, a clear methodology presents how to reduce this time. Fifthly, use QL algorithm to calculate the optimal trajectories, which minimize the energy consumption during the UAM. Finally, update the boundary conditions after updating the next two points. Repeat these procedures for calculating the other UAMs between other points and so on.

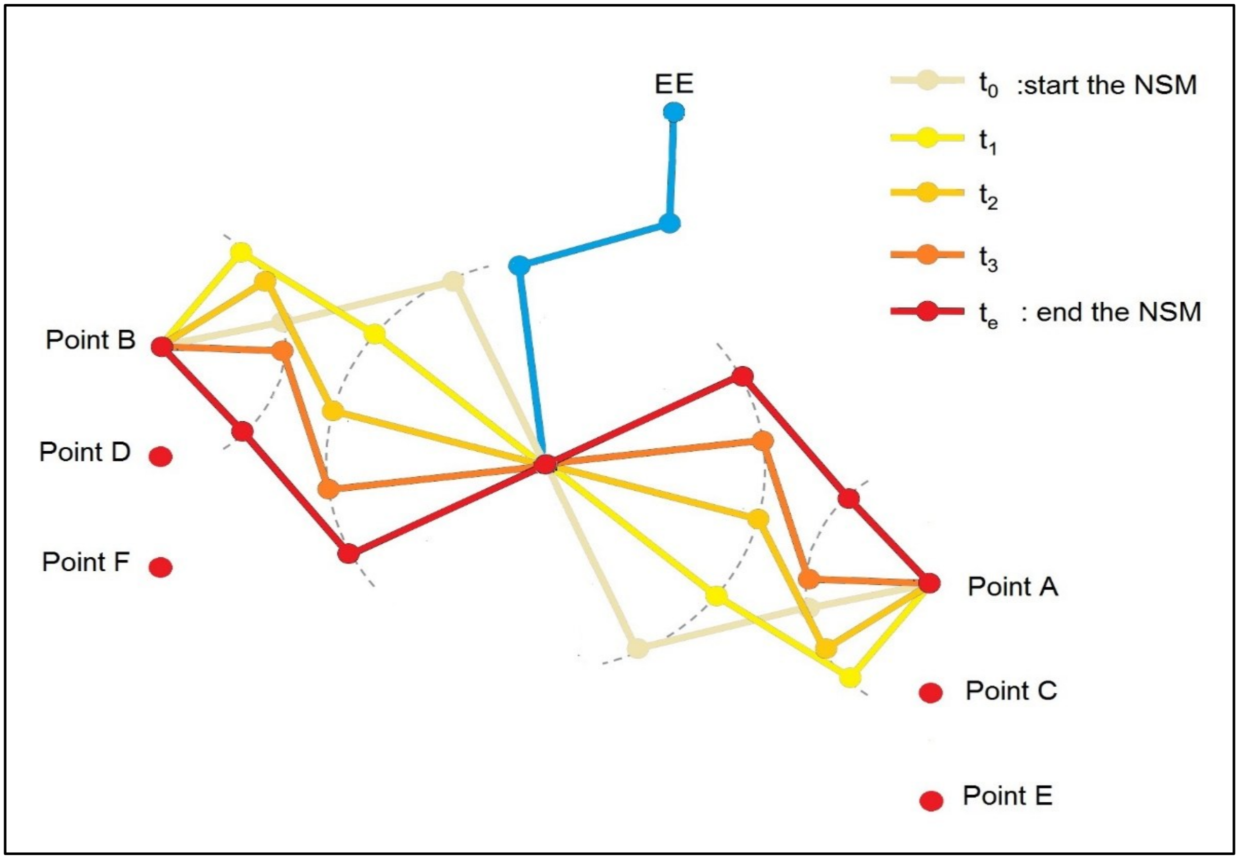


Figure 4:1: Virtual scenario to perform set of pick and place tasks.

4.2 Approach of decreasing the cycle time

Each cycle for SAMARA prototype contains four phases of motion. Two of these phases of motion are the UAMs and the others are the NSMs. As a result, the cycle time is the summation of the execution times for each of these phases as calculated in Eq. (4:8). Therefore, reducing the cycle time is achieved by reducing the time for the NSM and\or reducing the final time, also called the final time (t_f) for one or both of the UAM\s. However, reducing the time for the NSM affects the ability of lifting the payloads in the case of carrying a heavy payload, because the suction cup cannot perform this task in a short period. Consequently, this section concentrates on reducing the t_f for the UAM.

$$\text{Cycle time} = t_{f\_UAM1} + t_{f\_UAM2} + 2 * t_{NSM} \tag{4:8}$$

Where t_{f_UAM1} is the final time for the UAM$_1$ when the robot does not carry the payload. While t_{f_UAM2} is the final time for the UAM$_2$ when the robot carries the payload, and t_{NSM} is the time for performing each NSM. For simplification purposes and for similarity, t_f has been used instead of using t_{f_UAM1} or t_{f_UAM2}.

Conventionally, decreases in the final time for the robot have been achieved by using one of two approaches. The first approach focuses on increasing the handling rate for the robot by minimizing the final time. The second approach focuses on reducing the final time frequently until it reaches the target value.

Admittedly, minimizing the final time is defined as maximizing the effort of the robot during the motion to execute the tasks [Kir-04, p. 247]. The goal of minimizing the final time for the robot is to move the end effector from one configuration to another configuration as quickly as possible [Jaz-07, p. 607]. Usually, the performance index in Eq. (2:15) is used to find the time-optimal trajectory, while the expected result is equal to saturate at least one actuator to minimize the time and this principle is known as the bang-bang principle, or bang-bang control [Jaz-07, p. 633]. However, minimizing the final time for a robot to move from one state to another state occasionally is not possible [Kir-04, p. 242]. Unfortunately, there is no theory to check the existence of a time-optimal solution, to move from one state to another state [Kir-04, p. 244]. However, this approach is discarded due to minimizing the final time defined as maximizing the control effort during the motion and this behaviour is in contradiction with one of the major goals (minimizing the energy consumption). In addition, it suffers from singularity problems in some cases [Kir-04, p.241 ff.].

The second approach introduced by Li et al. calculates the time-optimal trajectory for a slewing motion for the spacecraft, that is, concentrate on using the energy consumption as a cost function as shown in Eq. (2:16). Then, add to the optimization problem the constraints related to the maximum\minimum allowable torque for each motor. Subsequently, solve the optimization problem by the first guess of the t_f, and reduce the value of t_f repeatedly to get a response similar to bang-bang principle [LI-90].

The last approach has been chosen after performing some modifications to decrease the t_f for one or both UAM/s. The modifications concentrate on the following points: Eq. (2:18) is used. Next, the algorithm uses the initial guess for the t_f to solve the optimization problem. In case the initial guess for t_f is still large or it needs more enhancement to increase the handling rate, it is reduced manually and then the problem is solved again until it reaches the desired value for t_f if possible; this value shall be suitable for the application field; see Figure 4:2.

4.3 Quasi-Linearization

The dynamic equations of the 2nd prototype of SAMARA robot are nonlinear equations. Thus, the motion of the robot with minimizing the energy consumption by solving a constraint optimization problem between two points as the case of the 2nd prototype of SAMARA is known as a nonlinear TPBVP.

QL algorithm is used as trajectory planning for the UAM and it is used to minimize the energy consumption during this phase of motion. Furthermore, the suggested approach has the capability to decrease the t_f as shown in Figure 4:2. The benefit of using QL rather than EA is to reduce the computation time, and to get results that are more accurate as shown in chapter 5.

The QL algorithm is a numerical solution to solve the nonlinear TPBVP. It solves the nonlinear TPBVP by series of linear TPBVP, which is easier to solve from a mathematical point of view [Kir-04, p. 357]. This algorithm is an iterative algorithm and it needs an initial guess to solve the optimization problem [Kir-04, p. 366]. Because of this feature, there is a possibility for divergence if the initial guess is so poor as stated in [Kir-04, p. 369 f.]. However, taking into account the physical equations, the user occasionally can provide a suitable initial guess in the majority of cases.

Preliminary configurations in the code of QL have been executed to start the simulation. Firstly, choose the correct state space representation. Secondly, provide the code by an initial guess for the t_f. Thirdly, provide the code by the values of the mass of the payload and its inertia if there is an option. Fourthly, enter the boundary conditions from the start/end NSM. Next, QL algorithm starts working and it does the following procedures stated from [Kir-04, p. 365 f.], as a general framework for QL and from [Alb-15] as an implementation of QL when using it as a trajectory planning for the UAM. However, all the equations in this section have been cited from [Alb-15].

The first step in the QL algorithm to solve the nonlinear TPBVP is to find the reduced model and this is achieved by substituting u^* calculated in Eq. (4:7) in equations (4:1), (4:2) and (4:3). Next, the linearized reduced model for the states and co-states is calculated by using the Taylor series as shown in Eq. (4:9). However, the linearized model is nonhomogeneous differential equations and it is time-varying [Kir-04, p. 361]. The linearized model is described in the nonlinear equations for the robot in an adequate way, as stated in [Alb-15]. Then, an initial guess for the nominal trajectories for the states $x^{(0)}(t)$ and the co-states $\lambda^{(0)}(t)$[5] is provided. Later, these nominal trajectories are changed in each iteration based on the QL calculations until the boundary conditions, optimality conditions, and the stop criteria are satisfied. After that the following terms $A, e_1(t),$ and $e_2(t)$ from Eq. (4:9) have been evaluated in each sample by using the trajectories $x^{(i)}(t)$, and $\lambda^{(i)}(t)$.

$$\begin{bmatrix} \dot{x}^{i+1}(t) \\ \dot{\lambda}^{i+1}(t) \end{bmatrix} = A(t) * \begin{bmatrix} x^{i+1}(t) \\ \lambda^{i+1}(t) \end{bmatrix} + \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \tag{4:9}$$

Where:-

$$A(t) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$a_{11} = \left[\frac{\partial f}{\partial x}\right]_{yo} = \frac{\partial f_y}{\partial x_o} \quad , \; a_{12} = \left[\frac{\partial f}{\partial \lambda}\right]_{yo} = \frac{\partial f_y}{\partial \lambda_o}$$

$$a_{21} = \left[\frac{\partial^2 \mathcal{H}}{\partial x^2}\right]_{yo} = \frac{\partial^2 \mathcal{H}}{\partial x_y \partial x_o} \; , \; a_{22} = \left[\frac{\partial^2 \mathcal{H}}{\partial x \partial \lambda}\right]_{yo} = \frac{\partial^2 \mathcal{H}}{\partial x_y \partial \lambda_o}$$

$$e_1(t) = f(x(t), \lambda(t), t) - a_{11}(t)x(t) - a_{12}(t)\lambda(t)$$

$$e_2(t) = -\frac{\partial \mathcal{H}}{\partial x} - a_{21}(t)x(t) - a_{22}(t)\lambda(t)$$

Now, to clarify the rest of code, it is necessary to discuss the Eq. (4:9). This equation contains two essential solutions, which are, the homogeneous solutions and the particular solution. In fact Eq. (4:9) contains ($2 *$ r) equations. Therefore, an essential step to solve Eq. (4:9) is to separate the homogeneous solution alone as shown in Eq. (4:10). Eq. (4:10) is the linearized homogeneous model for the problem and it includes also ($2 * r$) equations. QL algorithm solves Eq. (4:10) by using the forward integration for the entire period $[t_0, t_f]$ based on the r initial conditions shown in (4:11). Consequently, r solutions for Eq. (4:9) have been computed.

[5] The value of the first iteration for the index of the iterations (i) is zero

$$\begin{bmatrix}\dot{x}^{i+1}(t)\\ \dot{\lambda}^{i+1}(t)\end{bmatrix} = A * \begin{bmatrix}x^{i+1}(t)\\ \lambda^{i+1}(t)\end{bmatrix} \tag{4:10}$$

$$\left.\begin{array}{ll} x^{H1}(t_0) = [0\;0\;0\;0\;0\;0]^T, & \lambda^{H1}(t_0) = [1\;0\;0\;0\;0\;0]^T \\ x^{H2}(t_0) = [0\;0\;0\;0\;0\;0]^T\;, & \lambda^{H2}(t_0) = [0\;1\;0\;0\;0\;0]^T \\ \qquad\qquad . & \qquad\qquad . \\ \qquad\qquad . & \qquad\qquad . \\ x^{Hr}(t_0) = [0\;0\;0\;0\;0\;0]^T\;, & \lambda^{Hr}(t_0) = [0\;0\;0\;0\;0\;1]^T \end{array}\right\} r \text{ initial conditions} \tag{4:11}$$

Subsequently, the particular solution in Eq. (4:9) has been computed by using the forward integration for the entire period $[t_0, t_f]$ and by using the initial conditions in (4:12).

$$x^p(t_0) = x_0 \quad , \qquad \lambda^p(t_0) = [0\;0\;0\;0\;0\;0]^T \tag{4:12}$$

After calculating (r) homogenous solutions and the particular solution. Therefore, the final solution has been calculated by using the supervision principle and the result is shown in Eq. (4:13).

$$x^{i+1}(t) = [x^{H1}(t)\;\; x^{H2}(t) \ldots x^{Hr}(t)]c + x^p(t) \tag{4:13}$$

Eq. (4:13) is solved by storing the values of both solutions at the final time t_f as shown in Eq. (4:14).

$$x^{i+1}(t_f) = \left[x^{H1}(t_f)\;\; x^{H2}(t_f) \ldots x^{Hr}(t_f)\right]c + x^p(t_f) \tag{4:14}$$

As a result, the unknown vector c has been calculated by using Eq. (4:15).

$$c = \left[x^{H1}(t_f) \ldots x^{Hr}(t_f)\right]^{-1} * (x^{i+1}(t_f) - x^p(t_f)) \tag{4:15}$$

Where c ∈ R^{r} is the unknown vector. Hence, one iteration is completed by calculating the c vector, if the algorithm succeeds in finding the optimal trajectories for the UAM, i.e., the solution converges to a certain optimal value. Otherwise, the algorithm starts a new iteration; however, this time, the calculated trajectories are used rather than the nominal trajectory. Finally, the algorithm stops if and only if the stop criteria is satisfied, or in the case the code reaches the maximum iteration number. See the stop criterion in Eq. (4:16).

$$\left\|\begin{bmatrix}x^{i+1}(t)\\ \lambda^{i+1}(t)\end{bmatrix} - \begin{bmatrix}x^{i}(t)\\ \lambda^{i}(t)\end{bmatrix}\right\| \leq \varphi \tag{4:16}$$

In case the final time t_f is still large and the user decides to decrease it more, this approach starts from the beginning but after using an updated value for t_f, i.e., the reduced value, uses it in the preliminary configurations; see Figure 4:2.

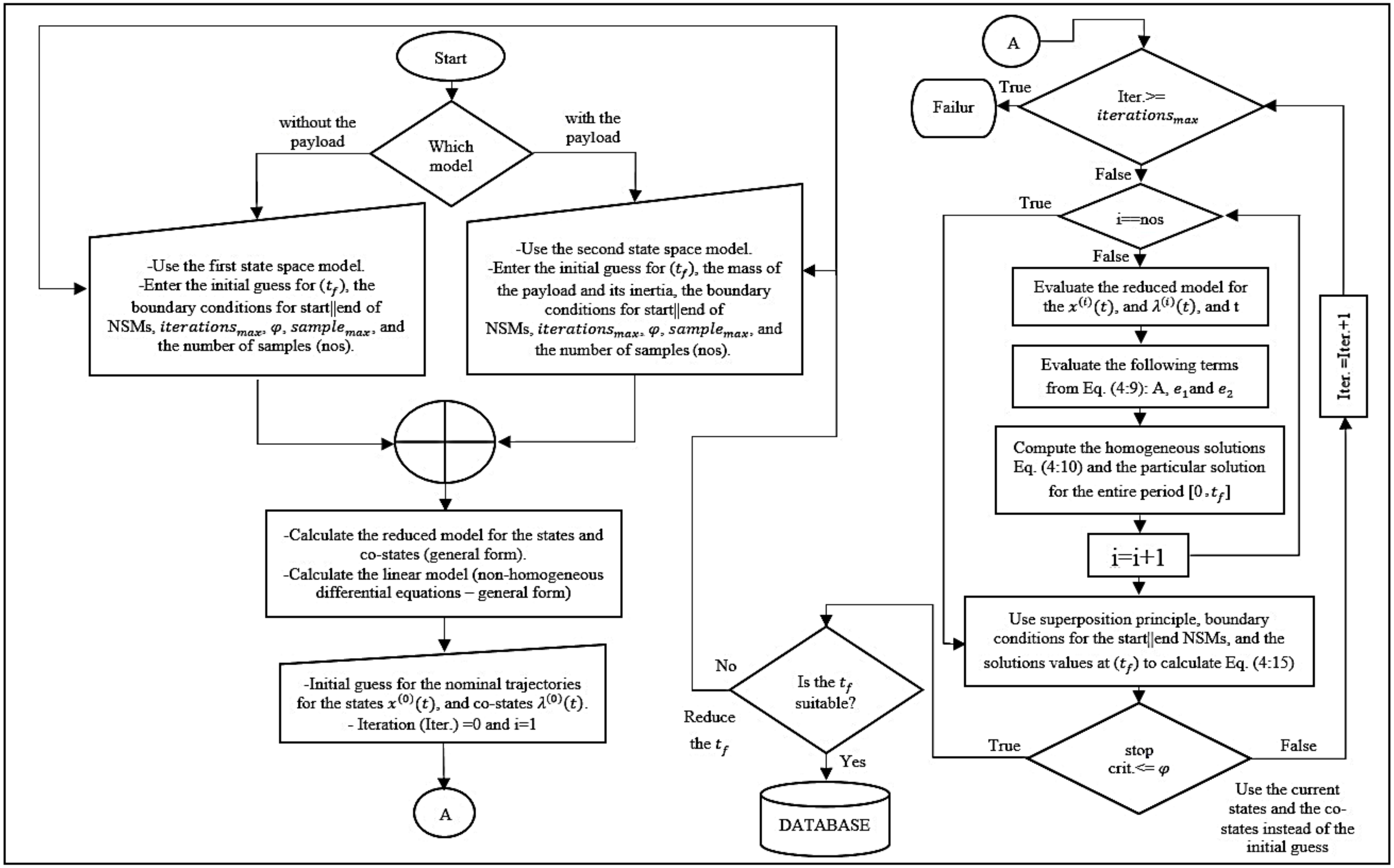
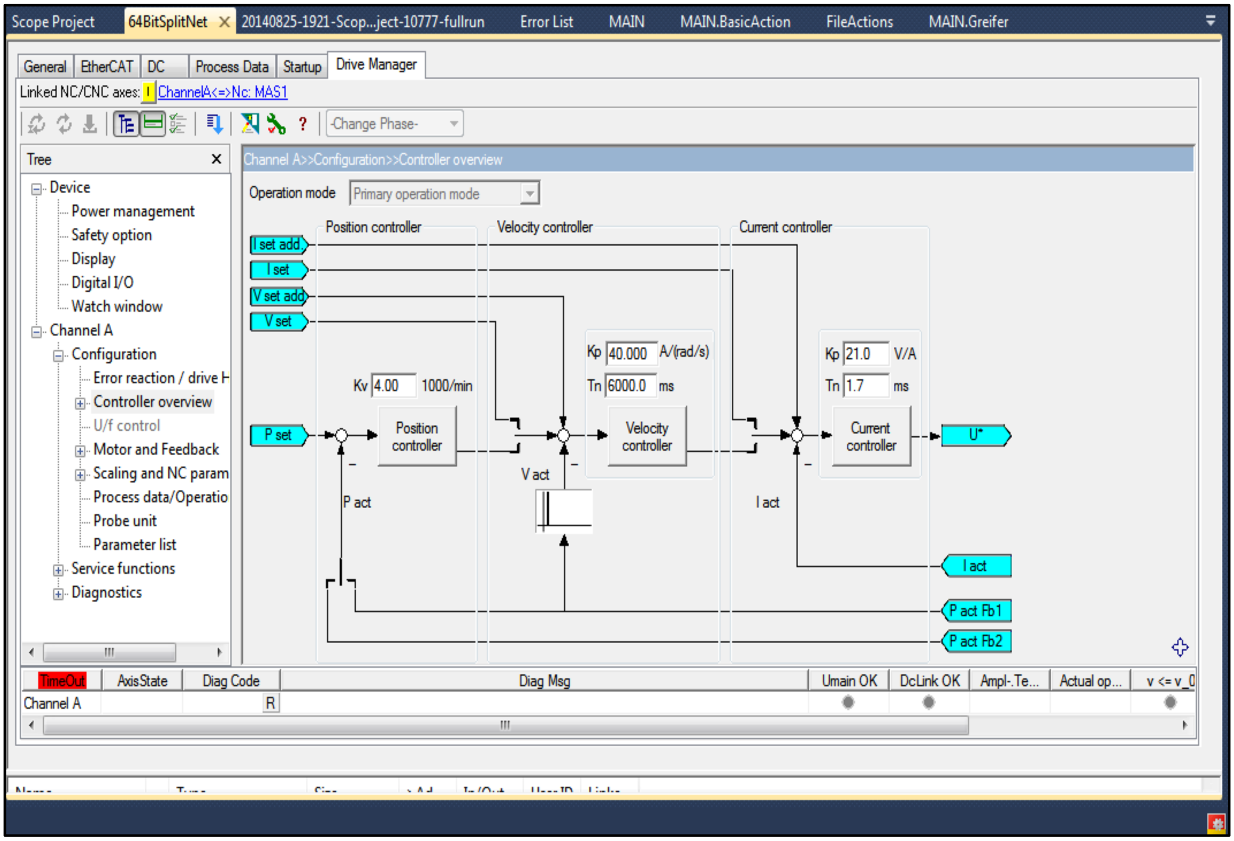

Start
Which model
without the payload
with the payload
-Use the first state space model.
-Enter the initial guess for (t_f), the boundary conditions for start||end of NSMs, $iterations_{max}$, φ, $sample_{max}$, and the number of samples (nos).
-Use the second state space model.
-Enter the initial guess for (t_f), the mass of the payload and its inertia, the boundary conditions for start||end of NSMs, $iterations_{max}$, φ, $sample_{max}$, and the number of samples (nos).
-Calculate the reduced model for the states and co-states (general form).
-Calculate the linear model (non-homogeneous differential equations – general form)
-Initial guess for the nominal trajectories for the states $x^{(0)}(t)$, and co-states $\lambda^{(0)}(t)$.
- Iteration (Iter.) =0 and i=1
A
No
Reduce the t_f
Is the t_f suitable?
Yes
DATABASE
A
Failur
True
Iter.>= $iterations_{max}$
False
True
i==nos
False
Evaluate the reduced model for the $x^{(i)}(t)$, and $\lambda^{(i)}(t)$, and t
Evaluate the following terms from Eq. (4:9): A, $e_1$and e_2
Compute the homogeneous solutions Eq. (4:10) and the particular solution for the entire period $[0, t_f]$
i=i+1
Iter. =Iter.+1
Use superposition principle, boundary conditions for the start||end NSMs, and the solutions values at (t_f) to calculate Eq. (4:15)
True
stop crit.<= φ
False
Use the current states and the co-states instead of the initial guess

Figure 4:2: Flowchart for the trajectory planning for the UAM.

4.4 Position control scheme

In the last three sections, a trajectory planning algorithm has been introduced for the UAM, while the trajectory planning for the NSM is clarified in section 2.4.2. The trajectories of NSM and the trajectories of UAM for each axis are integrated into the final trajectory by using the interpolation. The final trajectory for each axis is similar to the videotape as shown in Figure 2:5. In each portion of the trajectory, a specific motion is performed, e.g., ST, NSM, UAM, or EM for the MAS. Therefore, the results are recorded in a txt file; this file is defined by the number of samples, e.g., TwinCat_TRAJ_new8651.txt. The txt file contains the execution time, the angular position for each axis, the angular velocity for each axis, and the angular acceleration for each axis. The position trajectory for each axis is used as reference command for the controller to control the motion of the robot. The control diagram shows a top-level PID position control with the possibility to control the velocity and the current as shown in Figure 4:3.



Linked NC/CNC axes: I ChannelA<=>Nc: MAS1

Operation mode: Primary operation mode

Kv 4.00 1000/min

Kp 40.000 A/(rad/s)

Tn 6000.0 ms

Kp 21.0 V/A

Tn 1.7 ms

TimeOut	AxisState	Diag Code	Diag Msg	Umain OK	DcLink OK	Ampl-.Te...	Actual op...	v <= v_0
Channel A		R		●	●			●

Figure 4:3: Controller structure in the TwinCAT software.

This controller is an industrial controller provided by the Beckhoff New Automation Technology Company. The reason for using this controller is related to this type of controller being familiar in the industrial fields and it is used in several applications as stated in [Bec-16a]. In addition, the Beckhoff Company provides TwinCAT software; see Figure 4:3, which has the capability to accesses more than one PLC system; it also has the capability to access the NC axis control, and the controller works in a real-time framework [Bec-16a].

4.5 Optimal control scheme

The optimal control is focused on minimizing the cost function to get an optimal response for the system. Optimal control as a family contains several controllers, e.g., MPC, NMPC, AMPC, LQR, etc. Each one of these methods has an optimization problem and the target is to minimize the cost function to achieve an optimal performance. Usually, the cost function in the optimization problem is chosen based on the application and its requirements as shown in Table 2:3. This dissertation focuses on developing an AMPC controller as a second choice controller for the 2nd of SAMARA robot, because it has the features that can satisfy the features of the efficient controller.

The AMPC has been chosen as an optimal controller to control the MAS only. Furthermore, Albalasie et al. developed this controller to control the UAM in the 2nd in SAMARA robot [Alb-16]. However, because SAMARA has two main phases of motion with different characteristics, the control scheme is modified to suit both of them as shown in subsection 4.5.2.

The AMPC tracks the position trajectories for the MAS only as a reference signal in the final *.txt file e.g. TwinCat_TRAJ_new8651.txt. However, it has the capability to track other trajectories, which are computed by other trajectory planning algorithm, e.g., the PTP synchronized trajectory planning algorithm.

The control scheme contains two main processes namely, the AMPC, and the successive linearization process as shown in Figure 4:5. The idea is focusing on using the AMPC as a controller but it can solve the LTI models only, which is not the case of the SAMARA robot, because the models for SAMARA are nonlinear models as shown in section 2.4.1. Consequently, successive linearization process is used to solve the previous problem as shown in the following subsection.

4.5.1 Successive linearization

The majority of mechanical systems in our life have nonlinear models. Consequently, there are two famous methodologies to describe the system. The first methodology is focused on using the linearized model at the nominal operating point, which describes the nonlinearity well if it is operated in the neighbourhood of the operating point. While in many cases, the response is not acceptable at all because the system is operated far away from the operating point. The second methodology is focused on using the nonlinear model as it is. Nevertheless, using the nonlinear model in the optimization problem requires long computation time, because the optimization problem becomes a non-convex problem, which is also difficult to solve [Cam-07, p. 250]. Therefore, it is not suitable for the real-time framework in case the system has a fast dynamic. In addition, identifying dynamic parameters for the nonlinear model is harder than the same process for the linear model [Cam-07, p. 249 f.].

During this research, two nonlinear state space representations for the 2nd prototype of SAMARA robot are developed in section 2.4.1. Therefore, the goal of using the successive linearization process is to provide a collection of LTI models for the controller to use them instead of using the nonlinear model or one linearized model at specific nominal operating point [Mat-14]. Consequently, AMPC uses the accurate linearized model at each control interval.

$$\begin{aligned}\bar{x}_{k+1} &= A_k\bar{x}_k + B_k\bar{u}_k \\ \bar{y}_{k+1} &= C_k\bar{x}_k + D_k\bar{u}_k\end{aligned} \tag{4:17}$$

Where

$$A_k = \begin{bmatrix}\frac{\partial f(x,u)}{\partial x_1} & \frac{\partial f(x,u)}{\partial x_2} & \cdots & \frac{\partial f(x,u)}{\partial x_r}\end{bmatrix}_{r*r}$$

$$B_k = \begin{bmatrix}\frac{\partial f(x,u)}{\partial u_1} & \frac{\partial f(x,u)}{\partial u_2} & \cdots & \frac{\partial f(x,u)}{\partial u_m}\end{bmatrix}_{r*z}$$

Where z and o are the numbers of the inputs and outputs for the LTI model respectively. In addition, $\bar{x}_k$ represents the error with respect to the reference points, while $\bar{u}_k$ is associated with the control input. Moreover, $A_k \in R^{r*r}$ is the dynamic matrix while $B_k \in R^{r*z}$ is the input matrix. Furthermore, $C_k \in R^{o*r}$ is the output matrix and $D_k \in R^{o*z}$ is the feedback matrix which is a zero matrix. The successive linearization code linearized the nonlinear model at the initial nominal operating point. Then, the code linearized the nonlinear model at the whole nominal trajectories u, and x. Finally, a collection of LTI models have been achieved for describing the system as shown in Eq. (4:17). Then, these models are discretized to be suitable for AMPC, because AMPC is a discrete controller [Alb-16]. Finally, $A_k, B_k, C_k,$ and D_k are matrices varying with time [Mat-14].

4.5.2 Adaptive model predictive control

AMPC is a discrete controller and it solves QP optimization problems to perform optimal performance. It has the ability to take into consideration the constraints on inputs, the change rates of inputs, and on the outputs, if applicable. Furthermore, AMPC applies the three famous concepts as any MPC controller, which are discussed in subsection 2.5.3.

MPC uses a unique LTI model to control the system and to predict the future response of the system. Usually, the LTI model is calculated at the nominal operating point. However, if the controller is operated in another far operating point, the system has a failure or unacceptable performance. On the contrary, AMPC introduces the solution to the previous problem, because it uses a fixed LTI structure. This structure allows describing the system by using the current LTI model at the current operating point, while the current LTI model is used to predict the behaviour of the system during the prediction horizon [Mat-14]. A similar technique is presented as a solution to the non-convex optimization problem by using the current LTI model to predict the future response during the prediction horizon, while the nonlinear model has been used to calculate the effect of past input moves, i.e., the problem is converted to a set of QP problems. The effect of using this technique is the ability to solve the QP problem at each control horizon and it is suitable for a real time-frame [Cas-07, p. 270].

According to [Mat-14], AMPC is a suggested solution to solve problems that have strong nonlinear models or the time-varying models. On this basis, it is used to design and develop the optimal control scheme for the 2nd prototype for SAMARA robot. The flowchart for the optimal control scheme is shown in Figure 4:4, while the control scheme in the Simulink environment is shown in Figure 4:5.

$$\emptyset = \sum_{i=0}^{p-1} (\emptyset_1 + \emptyset_2 + \emptyset_3) \tag{4:18}$$

where

$$\emptyset_1 = (\mathrm{r}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - y(\mathrm{k}+\mathrm{i}|\mathrm{k}))^{\mathrm{T}} \frac{\mathrm{Q}}{S_y} (\mathrm{r}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - y(\mathrm{k}+\mathrm{i}|\mathrm{k}))$$

$$\emptyset_2 = \left(\mathrm{u}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - u_{target}(\mathrm{k}+\mathrm{i}|\mathrm{k})\right) \frac{\mathrm{R}}{S_u} \left(\mathrm{u}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - u_{target}(\mathrm{k}+\mathrm{i}|\mathrm{k})\right)$$

$$\emptyset_3 = (\mathrm{u}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - u(\mathrm{k}+\mathrm{i}-1|\mathrm{k})) \frac{\mathrm{S}}{S_u} (\mathrm{u}(\mathrm{k}+\mathrm{i}|\mathrm{k}) - u(\mathrm{k}+\mathrm{i}-1|\mathrm{k}))$$

Where $\emptyset$ is the total cost function for the quadratic programming problem, $\emptyset_1$ is the cost function for minimizing the error to track the reference vectors, $\emptyset_2$ is the cost function for minimizing the inputs or to become near the nominal target inputs (u_{target}), $\emptyset_3$ is the cost function for the change rates of the inputs, k denotes the current sampling time, and P is the prediction horizon; $\mathrm{r(k+i|k)}$ is the reference vector at the i-th prediction horizon in the current sample k, and $y(\mathrm{k+i|k})$ is the predicted vector for the response at the i-th prediction horizon in the current sample k. In addition, S_y and S_u are scaling factors.

The QP optimization problem is already programmed inside the AMPC block in MATLAB. Therefore, the cost function form for the QP is shown in Eq. (4:18) after implementing some modifications subject to the successive linearized dynamic equations as shown in Eq. (4:17). Moreover, in this research, the hard constraints are used for the optimization problem are shown in Eq. (4:19). The following AMPC equations are stated from [Mat-14] after neglecting the unused terms.

The flowchart in Figure 4:4 describes how the AMPC is working. Firstly, the trajectory planning algorithm is used to calculate the reference trajectories with/without the input torque for each axis. Then, the successive linearization process starts to calculate the LTI model for the current operation point. Next, the AMPC uses the current LTI model to predict the performance of the system during the prediction horizon. Furthermore, the same LTI model is used during the control horizon to calculate the optimal inputs torque for the actuators after satisfying the hard constraints in Eq. (4:19).

$$\begin{gathered}
\frac{y(i)_{min}}{S_y} \leq \frac{y(k+i|k)}{S_y} \leq \frac{y(i)_{max}}{S_y} \\
\frac{u(i)_{min}}{S_u} \leq \frac{u(k+i|k)}{S_u} \leq \frac{u(i)_{max}}{S_u} \\
\frac{\vartriangle u(i)_{min}}{S_u} \leq \frac{\vartriangle u(k+i|k)}{S_u} \leq \frac{\vartriangle u(i)_{max}}{S_u}
\end{gathered} \tag{4:19}$$

Where $y(i)_{max}$ and $y(i)_{min}$are the maximum and the minimum allowable outputs at the prediction i, respectively. While $u(i)_{max}$ and $u(i)_{min}$are the maximum and the minimum allowable inputs at the prediction i respectively. Finally, $\vartriangle u(i)_{max}$and $\vartriangle u(i)_{min}$ are the maximum and the minimum allowable change of rates in the inputs at the prediction i, respectively.

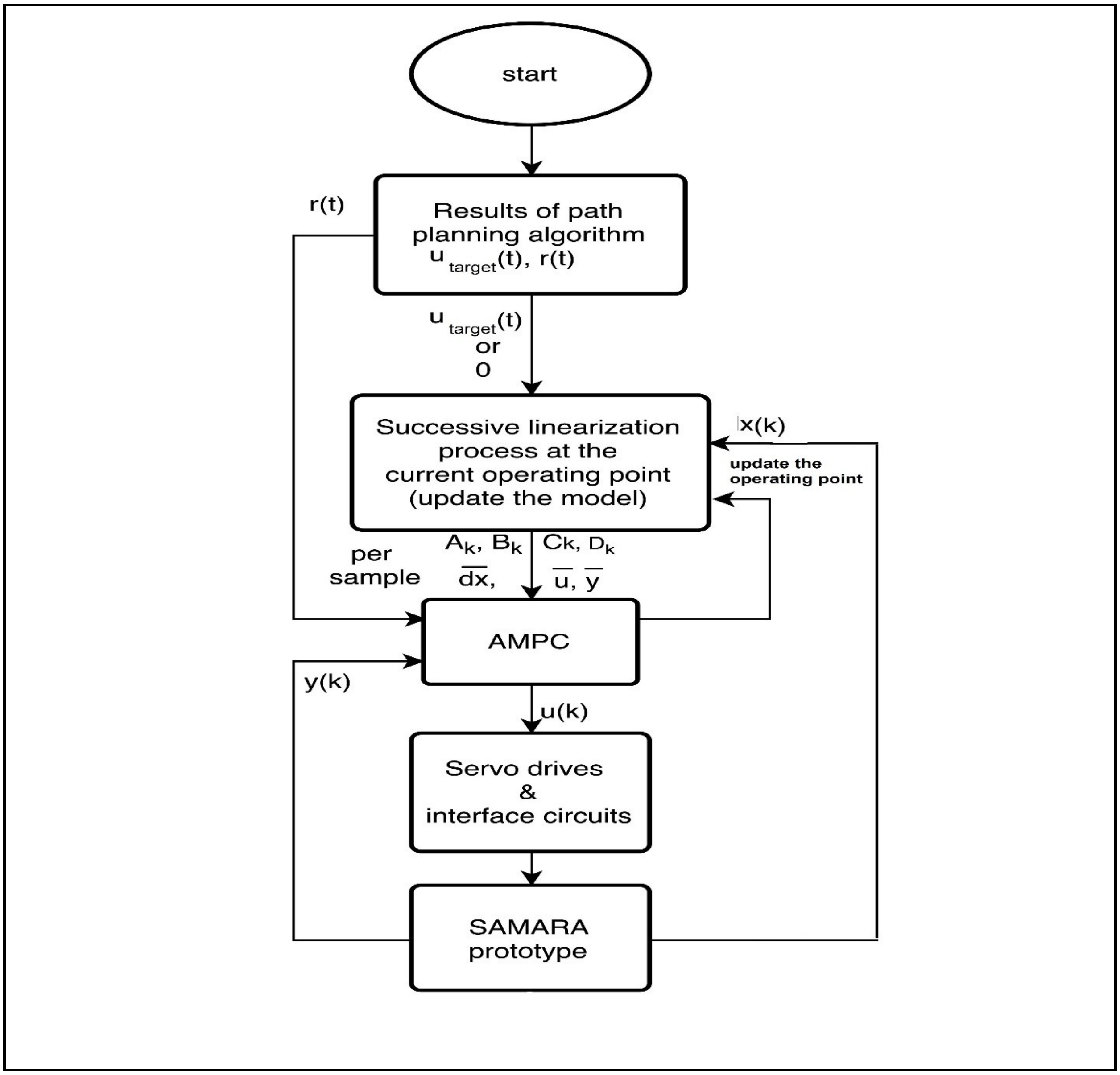

start
Results of path planning algorithm u target(t), r(t)
r(t)
u target(t) or 0
Successive linearization process at the current operating point (update the model)
x(k)
update the operating point
per sample
Ak, Bk dx,
Ck, Dk u, y
AMPC
y(k)
u(k)
Servo drives & interface circuits
SAMARA prototype

Figure 4:4: Control flowchart for the optimal control scheme.

Generally, there are two possibilities for providing an input for the successive linearization block, which provide $\mathrm{u}_{\mathrm{target}}$ trajectory or a null trajectory as shown in Figure 4:4. The difference between the two cases and the effect is also shown in Eq. (4:18). In the case of using u_{target}, the AMPC provide inputs to the system as the same values of u_{target} or at least the values are so close to the values of u_{target}, i.e., the controller tries to track the u_{target} [Mat-14]. While in the second case, the controller can select any input values to minimize the cost function $\emptyset_2$ in Eq. (4:18). However, in this research, the first choice is chosen because the provided commands are an optimal inputs.

Essential concepts must be clarified, because they affect the AMPC design, which are the feasibility/infeasibility, the stability, and the computation time. Each one of these concepts is a critical point for the performance of the controller and on the ability of implementing the controller in a real environment.

According to [Mat-16b], the infeasibility problem means the MPC controller cannot find the optimal solution at the control horizon for the optimization problem, because it cannot satisfy the hard constraints in Eq. (4:19) for the system. As a result, there is a possibility for a failure in control if this problem is not resolved. According to [Mat-16b], the infeasibility problem usually occurs when there are hard constraints on the maximum limits of inputs and hard constraints on the change rates of the inputs; hence, the controller cannot satisfy all the constraints at the same time, while using one set of these hard constraints will not cause infeasibility. The suggest solution for the previous problem is to relax the constraints, i.e., soften the constraints or redesign the system itself to achieve the required goals if it is necessary.

The second essential concept is the stability of the system and this is achieved by tuning the design parameters Q, R and S if the system is reachable. The effect of selecting larger values of matrix Q more emphasis is placed by optimal control on returning the response to the reference command. Furthermore, this increases the energy consumption, because it increases the values of the inputs, i.e., selecting larger values of matrix Q forcing the eigenvalues to move more to the left side on the S-plane. In contrast, the opposite effects occur if the selected values for matrix R are so large. Another parameter that affects the stability is the prediction horizon. According to [Cam-07, p. 271], using a large value for the control and prediction horizons are an essential element to guarantee the stability of the system in several cases. However, it has prejudicial effects on the computation time because it increases the complexity of the optimization problem.

In summary, a trade-off approach is used to select suitable values for the horizons P and C to guarantee the stability and to reduce the computation time at the same time, as shown in chapter 5.

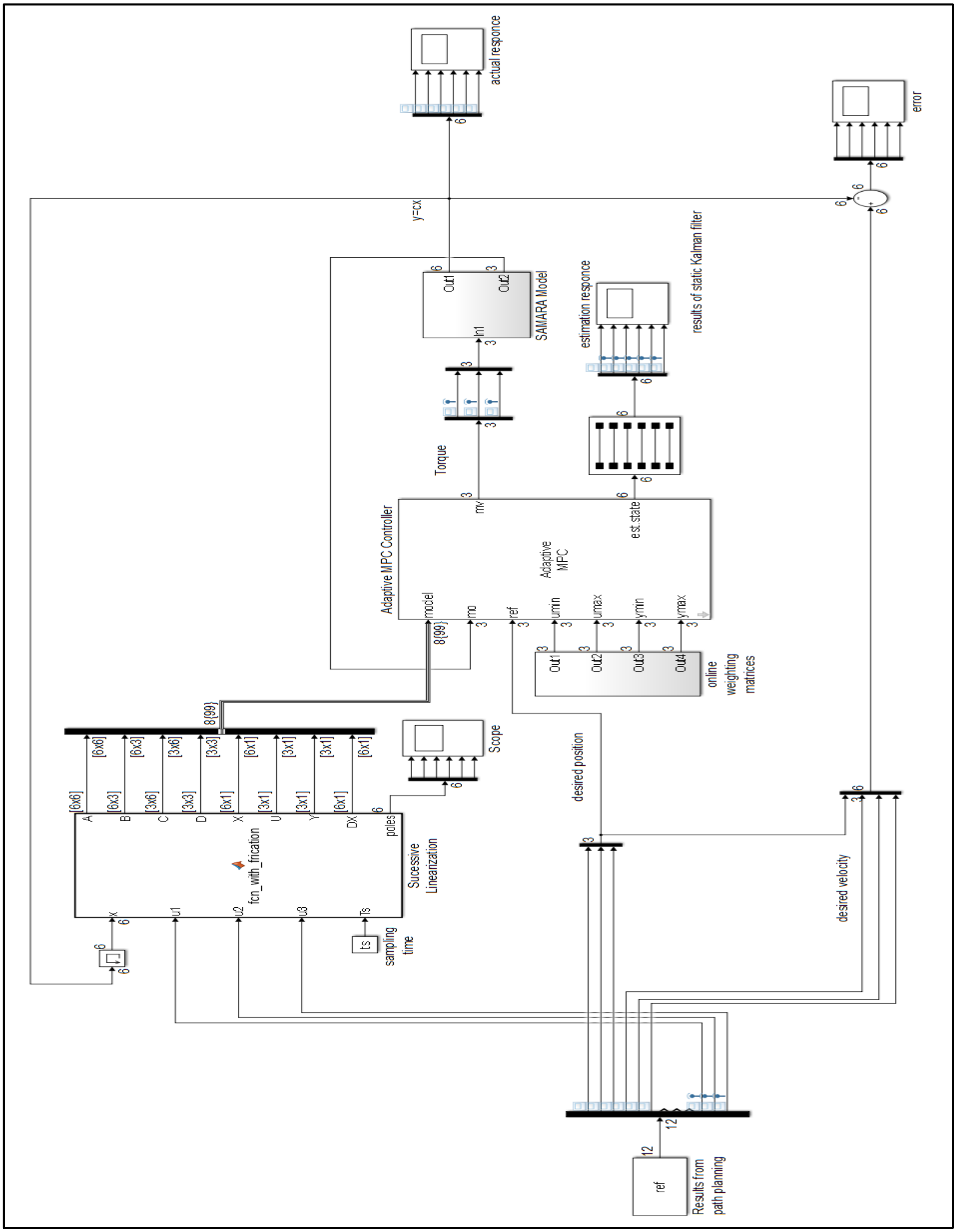


Figure 4:5: Simulink scheme for the optimal control scheme.

5 Evaluation of Control Techniques

The approach for how to achieve the goals of minimizing the energy consumption, decreasing the cycle time, reducing the computation time for solving the optimization problem, decreasing the gap between the simulation and the experimental results, and developing an efficient control scheme is verified in four group of experiments and six separated experiments. Some of these experiments were tested in the lab by using prototype, while the other experiments were tested within a simulation environment due to the lack of available equipment as shown in this chapter.

The first group of experiments (GE1) and the first experiment (E1) verified the trajectory planning algorithm based on comparing the power consumption for the MAS in the simulation environment and the experimental frameworks. Moreover, the criterion of measuring the computation time for the QL algorithm in Matlab environment is evaluated and the applicability of the position control scheme, which has been evaluated as well.

The second experiment (E2) checked the ability of the robot to carry a heavy payload in combination within 2.1 s as a cycle time. This experiment verified the applicability of using this type of robot to perform the pick and place tasks in the end line of the press lines as addressed in industrial applications in section 6.2. While the third experiment (E3) tested the ability of the robot to execute pick and place tasks for a light payload when the NSM time 30 ms. The E3 clarified the applicability to use the 2nd prototype of SAMARA robot in principle in the food industry as addressed in industrial applications in section 6.1.

The fourth (E4) and the fifth (E5) experiments evaluated the optimal control scheme based on simulations. E4 focused on evaluating the improvement on the power saving by using the optimal control scheme. E5 focused on evaluating the computation time for the AMPC into Matlab environment. In addition, the E5 suggested technical solutions to solve the problem of computation time. Furthermore, the sixth experiment (E6) evaluated the robust optimal control scheme after providing the optimal control scheme by the required information about a measured disturbance. The objective of the E6 is to test the ability of the robust optimal control to reject a measured disturbance.

The second group of experiments (GE2) focused on analyzing the power consumption for the 2nd prototype of SAMARA robot if the PTP synchronized trajectory planning algorithm is used to connect between the two NSMs instead of using the QL algorithm. These group of experiments clarified the benefit of using QL algorithm for the UAM phase to increase the power saving.

The third group of experiments (GE3) compared the electrical power consumption and the cycle time for the 1st prototype of SAMARA, the 2nd prototype of SAMARA, and the Adept one MV SCARA robot, abbreviated as (SC1). Finally, the fourth group of experiments (GE4) focused on comparing the mechanical power consumption for: the 2nd prototype of SAMARA, the simulated model for the Bosch Turboscara SR 60 SCARA robot, abbreviated as (SC2) when the PTP synchronized trajectory planning algorithm is used, and the simulated model for the SC2 when the QL algorithm is used as trajectory planning algorithm. All these experiments are summarized in Table 5:1.

A comparison between the 2nd prototype of SAMARA with other industrial robots has been done to evaluate the performance of the robot from different perspectives e.g. the type of the robot, the maximum reachable distance, the cycle time, the enhancement rate for cycle time of 2nd SAMARA prototype, the DOFs, the maximum payload, the mass of the robot, and the price etc. This comparison is discussed in section 5.4.

Now, lest us focus on the experimental setups. Firstly, three encoders are used to measure the angular positions of the axes in the MAS. These encoders are identical encoders and the manufacturer is Sick Company. The model used is the EKS36-0KF0A020A, see Figure 5:1. These encoder have resolution of 20 bits per revolution [Sic-16]. On the other hand, the torque for each axis in the MAS is estimated by the servo drives. The idea is each servo drive has the capability to measure the output current and by using this information the servo drive can estimate the output torque. In the case of the first axis in the MAS, the servo drive AX5106-000-0201 from Beckhoff Company is used to estimate the torque for axis one. While for the second and third axes another model is chosen, but the servo drives for the second and third axes are still identical. The model AX5206-0000-0201 from Beckhoff Company has been chosen to execute these tasks for the second and third axes as shown in Figure 5:1.

Measuring the power consumption for the whole robot can be achieved by using a 3-phase power measurement for alternating voltages. The model EL3403 from Beckhoff Company has been chosen but this device cannot measure currents higher than 1 A [Bec-16d]. Therefore, the model WSK 60 from MBS AG Company which is a wound current transformer is used to convert the current, with a ratio 15:1, to less than 1 A to be suitable for the EL3403 device, see Figure 5:1 [WSK-16]. On the other hand, the EL3433 from Beckhoff Company has been chosen to measure the input power of the two used AC to DC power supplies. These two provide the DC low voltage 24V and 48V for the following components: the fourth and fifth motors in the SAS, the valves, the logic, and the encoders. This device is a 3-phase power measurement for alternating voltages [Bec-16e].

It is worth to mention that the power consumed in the MAS can be calculated by subtracting the power measured by the EL3433 device from the power measured by the EL3403 device. In addition, it is important to mention this robot uses the EtherCAT field bus as a high-speed communication system to transfer the data and commands between its components.

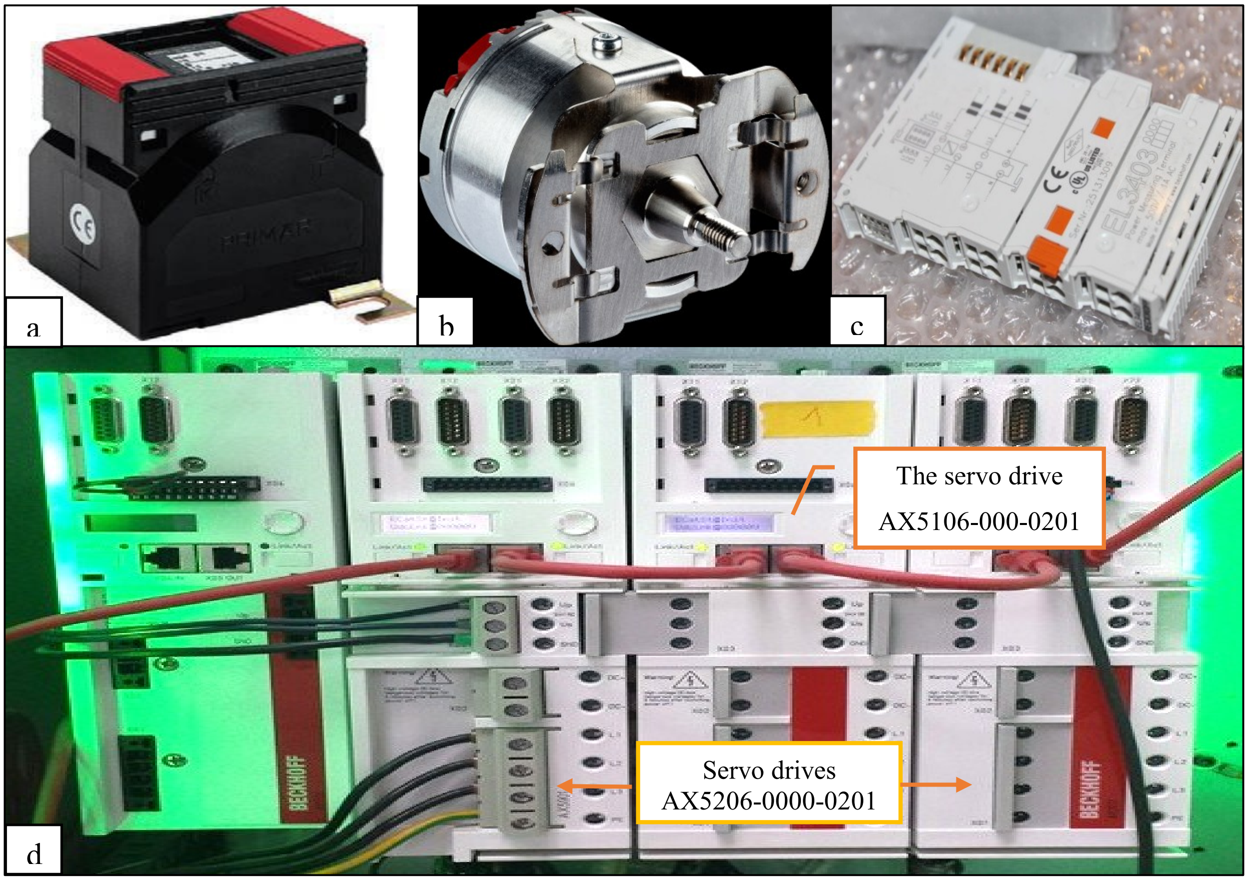


Figure 5:1: The measurement equipment: a) The WSK 60 wound current transformer [MBS-16] b) The EKS36-0KF0A020A encoder [Sic-16] c) The EL3403 three phase power measurement [Pho-16] d) The servo drives AX5106-000-0201 and AX5206-0000-0201.

Table 5:1: A general overview of the experiments which are performed.

Identification no.	Tests, evaluations, comparisons	Applied control scheme			Results by	
		Position	Optimal	Robust optimal	Simulation\s	Experiment\s
GE1	• Verifying the trajectory planning algorithm by comparing between the simulation and experimental results for the power consumptions in different conditions. • Testing the suggested approach for decreasing the cycle time. • Evaluating the computation time for the QL algorithm.	√			√	√
E1	• Evaluating the position control scheme by measuring the backlash error.	√			√	√
E2	• Testing the ability of the robot to execute pick and place tasks for a heavy payload in a short cycle time.	√			√	√
E3	• Testing the ability of the robot to execute pick and place tasks for a light payload when the NSM time was short.	√			√	√
E4	• Testing the optimal control scheme. • Evaluating the improvement of the power consumption if the optimal control scheme is used.		√		√	
E5	• Testing the optimal control scheme. • Evaluates the computation time for the AMPC controller.		√		√	
E6	• Testing the robust optimal control scheme in the existence of a measured disturbance.			√	√	
GE2	• Evaluating the power consumption for the 2nd prototype of SAMARA robot under the same conditions if: o The PTP synchronized trajectory planning algorithm is used to connect between the two NSMs. o The QL algorithm is used to connect between the two NSMs.	√	√		√	√
GE3	• Comparing the electrical power consumption and the cycle time for: o The 1st prototype of SAMARA robot. o The 2nd prototype of SAMARA robot. o The SC1.	√			√	√
GE4	• Comparing the mechanical power consumption and the cycle time for: o The 2nd prototype of SAMARA robot. o The SC2 when the PTP synchronized trajectory planning algorithm is used. o The SC2 when the QL algorithm is used.	√			√	√

5.1 Trajectory planning and the position control scheme

The trajectory planning algorithm for the MAS is evaluated with the position control scheme on the 2nd prototype of SAMARA robot from different perspectives, e.g., the power consumption, the minimum achieved cycle time, the maximum payload, and the achieved accuracy.

As mentioned before, a *.txt file is created and this file contains the absolute time, the angular position for each axis, the angular velocity for each axis, and the angular acceleration for each axis. These data are exported from Matlab to the position control scheme by using the *.txt file; then the TwinCAT reads the *.txt file to perform the required motion.

The position control scheme proposed in section 4.4 uses the data of the angular position of each axis as a reference command for each axis. Therefore, the developed algorithm for the trajectory planning with the position control scheme has been verified through several experiments by comparing the simulation and experimental results.

Table 5:2: Specification of the used payloads in all experiments.

Payload No.	Mass	Dimensions	$\mathbf{I_{zz}}$	$\mathbf{I_{yy}}$	$\mathbf{I_{xx}}$
	[kg]	length*width* thickness [mm]	$[\mathbf{kg.m^2}]$	$[\mathbf{kg.m^2}]$	$[\mathbf{kg.m^2}]$
1	0.300	120*120*3	0.00072	0.000360225	0.000360225
2	1.1	100.8*100.6*14.3	0.001859092	0.00094644458	0.00095013692
3	2.5	130*128.7*19.45	0.00697160208	0.003529581771	0.003599646354
4	3.4	15.8*148.3*19.4	0.01260631883	0.0064816353	0.0064816353
5	5.2	202*165*21	0.02947923334	0.0119886	0.0178728334

The GE1 is designed to measure the total of the average power consumption in the simulation and experimental frame works to evaluate the trajectory planning method by using the position control scheme. Therefore, a pick and place task is performed for five cycles between the same processing points (the initial point and the final point) with a specific cycle time and with a set of the payloads. After that the same procedures are repeated again, but after reducing the cycle time each attempt. The payloads used in these experiments are (see Table 5:2): the payload number 2, the payload number 3, the payload number 4, and finally the robot perform the tasks without any payload. The distance between the initial point and the final point was 1.6 m. In the GE1, the time used for the

NSM was 200 ms because the used payloads are heavy. The results of the total of the average actual and the simulated mechanical power consumptions for the MAS are shown in Figure 5:2. While the total of the average actual electrical power consumptions for all axes are shown in Figure 5:3.

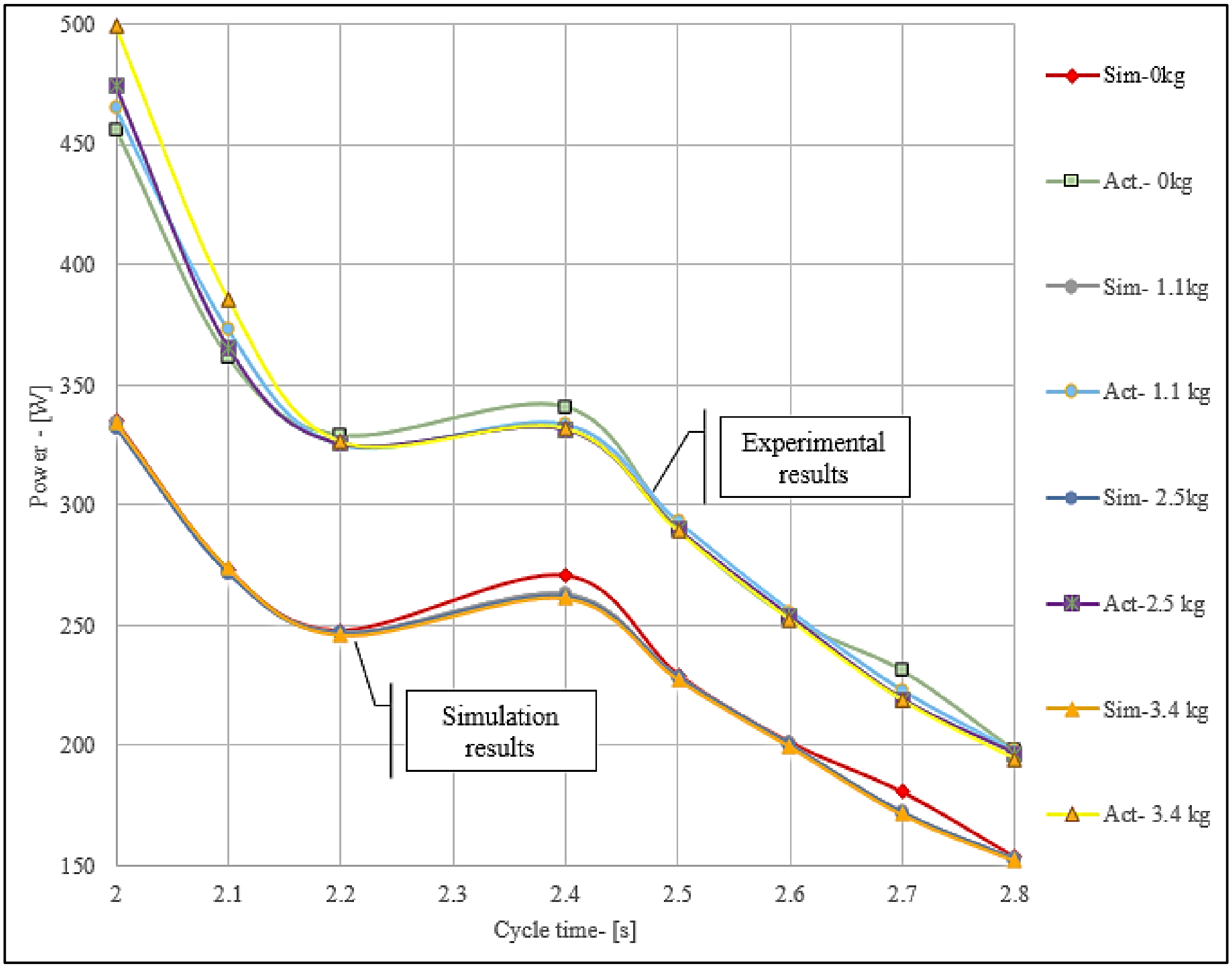


Figure 5:2: Comparison between the total of the average of the actual and the simulated mechanical power consumption for the MAS in GE1.

To clarify the GE1 in more detail, let us present one of the previous experiments in GE1 for executing the pick and place task for five cycles and this is the E1. The payload number four is used in E1 (see Table 5:2) and the distance between the processing points was 1.6 m. In addition, the cycle time was 2.4 s and the time for the NSM was 200 ms.

The path of the end effector is shown in Figure 5:4 with the coordinates of the start point, the initial point, the final point, and the end point, while the consumed power is measured for the MAS as shown in Figure 5:5. However, the trajectory planning algorithm can decrease the cycle time up to 1.6 s as shown later, if the first motor is replaced by another motor to provide sufficient torque for the SM phase. The reasons are because SAMARA has a high inertia, and a high mass, therefore, it needs higher torque limits, especially in the time for the SM phase.

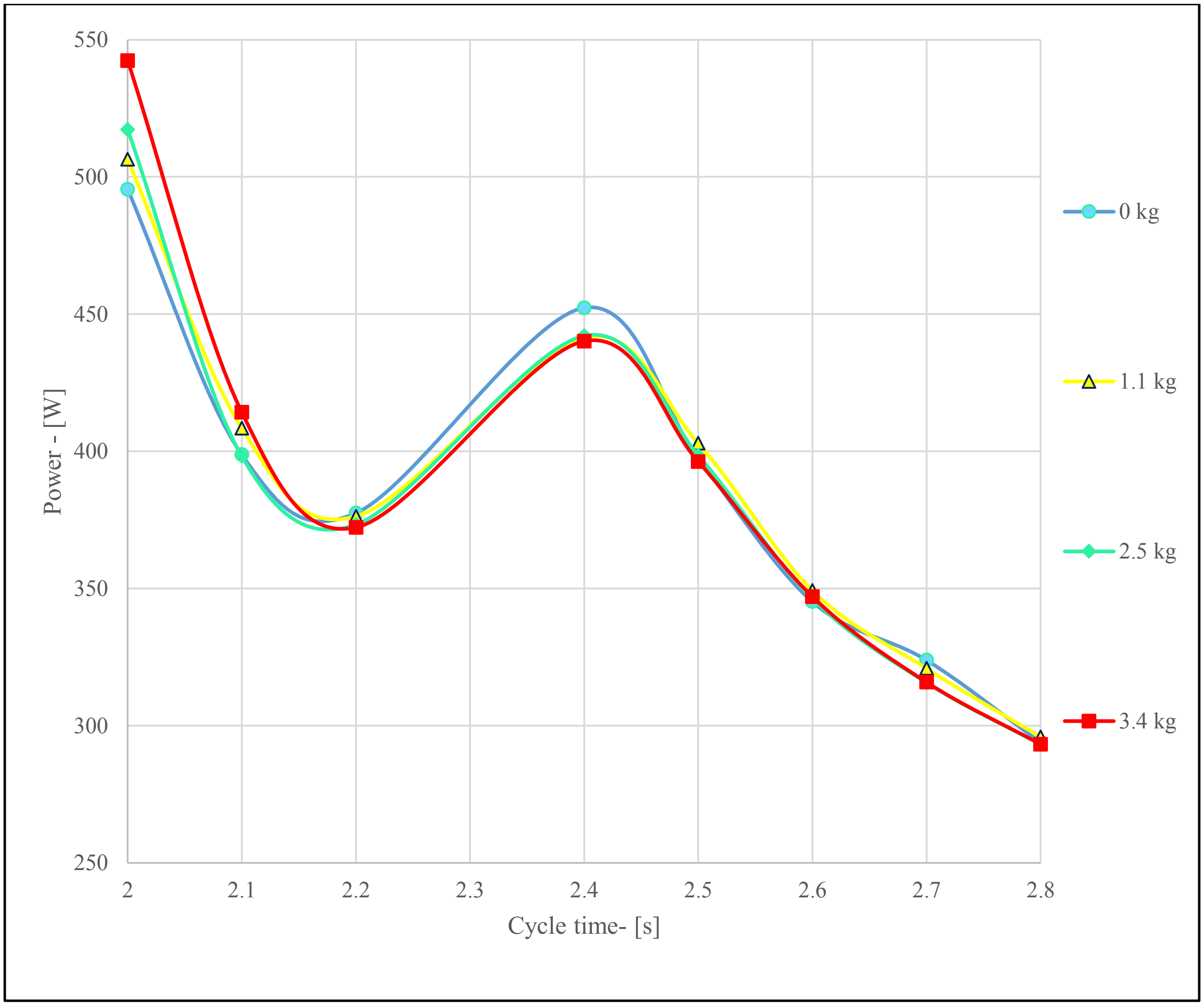


Figure 5:3: The total of the average of the actual electrical power consumption for all axes in the case of GE1[6]*.*

The start point and the end point in E1 are the same point. While the coordinate for the initial point was (-0.8,0) and the coordinate for the final point was (0.8,0) as shown in Figure 5:4.

[6] This power consumption includes the consumed power in the standby situation.

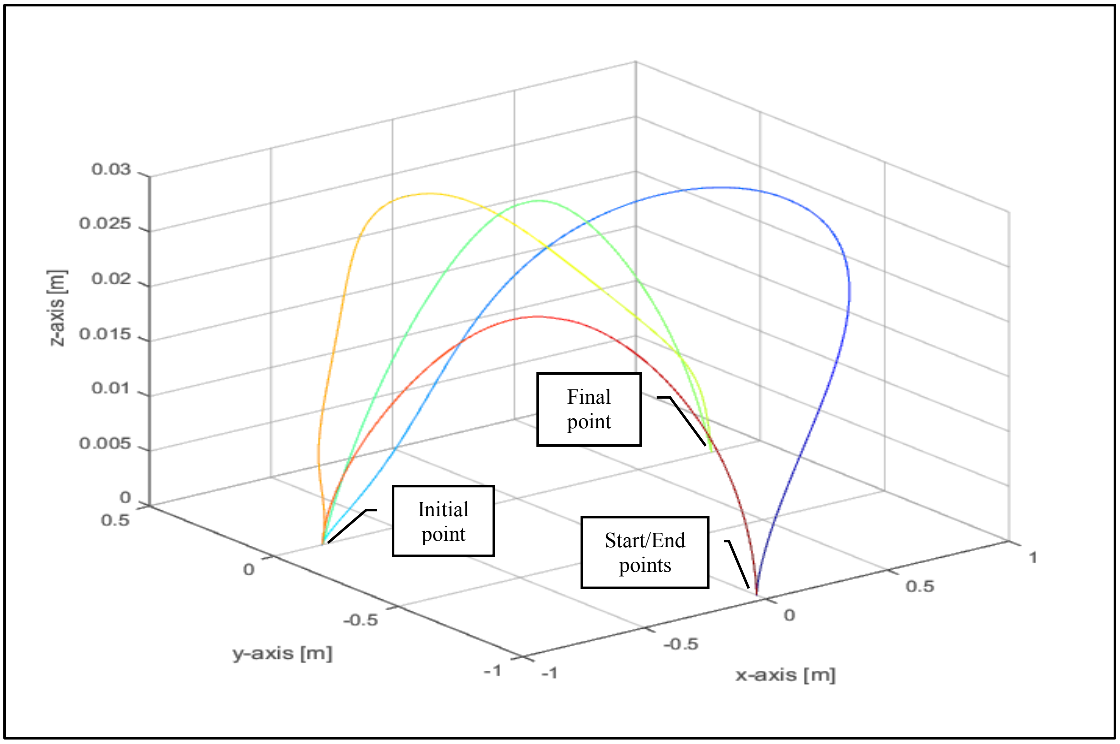


Figure 5:4: The path of the end effector in the case of E1.

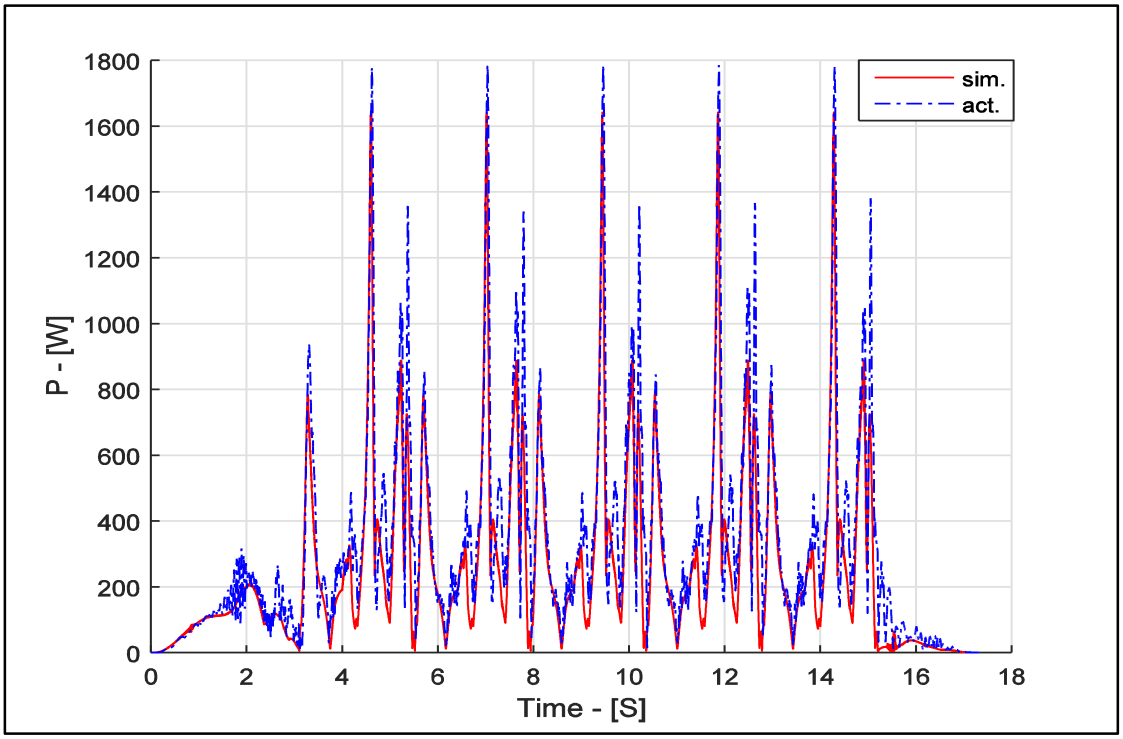


Figure 5:5: Comparison between the actual and the simulation mechanical power consumption for the MAS in the case of E1.

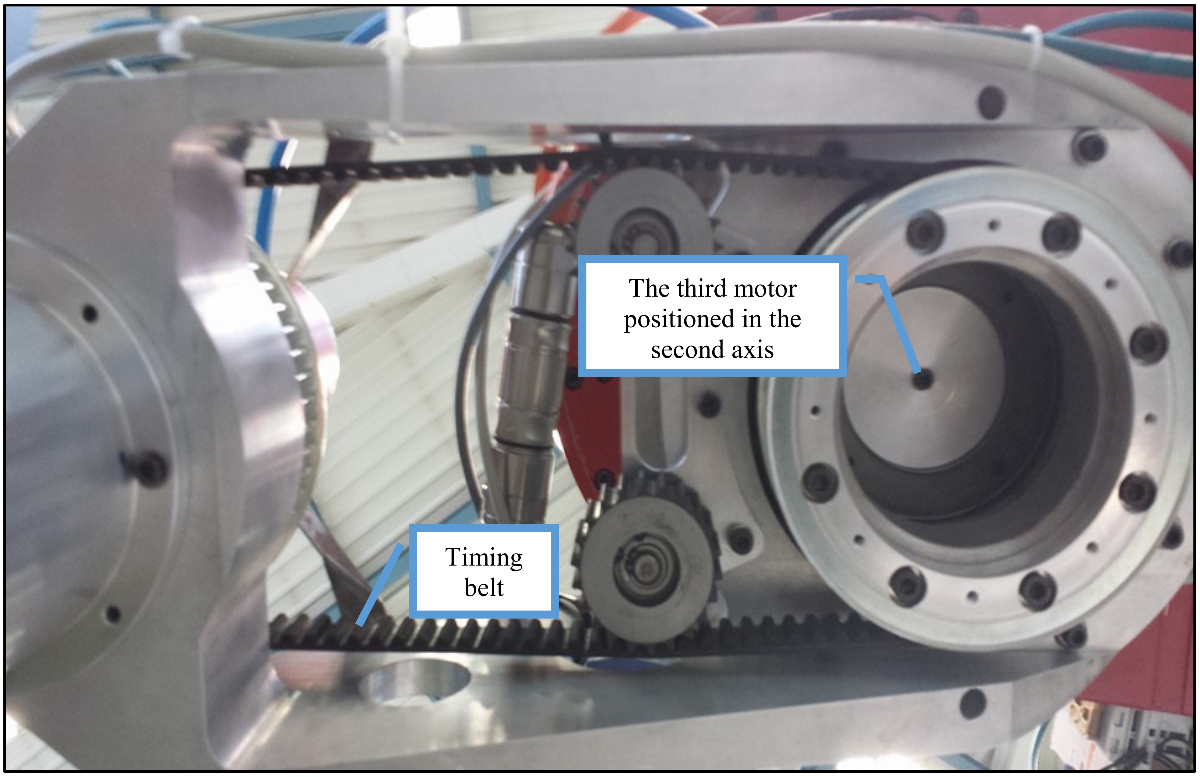


Figure 5:6: The timing belt is connecting between the second axis and the third axis.

However, there is a small value of torque observed in few cases in the third axis in the phase of the UAM (less than 5 Nm); the expected reasons for that are:

- The identified dynamic parameters are not so accurate.
- The existence of the timing belt that is connecting between the second axis and the third axis is reducing the accuracy; see Figure 5:6.
- The interpolation mistakes.
- The backlash error.
- The position control is used by the angular position as a reference signal instead of using the input torque.
- Mistake in the design phase, which is there is existence of an offset between the centre of percussion and the end effector location. While the necessary condition for decreasing the torque in the third axis to zero is to locate the end effector and the centre of percussion at the same point [Bre-13, p. 88].

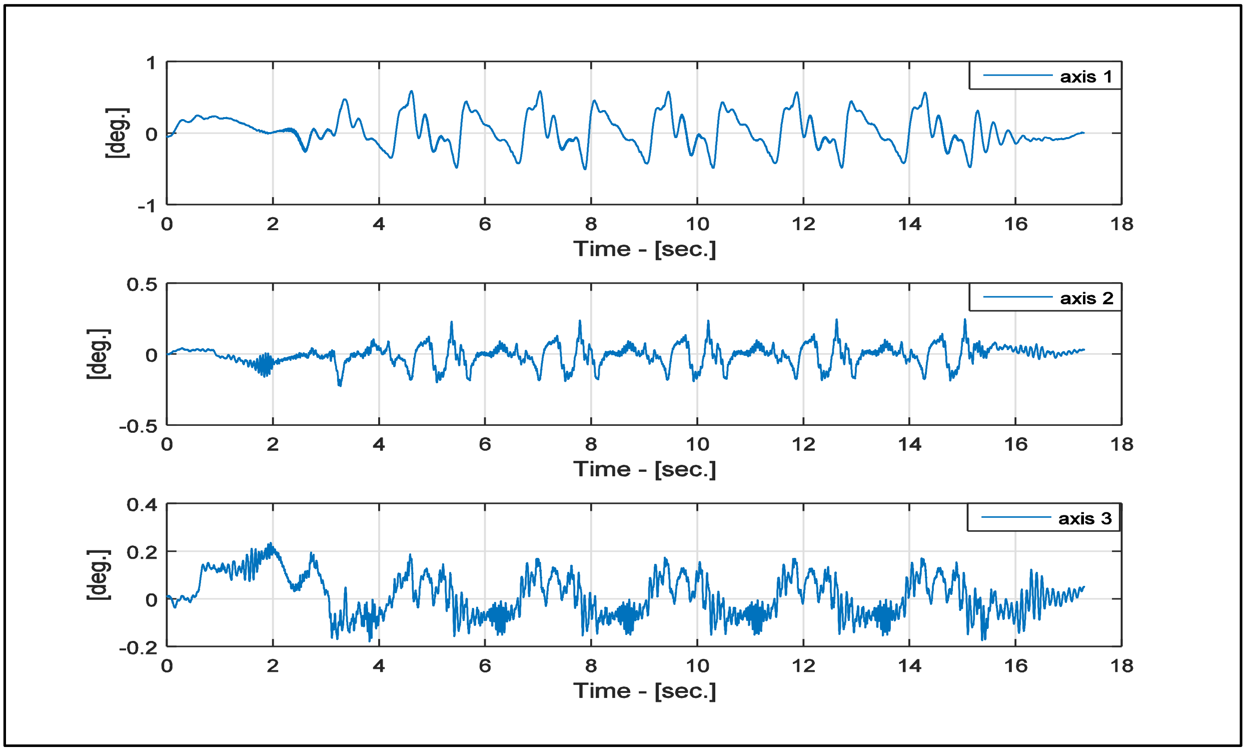


Figure 5:7: The backlash errors for the MAS where it is measured in the case of E1.

The results for the E1 have been achieved after tuning the control parameters for the position control scheme. Consequently, the backlash error measured to evaluate the control scheme and the result is shown in Figure 5:7. In addition, a comparison between the real response and the desired response for the angular positions and angular velocities for each axis in the MAS are shown in Figure 5:8.

Two other experiments executed to test the ability of the robot to perform pick and place tasks for heavy and light payloads respectively. In addition, these experiments evaluate the relative error between the simulation and experimental results for each axis in the MAS.

The E2 was designed to test the ability of the robot to carry a heavy payload. The payload number 5 (see Table 5:2) is used in the E2 and the time used for the NSM was 200 ms. In addition, the distance between the initial point and the final point was 1.4 m and the cycle time was 2.1 s. This experiment has only one cycle. The achieved results, the robot succeeded in executing the required task and the results of the power consumption and the relative error for each axis in MAS are shown in Table 5:3.

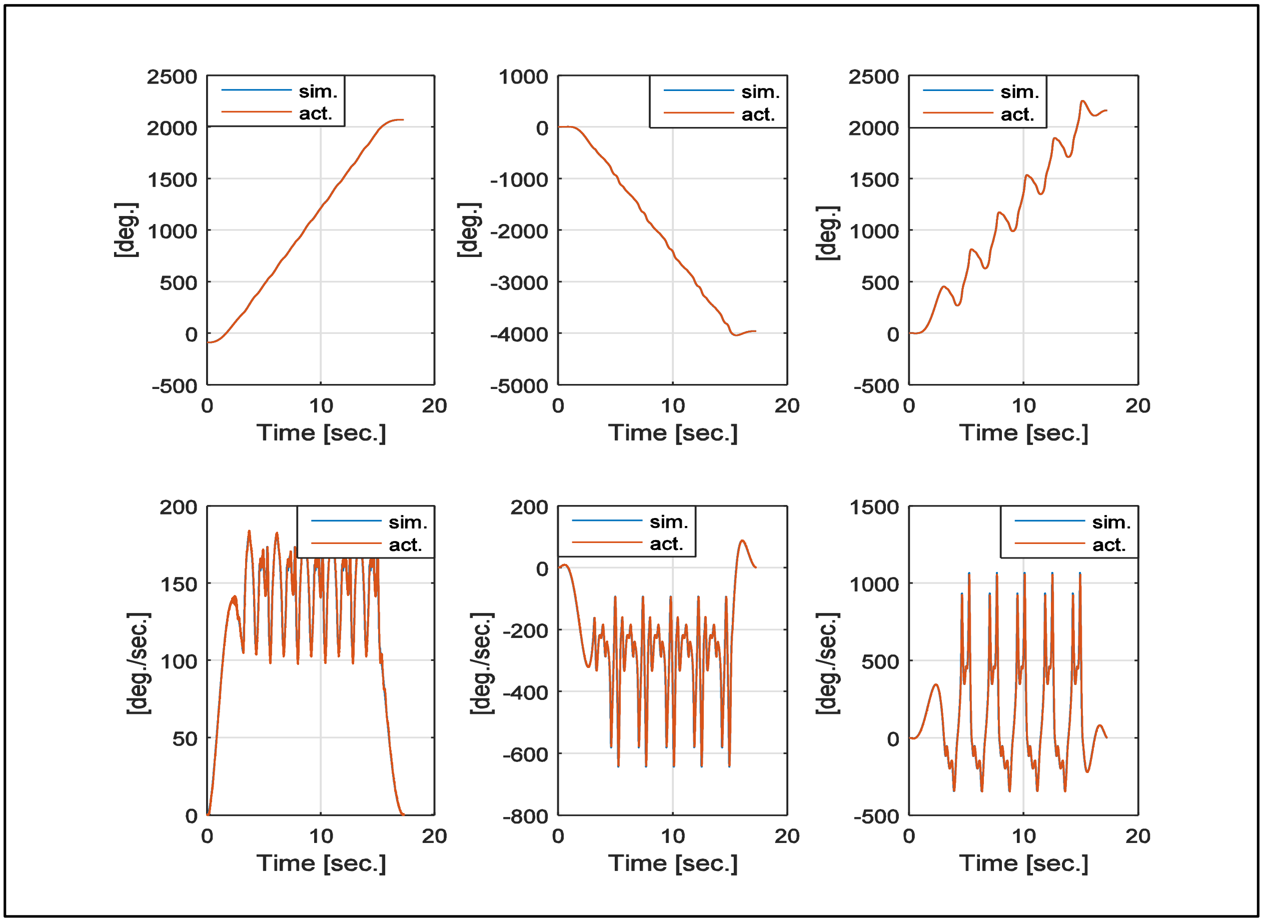


Figure 5:8: Comparison between the simulated and the experimental angular positions and angular velocities for the MAS in the case of E1.

Table 5:3: Comparison between the of average of the simulated and experimental results for the mechanical power consumption in the case of E2.

Axis No.	The total of the average mechanical power consumption		Relative error
	Simulation results	Experiment results	
	[W]	[W]	[%]
1	113.2	165.79	46.46
2	133.51	174.53	30.72
3	24.39	25.93	6.31
MAS	271.1	366.26	35.1

The E3 was designed to test the ability of the robot to perform pick and place tasks for a lightweight, but using a short period for the time of the NSM. On this basis, the used payload was the payload number one (see Table 5:2) and the time for the NSM was 30 ms. In addition, the distance between the initial point and the final point was 1.4 m while the cycle time for the robot was 1.96 s and it was programmed for executing one cycle only. Consequently, the robot performed the task and the achieved results of simulated and experimental power consumption with the relative error for each axis in MAS are shown in Table 5:4.

Table 5:4: Comparison between the of average of the simulated and experimental results for the mechanical power consumption in the case of E3.

Axis No.	**The total of the average mechanical power consumption**		**Relative error**
	Simulation results	**Experiment results**	
	[W]	**[W]**	**[%]**
1	106.32	139.58	31.28
2	123.74	154.20	24.62
3	20.07	20.5	2.14
MAS	250.14	314.27	25.64

It is obvious from all these experiments that the maximum relative error achieved between the simulation and experimental results for the total of the average mechanical power consumption was 35.1% and this occurred when a heavy payload is picked and within the minimum achieved cycle time. The main source for this error is the first axis, because the mass and the inertia for the first axis are too high. Moreover, the first axis as usual must accelerate the rest of the system and this causes a backlash in this axis if the actuator in axis one is not so strong. Consequently, other constraints are neglected in the position control scheme must be taken into considerations to improve the robot performance. These considerations can be summarized by the maximum allowable change rates of the torque inputs and the maximum limits for the torque inputs.

Another point is clarified here. It is observed in Table 5:3 and Table 5:4 that power was consumed in the third axis and the reasons are: in the SM phase, in the EM phase, and in the NSM phase, providing the third axis by torque is allowed, i.e. this axis in these phases is another active axis, while in the UAM phase, it is not allowed. Finally, it is important to mention the average of the computation time for calculating the UAM trajectories by using the QL algorithm is one minute in the Matlab environment. Therefore, it is expected to reduce the computation time from ten to one hundred times if the code is reprogrammed using C or C++.

5.2 Trajectory planning and the optimal control scheme

The optimal control scheme uses the AMPC and the successive linearization process to control the motion of the MAS only as shown in Figure 4:5. The AMPC is a discrete controller and it has the ability to solve a QP problem to optimize the performance of the robot. This controller satisfies the requirements for an efficient controller. These requirements are: the ability to predict the behaviour of the robot in the prediction horizon, the ability to minimize the backlash errors, and it can minimize the input signals or it has the ability to become near the input signals if it exists as stated in [Mat-14]. It has the ability to take into consideration the change rates of the inputs in each axis and the importance of this point is shown in E2 in the last section. It has the ability to take into account the maximum and the minimum allowable torque for each axis naturally. In addition, it has the ability to reject the measured or the unmeasured disturbances; see [Mat-14]. However, the ability to reject the disturbance is proved for MPC as stated in [Cam-07, p.63 f.].

The controller is developed for the MAS instead of all the whole axes, because it consumes more time in computation in the second case and this effect on real-time capability. In addition, reducing the power consumption in the MAS is more important than reducing the power in the SAS because the power consumed in the SAS is less than 10% from the total power consumption for the whole robot.

This control scheme is verified based on a simulation by a set of virtual experiments, but two simulations are shown here to avoid the repetition. The E4 used the same trajectories in E2 as reference commands for the optimal control scheme. The specifications of E2 is to perform a pick and place task for the payload number 5 (see Table 5:2) with a cycle time of 2.1 s and for a distance 1.4 m. In addition, the time used for the NSM was 200 ms. As a result, the reference trajectories contain the angular position for each axis, the angular velocity for each axis, the angular acceleration for each axis, and the input torque that is calculated by the trajectory planning algorithm u_{target}. Consequently, the calculated input from the optimal scheme must be close to u_{target}, while the controller parameters are shown in Table 5:5. The expected result is to reduce the relative error of the total power consumption to less than 5% by using this control scheme. However, the results are shown in Table 5:6.

Table 5:5: The used parameters for the AMPC in the case of E4.

Name	Symbol	AMPC configurations	value
Weighting matrix for the outputs	Q	mpcobj.Weights.OV	[8000 8000 8000]
Weighting matrix for the inputs	R	mpcobj.Weights.MV	[0.001 0.001 0.001]
Weighting matrix for the change rates of inputs	S	mpcobj.Weights.MVRate	[0.005 0.005 0.005]
Minimum allowable torque relative to u_{target}	U_{min}	mpcobj.MV.Min	[-1 -1 -1]
Maximum allowable torque relative to u_{target}	U_{max}	mpcobj.MV.Max	[1 1 1]
Scaling factor for the inputs	Δu	Uscale	= [4600 2400 200]
Prediction horizon	P	mpcobj.p	2 ms
Control horizon	C	mpcobj.c	1 ms
Sampling time	T_s	mpcobj.ts	1ms

Table 5:6: Comparison between the power consumed by trajectory planning method and the power consumed by using the optimal control scheme in the case of E4.

Axis No.	**The total of the average mechanical power consumption**		**Relative error**
	Reference results	**The optimal control scheme results**	
	[W]	**[W]**	**[%]**
1	113.13	113.43	0.27
2	133.42	133.46	0.03
3	24.38	24.46	0.33
MAS	270.94	271.35	0.15

It deserves to mention in the E4 that the prediction horizon and the control horizon are chosen with a small period; therefore, there is no necessity to predict the behaviour for a long period, because the model is updated continuously, also to reduce the computation time. In addition, the response based on using the optimal control scheme is compared with the desired response as shown in Figure 5:9.

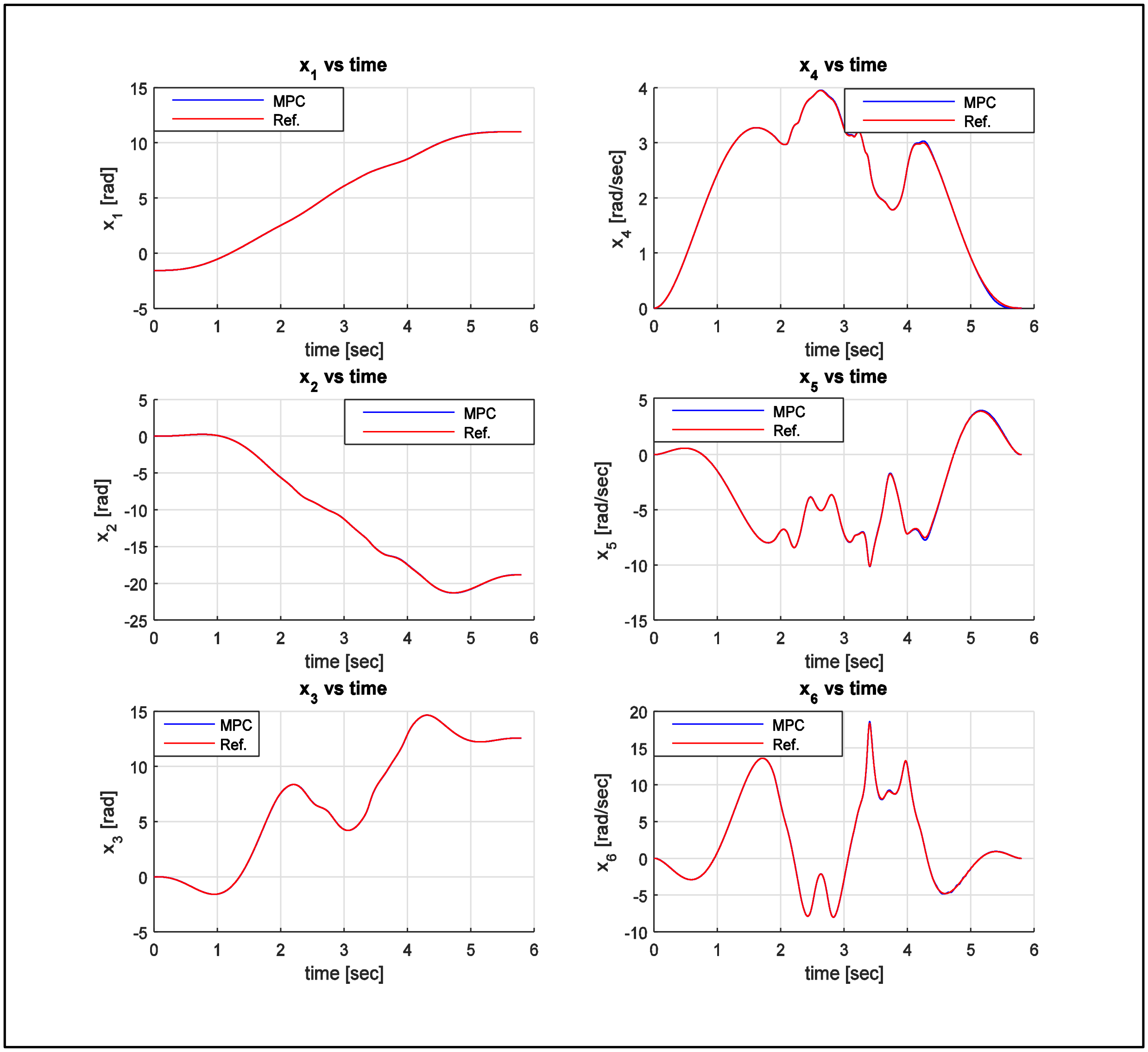


Figure 5:9: Comparison between the desired response and the system response by using the optimal control scheme in the case of E4.

Another simulation is designed to evaluate the optimal control scheme and the computation time for the optimal control scheme. This E5 and the target of this experiment is to execute a pick and place task for five cycles. The payload number two (see Table 5:2) is used and the distance was 1.6 m between the initial and the final point, while the time used for the NSM was 200 ms and the cycle time for this motion was 2 s.

All the controller parameters in Table 5:5 are used except the following parameters: the prediction horizon this time is one (P=1) and the sampling time was changed to 10 ms (T_s=0.01), while the upper and the lower limits are relaxed for the inputs constraint ($\mathrm{U_{min}} = [-1 - 1 - 3\ ]$ and $\mathrm{U_{max}} = [1\ 1\ 3\ ]$). Consequently, the total of the average of the mechanical power consumption for the MAS is shown in Figure 5:10, while the response of the system is shown in Figure 5:11.

The main problem of using the optimal control scheme in a system that has fast dynamics, e.g., robot, is the ability of this control scheme to work in real time framework. However, nowadays, with the revolution in the industry of manufacturing high quality of CPUs, this can be achieved by using a high-quality industrial controller that has a high computation capability. Moreover, it is mentioned in [Mat-16a] that the applicable sampling time for a small multi-inputs-multi-outputs control scheme is in the range of [1-10] ms and this our case.

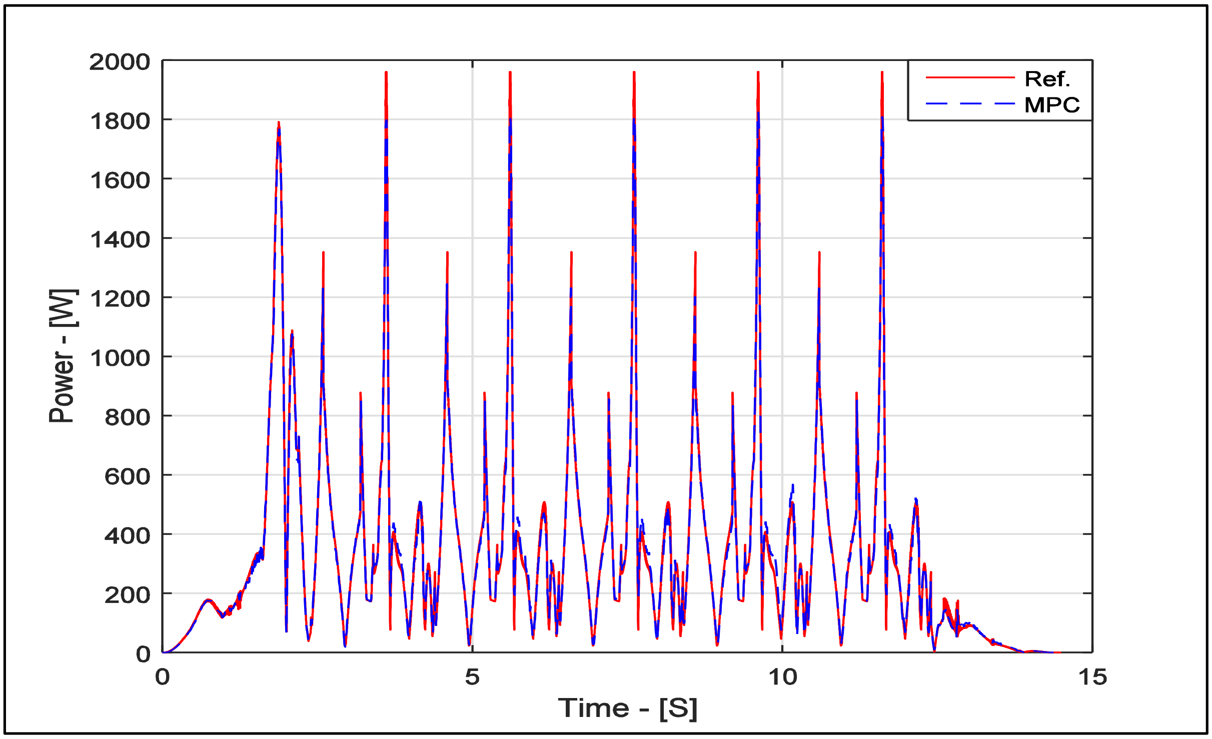


Figure 5:10: The total of the average power consumption for the MAS in the case of E5.

In the case of E5, the computation time is measured in Simulink to evaluate the ability of the controller to work on real time framework. One of the performance tools in Simulink is used to measure the computation time. This tool is “show profiler report” to measure the computation time for each process in each block in the Simulink scheme; see Figure 4:5.

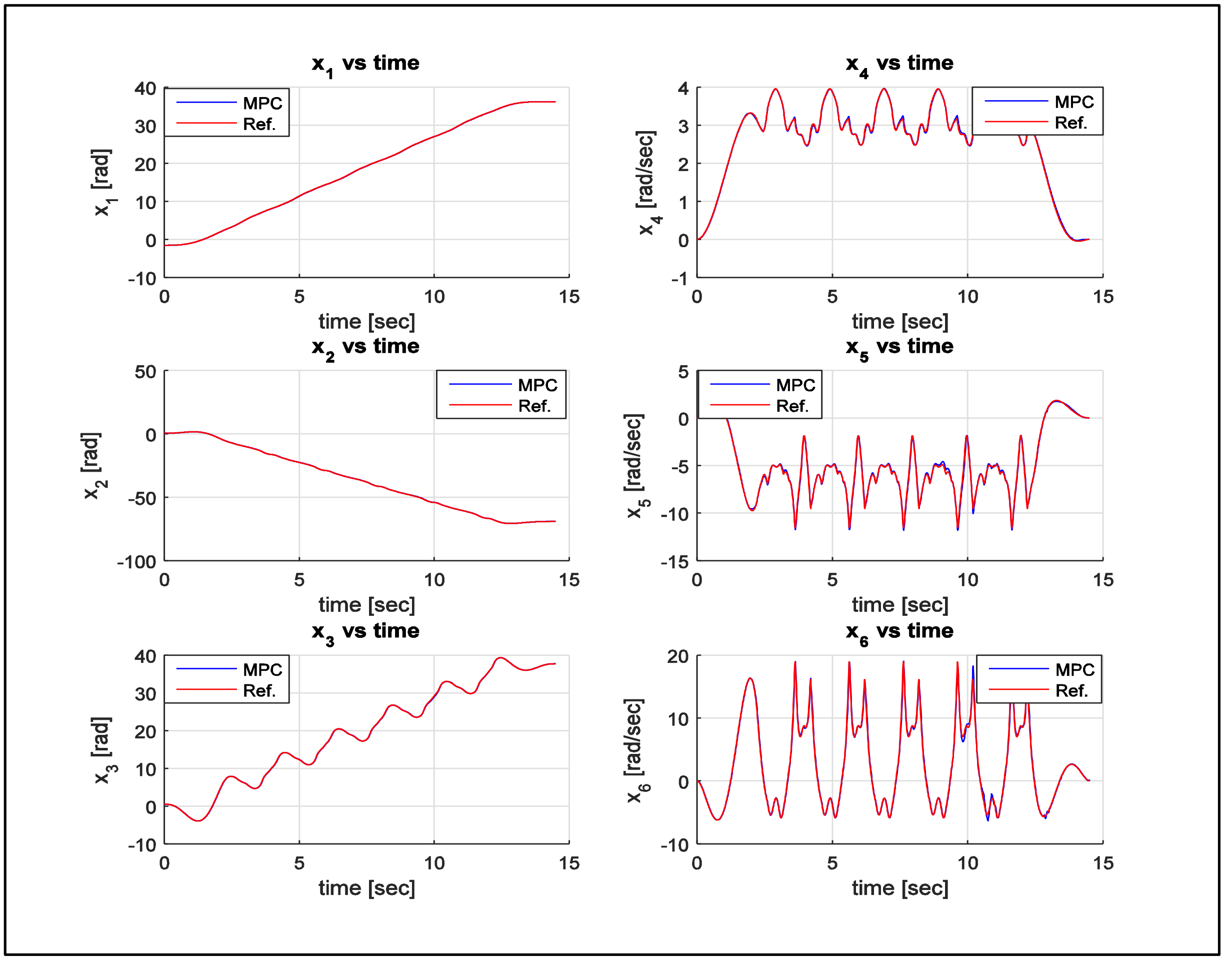


Figure 5:11: Comparison between the desired response of the system and the response by using the optimal control scheme in the case of E5.

The most of the computation time is consumed in the process (A) as shown in Appendix 9.3. This process consumed 7.39 s; also, this part contains the main calculations for the AMPC as shown in the same figure; therefore, converting the Simulink scheme in Figure 4:5 to the C or C++ language will dramatically reduce the computation time [Mat-16a]. Another solution is to use Simulink PLC coder to generate hardware-independent IEC 61131-3 Structured Text, which is accepted in several models of PLCs, e.g., Siemens TIA Portal, and Omron Sysmac Studio [Mat-16a], [Mat-16c]. According to [Bec-15, p. 8], a third solution was suggested by Beckhoff Company by converting the Simulink scheme to TcCOM module. This is achieved by using the Simulink coder toolbox in Matlab to convert the Simulink scheme to TcCOM module created by using the TE1400 TwinCAT target for Matlab Simulink toolbox. The TcCOM module has the capability to work in real time and it has the capability to work under the TwinCAT 3.0 environment.

One of the major challenges in control theory is to develop controllers that can deal with disturbances. However, the AMPC has the ability to deal with measured and unmeasured disturbances [Mat-14]. An example for an unmeasured disturbance is white noise, while an example for a measured disturbance is an external force or torque on a specific point on the structure of the robot. Therefore, a robust optimal control scheme has been developed to simulate the ability of the controller to control the MAS of the 2nd prototype of SAMARA robot in the existing a measured disturbance. This experiment is E6; the behaviour of the MAS for the 2nd prototype of SAMARA robot is simulated by using the robust optimal control scheme to execute a pick and place task for one cycle in the case of the existing torque disturbance in the third joint, where the disturbance value was 5 Nm. The conditions in the E6 were: the duration of the NSM was 200 ms and the cycle time for this motion was 2.1 s. In addition, the payload number five (see Table 5:2) is used and the distance was 1.6 m between the initial and the final point. In the current experiment, the sample time for the controller was 1 ms ($T_s = 1\ ms$) and the prediction horizon was 10 ms ($P = 10$). The relative error of the power consumption for each axis is shown in Table 5:7. Finally, the robust optimal control scheme is shown in Figure 5:12.

Table 5:7: Comparison between the power consumed by trajectory planning method and the power consumed by using the robust optimal control scheme in the case of E6.

Axis No.	**The total of the average mechanical power consumption**		**Relative error**
	Reference results	**The robust optimal control scheme results**	
	[W]	**[W]**	**[%]**
1	113.13	113.27	0.12
2	133.42	131.97	1.09
3	24.38	26.45	8.49
MAS	270.94	271.68	0.27

As expected the control scheme has the ability to track the desired response in the existence of the measured disturbance and it has the ability to reject the measured disturbance. But also the robot consumed more power in the third axis to reject the disturbance input as shown in Table 5:7.

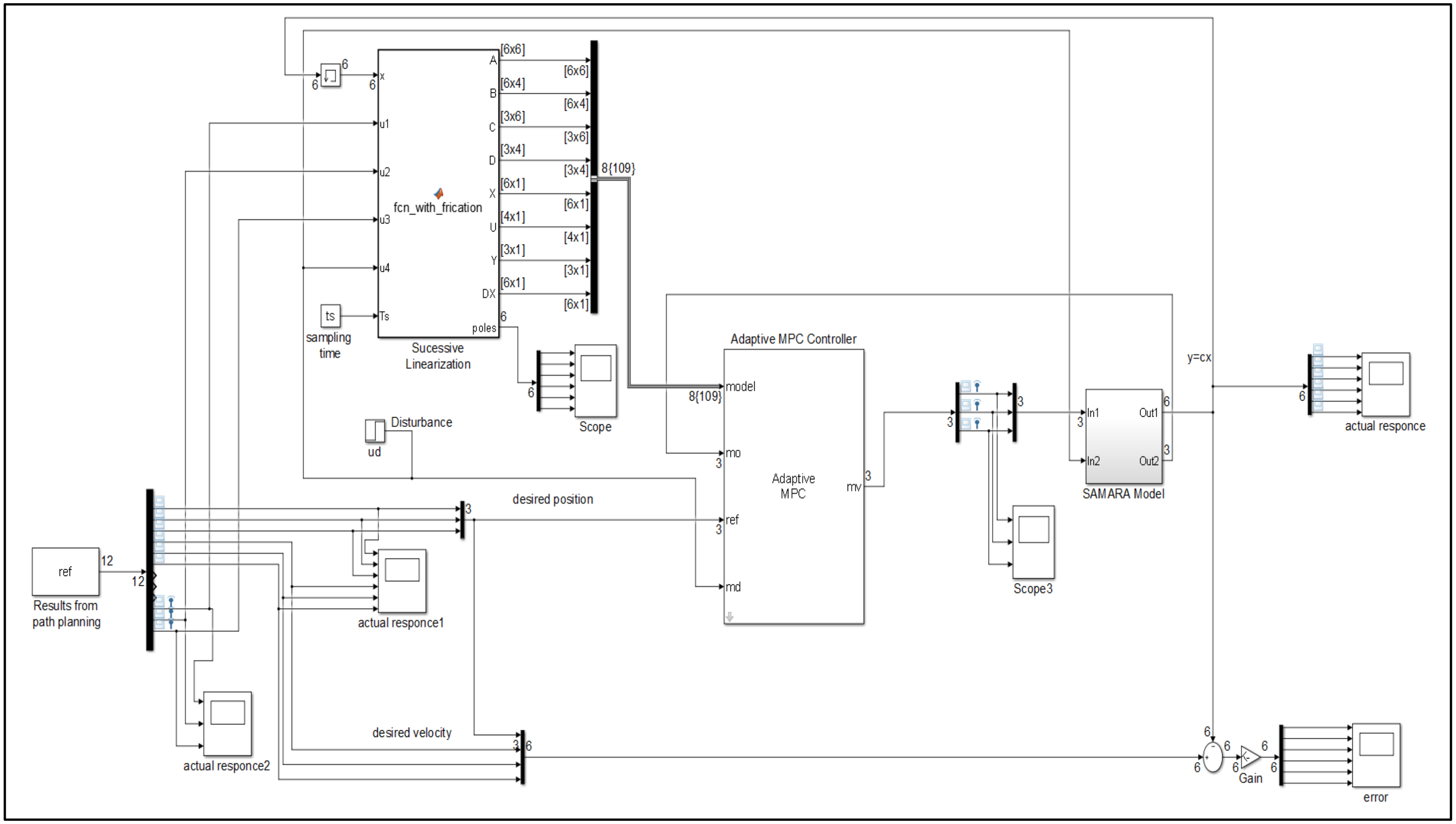


Figure 5:12: Simulink scheme for the robust optimal control scheme.

5.3 Power consumption analysis

This section uses three approaches to analyze power consumption. The first approach focuses on evaluating the enhancement rate for the power saved in comparison with another PTP synchronized trajectory planning algorithm, while the second approach focuses on developing a comparison between the SC1, the 1st prototype of SAMARA, and the 2nd prototype of SAMARA to evaluate the electrical power consumed for each robot. The third approach focuses on comparing the mechanical power consumption for the 2nd prototype of SAMARA and the mechanical power consumption for the simulated model for the SC2. In addition, a comparison between the 1st and the 2nd prototype of SAMARA robots have been achieved based on several perspectives, e.g., the accuracy, the power consumed in the simulation and in the experiment frameworks, the minimum cycle time, the computation time, the total mass of the robot, the DOFs, and other criteria.

The trajectory planning algorithm, which is developed using QL algorithm for the UAM and the analytical method for the NSM, is abbreviated as (M1), while another trajectory planning algorithm (called M2) is developed to compare the power consumption by using it with M1. M2 uses the analytical method for the NSM and PTP synchronized trajectory planning algorithm based on the fifth order polynomial method instead of using the QL algorithm for connecting between the initial and the final points, as shown in Figure 5:13. The fifth order polynomial method is used for the following reasons: it uses three input torques for the MAS, the fifth order polynomial method can connect the initial point and the final point smoothly by using the angular position, the angular velocity and the angular acceleration for each axis at the processing points, and because it is used frequently in trajectory planning for robotics. Then, M2 uses the interpolation to connect the trajectories of the NSM and the trajectories of fifth order polynomial method together into a final trajectory for each axis.

A series of experiments were implemented to execute pick and place tasks, i.e., without a payload for five cycles between two points in the GE2. In the GE2, the position control scheme is used. In each experiment in GE2, the cycle time is decreased continuously, while the power consumption measured in each case; the time used for the NSM was 200 ms, while the fixed distance between the processing points was 1.6 m. Consequently, the enhancement rates of the total power saved for the MAS are calculated by computing the total power consumed using M1 and M2 on simulation and experimental basis, respectively; the results are shown in Table 5:8.

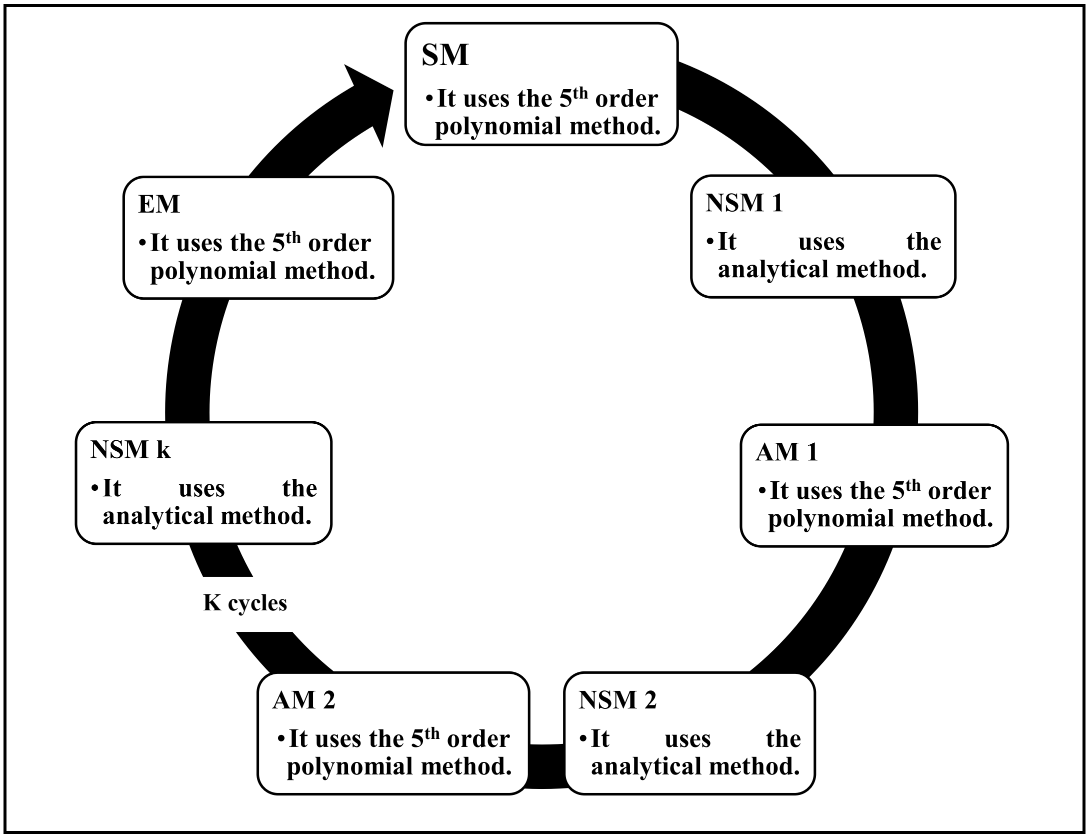

SM
•It uses the 5th order polynomial method.
NSM 1
•It uses the analytical method.
AM 1
•It uses the 5th order polynomial method.
NSM 2
•It uses the analytical method.
AM 2
•It uses the 5th order polynomial method.
K cycles
NSM k
•It uses the analytical method.
EM
•It uses the 5th order polynomial method.

Figure 5:13: The phases of motion for the k cycles by using M2.

It is obvious from Table 5:8 that the average of the total power saved in GE2 is 18.41% in the simulation environment, while in reality, it is 18.17%. Moreover, the experimental results of the total of the average electrical power for all the axes is shown in Figure 5:14. The average of the enhancement rate for the electrical power saved by using M1 is 22.54%. On the other hand, the minimum cycle time achieved using M1 was 2 s, while the minimum cycle time achieved using M2 was 2.2 s, as shown in Table 5:8. Consequently, using M1 instead of using M2 increases the handling rate by 9.1%. All previous results in the case of GE2 were achieved by using the position control scheme. However, if the optimal control scheme is used instead of the position control scheme, with M1 as a trajectory planning method, the previous results are expected to become better, because the relative error between the desired response (simulation) and the response of the system by using the optimal control scheme is less than 1%, as verified in Table 5:6 and Figure 5:10. Therefore, the expected enhancement rate for the total average of the mechanical power saved by using the M1 and the optimal control scheme in comparison to the experimental results of M2, as a trajectory planning algorithm with the position control scheme, is expected to be 35% on average.

Table 5:8: Comparison between M1 and M2 for the total of the average mechanical power consumption for each axis in the MAS in the case of GE2.

Cycle time	**Simulation results**			**Experimental results**		
	Power		**Enhancement rate for power saved**	**Power**		**Enhancement rate for power saved**
	M1	**M2**		**M1**	**M2**	
[sec]	[watt]	[watt]	%	[watt]	[watt]	%
2	334.65	395.82	15.45	456.00	n.a.	n.a.
2.1	272.97	377.90	27.77	361.43	n.a.	n.a.
2.2	247.88	283.01	12.41	329.08	357.42	7.93
2.4	270.66	351.71	23.04	340.89	454.28	24.96
2.5	229.49	294.33	22.03	290.14	347.01	16.39
2.6	201.14	247.99	18.89	253.01	330.54	23.46
2.7	180.46	209.53	13.87	231.15	275.25	16.02
2.8	153.39	178.03	13.84	197.86	248.14	20.26
	Average		18.41	Average		18.17

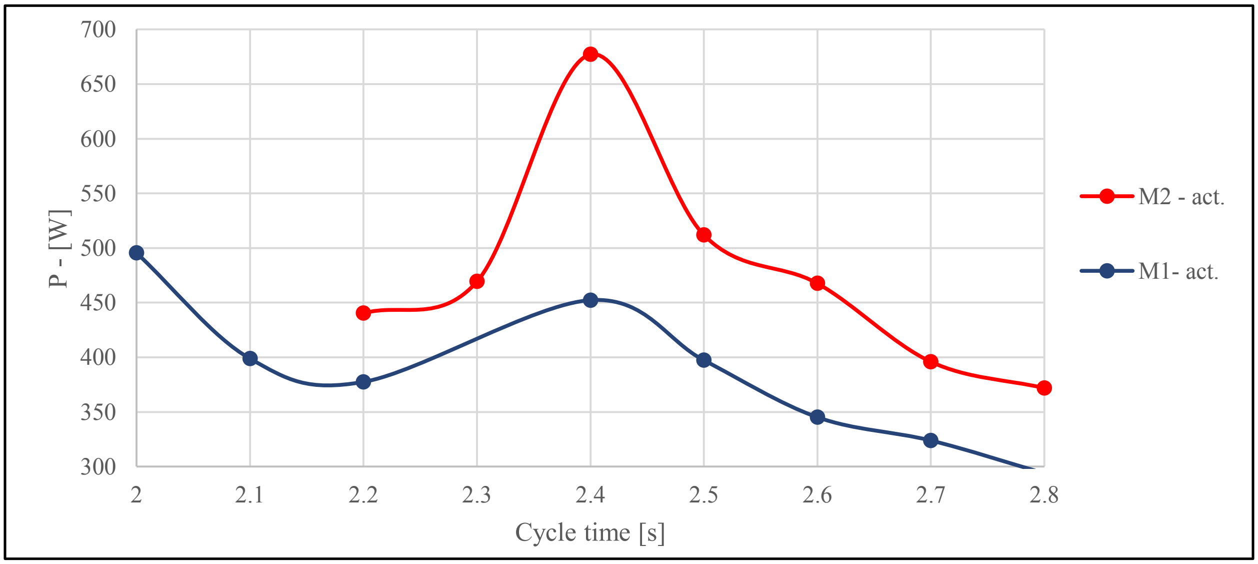


Figure 5:14: The experimental result of the total of the average electrical power consumption for all the axes by using M1 and M2 under the specification of the case of GE2.

The second approach focuses on developing a comparison between the 1st prototype of SAMARA, the 2nd prototype of the SAMARA robot, and the SC1 robot, to compare the electrical power consumed for each one, then, as a third approach to evaluate the mechanical power consumption for the 2nd prototype of SAMARA robot, the model of the SC2 when the PTP synchronized trajectory planning algorithm is used, and for the model of the SC2 when the QL algorithm is used.

Unfortunately, both SC1 and SC2, or any other SCARA robot were not available in the laboratory during the period of this research. However, the results of the power consumption of the SC1 are cited from [Bre-13, p. 133], while the results of the power consumption of the SC2 are simulated based on the dynamic parameters as published in [Inc-07]. However, some of the dynamic parameters were missing in the last reference, but after contacting the author, we were provided with the remaining dynamic parameters for more details about this simulation see Appendix 9.4.

The comparison is between the experimental results for the consumed electrical power by the 1st and 2nd prototypes of SAMARA, and the consumed electrical power by the SC1; this group of experiments is the GE3. The cycle time is decreased continuously each time until the minimum cycle time for each robot is achieved. In this this group of experiments, the 2nd prototype of SAMARA robot performed several pick and place tasks for various payloads; for example, it executed the first pick and place task without any payload, then performed the same motion but with payloads 2, 3, and 4 (see Table 5:2), while the other robots perform the required tasks, but without carrying any payloads. The distance between the processing points was 1.6 m in the case of using 2nd prototype of SAMARA and in the case of the SC1, while the 1st prototype of SAMARA moved for a distance 1.8 m instead of 1.6 m [Bre-13, p. 133]. According to [Bre-13, p. 133], two experiments have been developed and the results measured the power consumption for the 1st prototype of SAMARA by using two industrial controllers from different companies. The first experiment is called (FE) and it used a controller from the Bosch Company to execute the required motion. In this experiment, the backlash errors were small, while in the second experiment (SE), another controller is used from Beckhoff Company; however, in this experiment the accuracy was very poor because the backlash errors were too high, i.e., the robot did not track the reference commands adequately. On this basis, the results of the SE are shown here, but the comparison does not depend on their results. Also, the minimum cycle time for the 1st prototype of SAMARA was 2.15 s, which is not 1.57 s as shown in point A (see Figure 5:15), because the motion for the robot in point "A" is stable for only one cycle while the motion for the 1st prototype of SAMARA was stable for multi-cycles when the cycle time was 2.15 s.

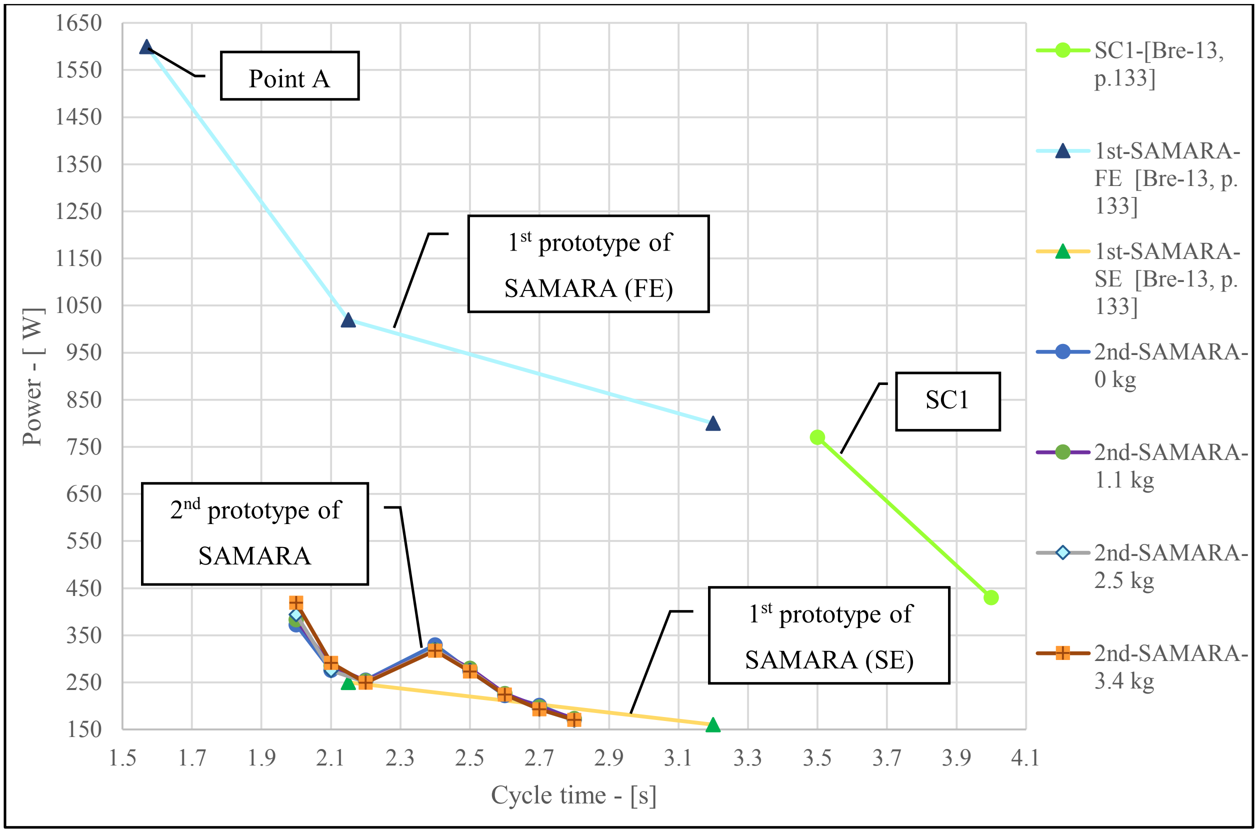


Figure 5:15: The electrical power consumed [7] for: the 1st prototype of SAMARA, the 2nd prototype of SAMARA, and the SC1 in the case of GE3.

The results of GE3 are shown in Figure 5:15. It was observed that the 2nd prototype of SAMARA succeeded in performing 30 picks per minute (ppm), while the SC1 succeeded in performing only 17.14 ppm under the same circumstances. The 2nd prototype of SAMARA consumed less than 0.45 kW in all the cases, while the SC1 consumed around 0.77 kW when the cycle time was reduced to the minimum cycle time. Therefore, if the 2nd prototype of SAMARA is used instead of the SC1, at least, more than 43.11% of the power will be saved and the handling rate will improve by 42.87%, as shown in Figure 5:15. On the other hand, a comparison between the 2nd prototype of SAMARA and the FE for the 1st prototype of SAMARA has been investigated as a rough indication to evaluate the performance of the QL against the EA. In this experiment, the 2nd prototype of SAMARA saved up to 70% of the power consumed by the 1st prototype of SAMARA in the FE. The 2nd prototype of SAMARA can execute 30 ppm for the heavy payloads, but for light payloads, this number increases to 31.5 ppm by reducing the time for the NSM. However, the 1st

[7] The standby power consumption is removed in all these measurements.

prototype of SAMARA can execute 28 ppm. Consequently, the cycle time improved by 7% and 11.63% for the heavy and for the light payloads, respectively. Finally, a comparison between the 1st and the 2nd prototypes of SAMARA is shown in Table 5:9 to evaluate the enhancement in this project.

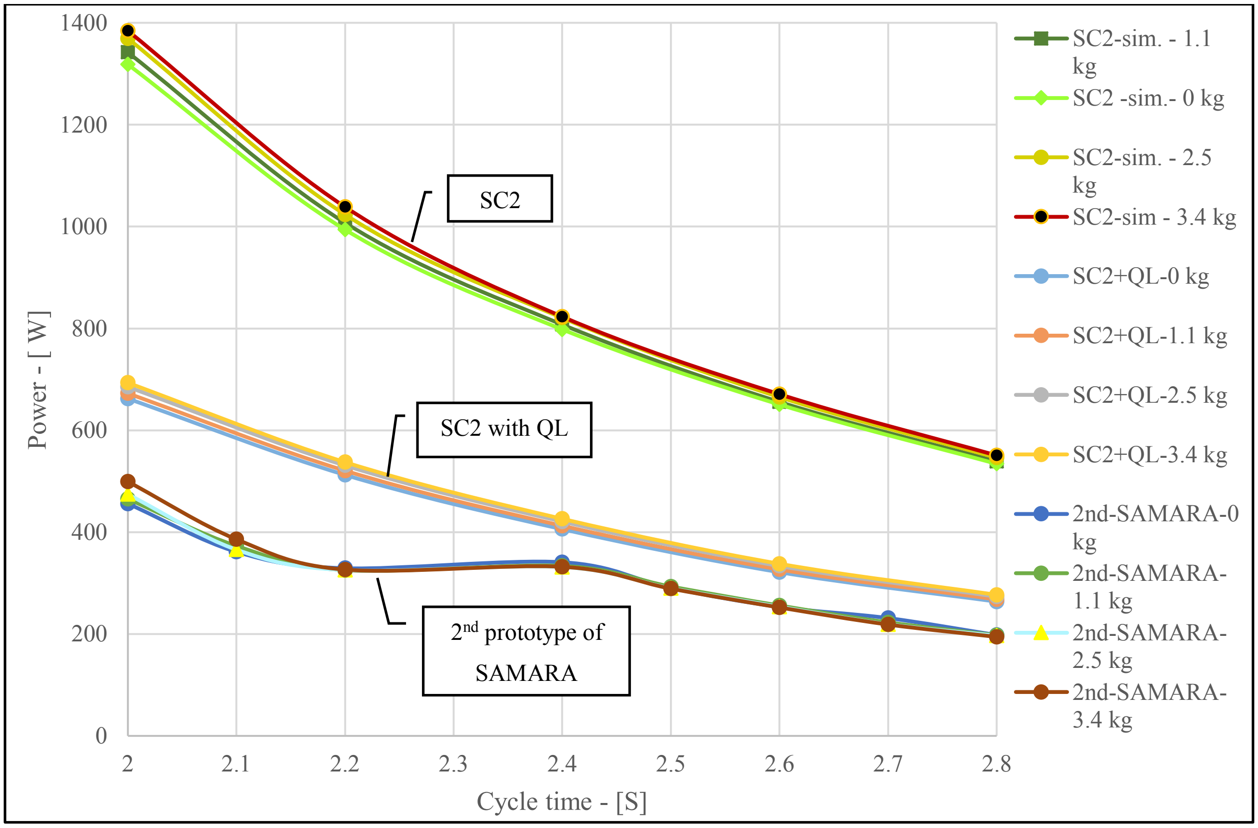


Figure 5:16: The mecahnical power consumed for the 2nd prototype of SAMARA, the SC2, the SC2 when the QL algorithm is used for trajectory planning in the case of GE4.

The third approach focused on developing a comparison between the experimental results for the consumed mechanical power for the 2nd prototype of SAMARA and the simulated consumed mechanical power for SC2 which is called the GE4. The conditions for GE4 are similar to the conditions and goals for GE3, with slight modifications. The modifications are that SC2 is used instead of SC1, and the motion of the SC2 is simulated when it does not carry a payload, and when it carries the payloads 2, 3, and 4 (see Table 5:1). It deserves mention that both robots are moved to a distance of 1.6 m, and the cycle time decreased continuously in each attempt. In the GE4, 60% to 65% of the consumed power can be saved by using the 2nd prototype of SAMARA instead of using the SC2 as shown in Figure 5:16. On the other hand, under the same conditions for the GE4 the QL algorithm is used as trajectory planning algorithm for the motion of the SC2 when it carries the payloads 2, 3, and 4 (see Table 5:2). It is worth to mention that this approach can save power up to 50% but the only point that must be taken into consideration is this method can be applied for SCARA robots that have a ceiling mounting e.g. the model IX-HNN5020H or the model IX-HNN6020H from IAI Company because until now the QL doesn’t take into account the allowable range for each axis [IAI-16].

Table 5:9: Comparison between the 1st and the 2nd prototypes of SAMARA.

<table>
<tr><th>-</th><th>Item</th><th>Reference</th><th>Unit</th><th>1st prototype of SAMARA</th><th colspan="2">2nd prototype of SAMARA</th></tr>
<tr><td rowspan="8">Concepts</td><td>Momentum reservation concept and NSM.</td><td>[Bre-08]</td><td rowspan="8">[-]</td><td>Yes</td><td colspan="2">Yes</td></tr>
<tr><td>Optimization algorithm for the UAM.</td><td>[Bre-10]
[Alb-15]</td><td>EA</td><td colspan="2">QL</td></tr>
<tr><td>Position control scheme.</td><td>[Bre-13, p. 134]
[Alb-15]</td><td>Yes</td><td colspan="2">Yes</td></tr>
<tr><td>Optimal control scheme.</td><td>[Alb-16]</td><td>No</td><td colspan="2">Yes</td></tr>
<tr><td>Classification of the optimization method.</td><td>[Alb-15]</td><td>Stochastic</td><td colspan="2">Deterministic</td></tr>
<tr><td>Methodology to reduce the cycle time.</td><td>-----</td><td>No</td><td colspan="2">Yes</td></tr>
<tr><td>The ability to perform a physical pick and place task.</td><td>[Bre-13]</td><td>No</td><td colspan="2">Yes</td></tr>
<tr><td>DOFs.</td><td>[Bre-13, p. 97]
[Alb-16]</td><td>Three</td><td colspan="2">Five</td></tr>
<tr><td rowspan="10">Features</td><td>Mass of the robot.</td><td>[Bre-13, p. 117]</td><td>[kg]</td><td>197.91</td><td colspan="2">228.37</td></tr>
<tr><td>NSM time.</td><td>[Bre-13]</td><td>[ms]</td><td>30</td><td colspan="2">30-200</td></tr>
<tr><td rowspan="2">The average of the relative error between the real and the simulation power consumption.</td><td rowspan="2">[Bre-13, p. 134]</td><td rowspan="2">[%]</td><td rowspan="2">200%-400%</td><td colspan="2"><30%[8]</td></tr>
<tr><td colspan="2"><1%[9]</td></tr>
<tr><td rowspan="2">Minimum cycle time for five cycles.</td><td rowspan="2">[Bre-13, p. 133]</td><td rowspan="2">[s]</td><td rowspan="2">2.15</td><td>>0.5 kg</td><td>2</td></tr>
<tr><td><0.5 kg</td><td>1.9</td></tr>
<tr><td>Accuracy.</td><td>[Bre-10]</td><td>[cm]</td><td>several</td><td colspan="2">0.2</td></tr>
<tr><td>Computation time.</td><td>[Alb-15]</td><td>[min]</td><td>120</td><td colspan="2">1</td></tr>
<tr><td>Maximum payload.</td><td>-----</td><td>[kg]</td><td>-----</td><td colspan="2">5.5 kg</td></tr>
<tr><td>The electrical power consumed with different cycle time in the case of GE3.</td><td></td><td>[kW]</td><td>[0.8-1.2]</td><td colspan="2">[0.17-.42]</td></tr>
</table>

[8] The results achieved by using the position control scheme.

[9] The results are expected to achieve by using the optimal control scheme.

5.4 Comparison with other industrial robots

One of the innovation aspects in the last century in the field of manufacturing is developing various classes of industrial robots to support diverse applications. Each class of robots has specific features, which is chosen based on the requirements of the industry, e.g., the delta robot usually is used for satisfying high handling rate because the delta robot has the ability to reach very high accelerations, as all actuators are located in the fixed base [Bre-08]. Therefore, in the last decade, researchers focused on developing modern industrial robots that have the ability to satisfy energy efficiency objectives. As a result, the 2nd prototype of SAMARA has the capability to consume less energy in comparison to other industrial robots and it has the capability to decrease the cycle time at the same time.

An essential feature for the 2nd prototype of SAMARA is the ability to pick and place payloads for long travel distances between the processing points with the capability of increasing the energy saving and with the capability for reducing the cycle time as much as possible, independent of the weight of the payload. Therefore, it is desirable to compare the performance of the 2nd prototype of SAMARA with other industrial robots to evaluate its performance.

In this section, the comparison is done based on the DOFs, the range of the robot, the maximum payload, the nominal payload, the mass of the robot, the cycle time, the repeatability, the accuracy, and the vertical stroke. Also, in this comparison, modern industrial robots from reputed companies have been chosen to evaluate their performances against the performance of the 2nd prototype of SAMARA. The results of this comparison based on the previous criteria are shown in Table 5:10. Unfortunately, there is no possibility of comparing the power consumption for each robot with the power consumption of the 2nd prototype of SAMARA, because they are not available in the laboratory or due to the lack on the dynamic models and dynamic parameters.

The 2nd prototype of SAMARA has advanced features in comparison with other industrial robots, e.g., it uses the principle of momentum reservation, which is not used in the design of other industrial robots [Bre-08]. In addition, it can minimize the power consumption in the phase of the UAM by using the QL algorithm; moreover, it has the ability to execute the same average of pick and place tasks regardless of the travel distance between the processing points. In addition, it has the capability to reach further points, because it has a bigger operational space than others has. In addition, it has the ability to reach further points with a short cycle time and with the ability to carry the maximum payload.

As a clarification about the last point, the 2nd prototype of SAMARA can pick the maximum payload (5.5 kg) from the initial point and place it at the final point. Then the end effector comes back to the initial point during a short cycle time while using a long cycle e.g. 25-1600-25 mm. The 2nd prototype of SAMARA can perform 30 ppm as a handling rate for the required tasks under these conditions. The IRB 360-6/1600 robot from ABB Company can perform this task with 18.75 ppm based on using a linear interpolation as a rough estimation. The robot can carry 6 kg as the maximum payload [ABB-13]. In comparison, the model IX-NNN8020H from IAI Company can perform the previous task with a 21.66 ppm based on using a linear interpolation with a capability to pick and place 5 kg [IAI-16]. Nevertheless, it is worth mentioning the robot IX-NNN8020H from IAI Company can reach this handling rate only when using this payload; increasing the payload means the handling rate will become slower [IAI-16]. The only robot that can perform the previous task with a handling rate 31.91 ppm is STÄUBLI TP80 but the disadvantage is this robot cannot carry more than 1 kg as a maximum payload. Consequently, the 2nd prototype of SAMARA optimized several criteria at the same time, e.g., a wide range for a long travel distance, the ability to increase the power saving, and the ability of carrying high payload in a short cycle time. In addition, the cycle time does not depend on the travelling distance or the payloads. However, the author believes the handling rate for the 2nd prototype of SAMARA can be improved by using stronger actuators to avoid the problems of the SM phase, or by reducing the inertia and the mass of the first link. On this basis, the handling rate can increase up to 37.5 ppm with a low power consumption, if the selected time for the NSM is 200 ms as shown in Figure 5:17.

The price for the 2nd prototype of SAMARA about 55 thousand € (T€), which means the 2nd prototype of SAMARA is the most expensive in comparison to the other industrial robots, see Table 5:10. But, this price can be reduced to 38 T€ if the robot is produced in serious like other industrial robots. There is still a gap in the price criterion between the 2nd prototype of SAMARA and the prices of other industrial robots which are around 32,000 €. Satisfying the three desirable requirements from the manufacturing point of view can justify this additional cost e.g. it is obvious from Table 5:10 the 2nd prototype of SAMARA covers more task space with the comparison to the others, also it has the capability to carry payloads up to 5.5 kg in a shorter cycle time. In the majority cases, the improvement on the cycle time reaches to 60%, 38.5%, or 20% as shown in Table 5:10. The only robot that has a shorter cycle time is STÄUBLI TP80 but it cannot carry payloads more than 1 kg.

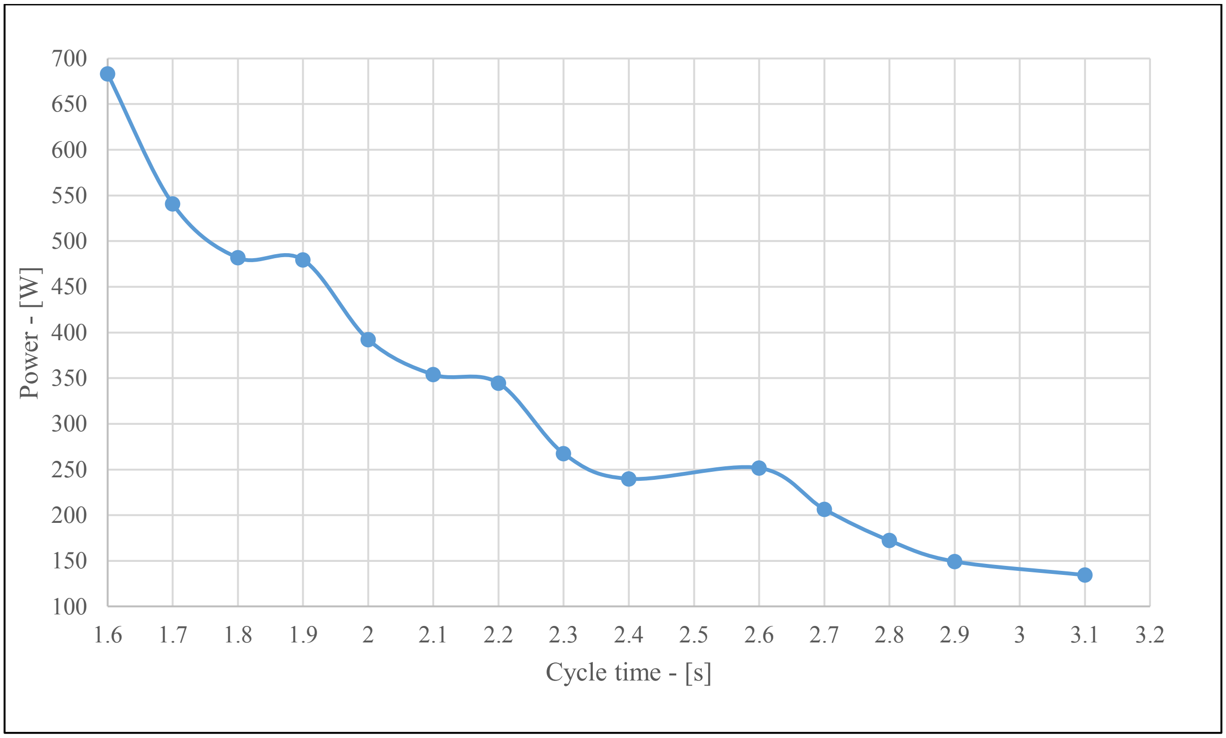


Figure 5:17: The simulation result for the total of the average mechanical power consumption for the MAS if the selected time for the NSM is 200 ms.

Table 5:10: Comparison between the 2nd prototype of SAMARA prototype and other industrial robots.

Item		units	2nd prototype of SAMARA		STÄUBLI TP80		ABB IRB 360-6/1600		Kawasaki2626 YF03N		IAI IX-NNN8020H		KUKA KR6 R900 sixx	
Class of the robot		[-]	Underactuated		SCARA		Delta		Delta		SCARA		Articulated	
DOF			5		4		4		4		4		6	
Max/min diameter		[mm]	1926/982		1600		1600		1300		1600		1802	
Vertical stroke			160		100 or 200		460		500		200 or 400		------	
Repeatability			------		± 0.05		±0.1		±0.1		0.015		±0.03	
Maximum payload		[kg]	5.5		1		6		3		20		6	
Nominal payload			2.75		0.5		------		------		5		------	
Mass of the robot			228.37		68		120 or 145		145		60		52	
cycle time	first cycle[10]	[s]	------		1.0 kg	0.353	6.0 kg	0.60	3.0 kg	0.45	5.0 kg	0.52	1.0 kg	0.458
	second cycle[11]		------		------		6.0 kg	0.80	------		------		------	
	third cycle[12]		< 0.5 kg[13] > 0.5 kg[13]	1.9 2	1.0 kg	1.88[14]	6.0 kg	3.2[14]	n.a.		5.0 kg	2.77[14]	1.0 kg	2.4[14]
Enhancement rate for cycle time of 2nd SAMARA prototype		[%]	-----		1.0 kg	-[1.06-6]	6.0 kg	60	n.a.		5.0 kg	38.5	1.0 kg	20
Price		[T€]	A[15] B[15]	55 38	30		32		34		32		32	
Reference					[Stä-16]		[ABB-13]		[Kaw-16]		[IAI-16]		[Kuk-16b]	

[10] 25-300-25 mm cycle while for the robot IRB 360-6/1600 from ABB and the robot KR6R900 sixx from KUKA the cycle is 25-305-25 mm.

[11] 90-400-90 mm cycle.

[12] 25-1600-25 mm cycle.

[13] In the case the payload < 0.5 kg the selected time for the NSM is 30 ms. While if the payload >0.5 kg the selected time for the NSM is 200 ms.

[14] Predicted value based on the linear interpolation.

[15] A: denotes to the price of the robot for producing one unit only. B: denotes for the predicted price for the robot in the case if the robot is produced in a high scale.

5.5 Conclusions

This chapter verifies M1 as a method to minimize the power consumption and to reduce the cycle time. In addition, this chapter evaluates two control schemes for improving the performance of the 2nd prototype of the SAMARA robot. The reduction of the power consumption has been analyzed by three approaches. The first approach focused on comparing the ability of M1 to reduce the power consumption instead of using M2 under the same circumstances. The achieved results clarify the ability of M1 to increase the electrical power saving by 22.54% with the ability to reduce the cycle time to 2 s instead of 2.2 s. While the other approaches focus on comparing the power consumption for the 1st and the 2nd prototype of SAMARA with other industrial robots based on real measurements or a simulated model for their dynamic to analyze the power consumption for each robot. Consequently, superior results were achieved because the power consumption for the 2nd prototype of SAMARA is less by 60% to 65% in comparison with the power consumption for the SC2. In addition, the 2nd prototype of SAMARA saved power at least more than 43.11% with comparison with the SC1, and it succeed to reduce the cycle time by 42.87%. Moreover, other industrial features have been tested for the 2nd prototype of SAMARA and compared them with those of the other modern industrial robots. What is noticeable is the ability of the robot to carry a high payload while maintaining a short cycle time in combination with large travel distances. Therefore, this improves the cycle time by 38.5% and 60% in comparison with the model IX-NNN8020H from the IAI Company and the model IRB 360-6/1600 from ABB Company, respectively, under the same conditions. While the handling rate for the STÄUBLI TP80 is higher than the handling rate of the 2nd prototype of SAMARA by 1.06% to 6%, but it is worth to mention the STÄUBLI TP80 cannot carry payloads more than 1 kg.

It is noticeable also that the relative error between the experimental and simulation results for M1 have been improved dramatically, e.g., in the case of the worst case, the relative error was less than 35% by using the position control scheme while the average of the relative error in the majority of experiments is less than 30%. However, the relative error it is expected to reduce to less than 1% by using the optimal control scheme instead of the position control scheme.

6 Industrial Applications

In the last decades, industrial robots have been employed as a major equipment of the materials handling systems in diverse industries. The usage of industrial robots into the materials handling systems focus on the transfer of materials, workpieces, or products, or loading or unloading the workpieces or products to or from the machine respectively [Ray-08, p. 189].

Transfer of products or workpieces between the stations is known as pick and place tasks, while picking the heavy workpieces or heavy products from the conveyor and placing them into the pallet is known as palletizing. When, in the case of light workpieces or light products, the robots sort or place the products in trays instead of using the pallets, the trays are collected into racks. The opposite of this operation is called de-palletizing. On the other hand, loading the machine by using the robot means that the robot feeds the machine by the raw material or workpieces to execute the required task only, while unloading the machine by the robot means that the robot is used to take the output of the machine and place it in another place, e.g., the station or a conveyor. However, there are some cases where the robot can perform the loading and the unloading tasks for the same machine at the same time.

Designing or selecting the suitable robots to perform the required tasks is essential to increase the efficiency of the production lines. In the case of selecting the most applicable pick and place robot, several criteria must be taken into consideration, e.g., the product or the workpiece shape, the product or the workpiece mass, the product or the workpiece inertia, the space constraints, the cycle time for the robot, the safety rules, the vertical stroke for the robot, the reachability cover area, the position accuracy, the repeatability, the maximum and the nominal payloads, the environment, and the ability of the gripper to lift the product. However, focusing on one criterion will help clarify the importance of these criteria, e.g., selecting a robot with a cycle time less than the cycle time for the manufacturing task or for the production task makes the performance of the robot an ideal case, while the opposite condition means the robot cannot execute the required task in proper way [Val-99, p. 1077 f.].

In this dissertation, two control schemes are developed and tested for the 2nd prototype of SAMARA to execute the pick and place tasks between processing points as shown before, and it has the ability to sort the workpieces or the products in the trays or boxes as required. The competitive features for the 2nd prototype of SAMARA to execute the required tasks can be summarized as follows:

- It covers a large operational space.
- It can carry workpieces or products up to 5.5 kg for long distances.
- It has the ability to pick and place workpieces or products up to 5.5 kg with 2 s without depending on the mass and inertia of the workpieces or products.
- It consumes less energy due to the principles used in its design and due to the optimization algorithm which minimizes the energy in the UAM phase.

Consequently, this robot can be used in several fields e.g. assembly field, food field, and press lines field based on its features and the requirements of each industry. Two application fields are chosen as exemplary fields. These fields are the food industry and the industries that use the press lines for producing small products or small workpieces.

6.1 Food Industry

Recently, the automated food sector has become an essential concern for the development of this industry because this industry is classified as a mass production industry. Therefore, utilizing automated production lines increases the production capacity, reduces costs and saves time. In 2009, the food industry was ranked as the biggest industry in the EU. At the time, it employed a lot of the manpower, e.g., in the United Kingdom, it employed about 440,000 employees [Wil-10]. Another benefit of using industrial robots is their ability to apply a strict system to maintain hygiene and cleanliness.

Industrial robots can perform several operations through which they support the food industry, e.g., packaging, repackaging, palletizing, de-palletizing, and picking. In this dissertation, the discussion is focused on pick and place operations because the 2nd prototype of SAMARA is designed for these tasks specifically. Therefore, the industrial robots in this field are used to pick the goods or the products from the conveyor and place them, or sort them in the trays or in the boxes as a first step for packaging operations as shown in Figure 6:1. However, performing the opposite task in the opposite flow direction is also possible. Other possibilities such as executing the pick and place tasks between two conveyors or between two stations are still possible solutions if required; see Figure 6:2.

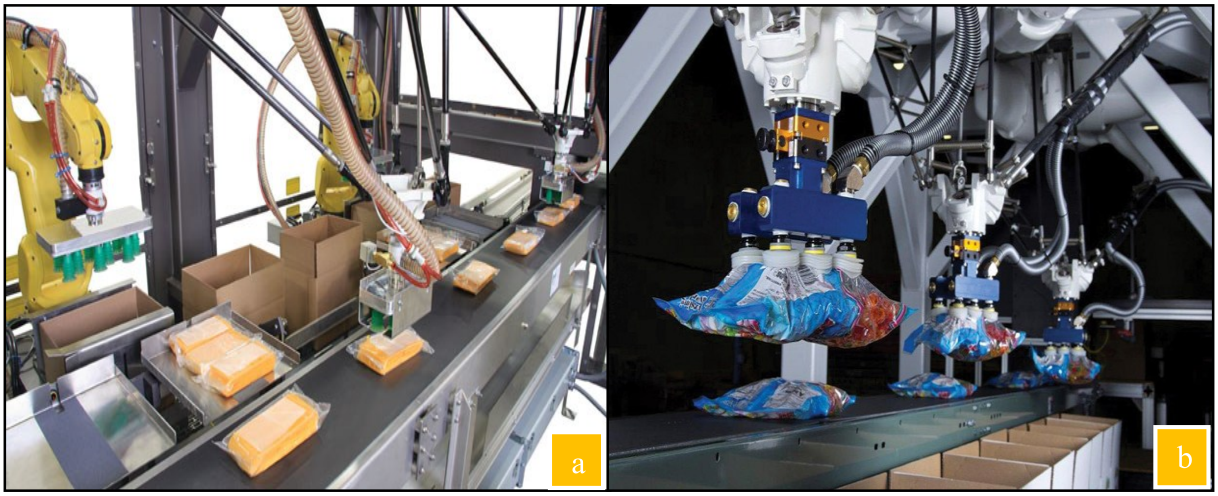


Figure 6:1: Pick and place operation between conveyor and station a) the robots are placing the sliced cheddar cheese in trays [Qib-16] b) the robots are packaging snacks and baked goods in the boxes [Sna-16].

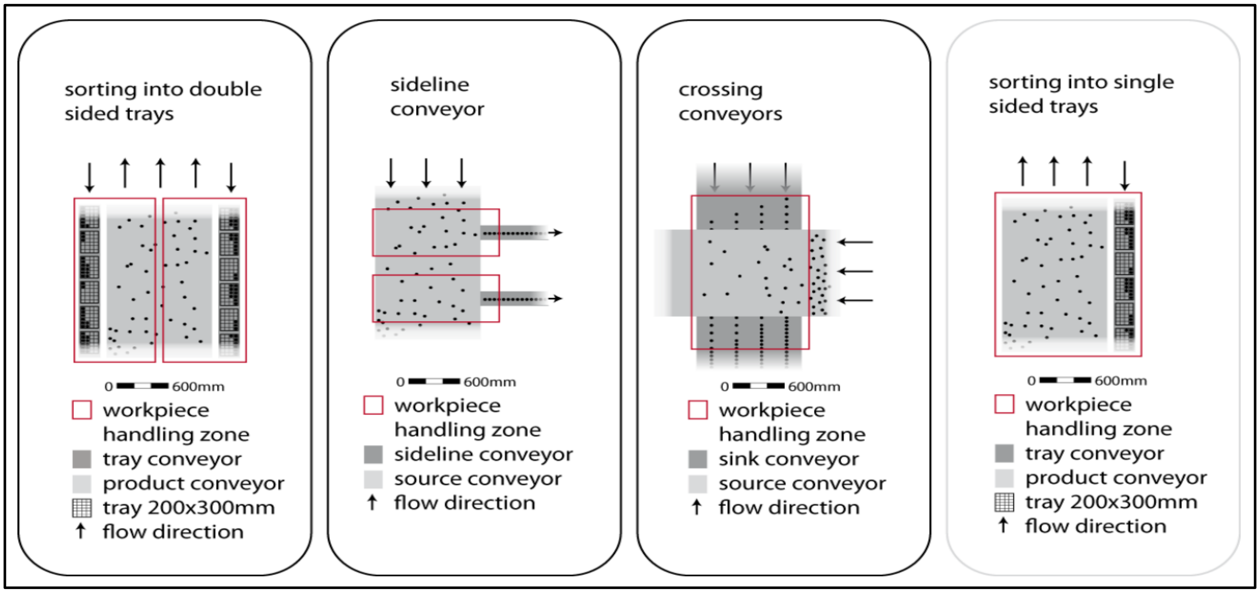


Figure 6:2: Suggested layouts for the food industry application fields [Bre-13, p. 48][16].

The food industry includes a diverse range of products, e.g., frozen pizza, candy, canned foods, bakery goods, chocolates, meats, etc. These products have common factors which must be taken into account in this application field, e.g., the necessity to respect and maintain the production process, to be hygienic and clean by excluding any source of pollution sources; for example, the rotary robot workpieces should be covered by light sheet metal or plastic to prevent leakage of any

[16] Modified picture

impurities to the food products. Also, this industry is classified as a mass production industry, which means that it is desirable to use wider and taller conveyers to increase the efficiency of the handling system. Consequently, it is expected that the traveling distances for the robot will be increased, and so, it is important to utilize robots that can cover a wide area. Also, one of the critical challenges in this application field is to decrease the cycle time to increase productivity. This is can be achieved by using the 2nd prototype of SAMARA; if the robot is covering long traveling distances under the condition of using a taller and broader conveyers' hypotheses as shown in Figure 6:3, or by redesigning the end effector to give it the possibility to pick more than one product in each cycle.

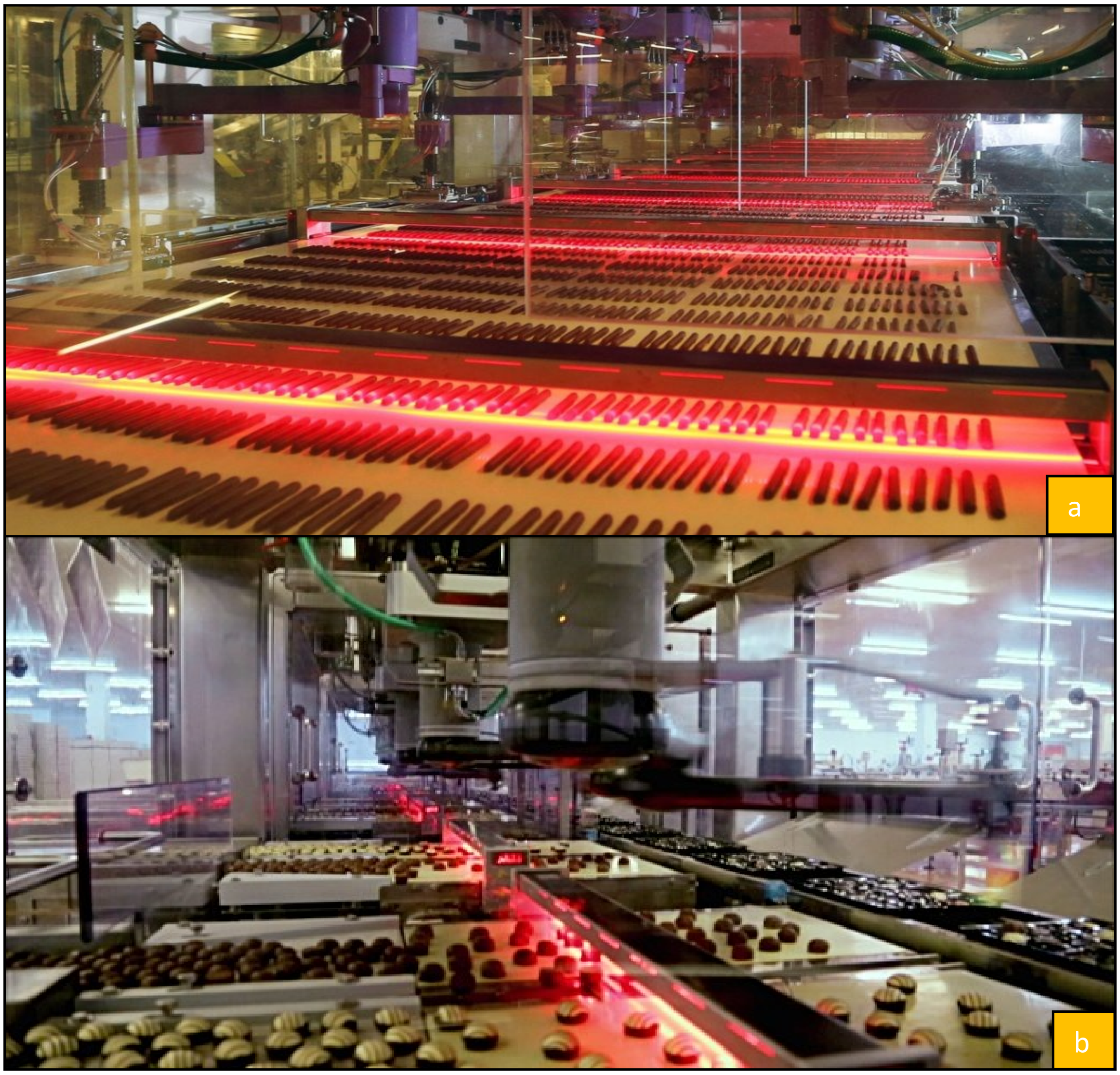


Figure 6:3: Pick and place tasks in long and wide conveyors a) robots sort and place Burton's cadbury fingers [Foo-13] b) robots place chocolates into trays [Cac-16].

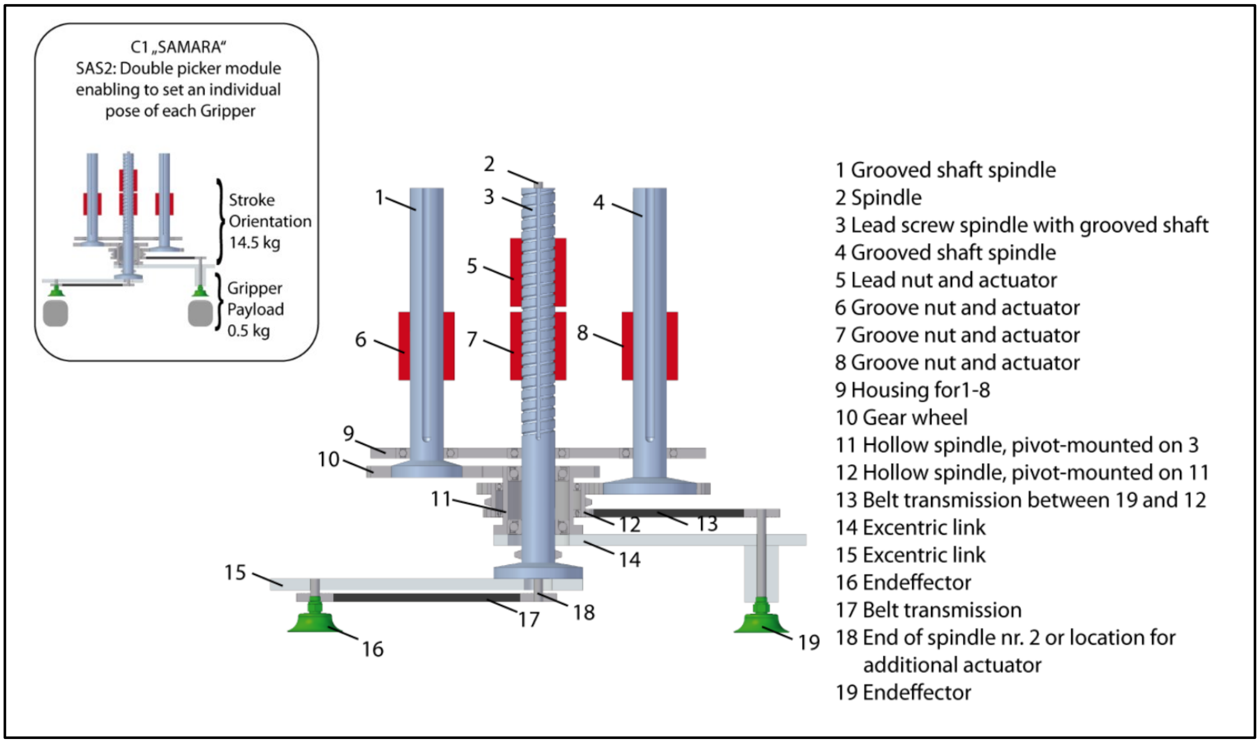

C1 „SAMARA"
SAS2: Double picker module enabling to set an individual pose of each Gripper
Stroke
Orientation
14.5 kg
Gripper
Payload
0.5 kg
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
1 Grooved shaft spindle
2 Spindle
3 Lead screw spindle with grooved shaft
4 Grooved shaft spindle
5 Lead nut and actuator
6 Groove nut and actuator
7 Groove nut and actuator
8 Groove nut and actuator
9 Housing for1-8
10 Gear wheel
11 Hollow spindle, pivot-mounted on 3
12 Hollow spindle, pivot-mounted on 11
13 Belt transmission between 19 and 12
14 Excentric link
15 Excentric link
16 Endeffector
17 Belt transmission
18 End of spindle nr. 2 or location for additional actuator
19 Endeffector

Figure 6:4 : Independent double suction cups grippers [Bre-13, p. 118][17].

It is observed that the masses for the majority of the food products are less than 0.5 kg. Therefore, it is recommended to redesign the end effector by providing multi-suction cups instead of using one suction cup, which is adequate only for the case of heavy masses. The expected result for redesigning the end effector is to increase the productivity rate by the number of suction cups. However, an independent double gripper has been suggested to utilize it in the case that the masses of the products are less than 0.5 kg in total as shown in Figure 6:4 [Bre-13, p. 48]. But there still is an opportunity to utilize the primary design for the end effector in case the mass of the product is more than 0.5 kg (see Figure 2:7). Another possibility is to use the jaw gripper based on the opening and the closing time for the jaw gripper, the shape of the product, and the mass of the product if the section cup is not suitable. Therefore, the model EGP 40-N-N-B from a Schunk GmbH & Co. KG with a gripping cycle time of 200 ms can be used instead of the suction cup if the mass of the product is less than 0.7 kg [Sch-16b].

There remains the possibility to improve the performance of the 2nd prototype of SAMARA by providing the robot with supporting systems to develop the performance, e.g., to achieve the ability

[17] Modified picture

of pick and place of random orientation products, it is recommended to integrate a vision system with the controller to perform this task [Wil-10]. But the constraint of the previous problem is the ability of the controller to solve the optimization problem in the real-time frame work.

6.2 Press Lines Industry

The revolution in developing and using high quality press lines reflects on developing a high quality of parts or products which are used in several application fields, e.g., to develop more reliable doors in the cars with less weight to reduce the fuel cost. To achieve this, a famous class of press lines are utilized to have the capability to use diverse dies to form or blank the workpiece if necessary. These lines which have these capabilities are called universal production lines [Sch-98, p. 33]. But, occasionally, this wide range of flexibility is not necessary in several applications, therefore, one press machine is sufficient to execute the required operation, as shown in Figure 6:5. On the other hand, multi-press machines are used in a straight line layout usually to perform the required multi-tasks sequentially as shown in Figure 6:5.

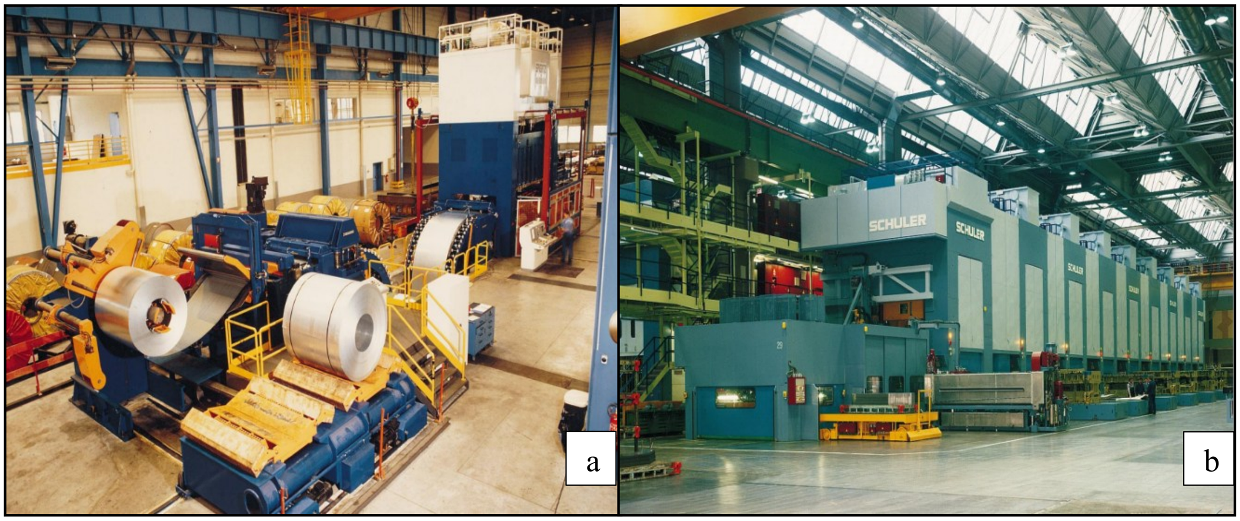


Figure 6:5: Examples of press line layouts a) press machine for blanking the sheet metals [Sch-98, p. 195] b) mechanical press line for multi-task operations [Sch-98, p. 223].

As mentioned in [Aut-16], diverse application fields are used in the sheet metal forming operations in their products, e.g., in the food equipment industry, where, for instance, cooking pots as well as canned goods are formed from sheets metal. The same situation exists in the automotive application field, e.g., the hood, the chassis, the hubcaps, and the fenders, while the sheet metal forming exists in home equipment, e.g., cooker hoods and freezers.

Usually, handling and sorting parts or products in pallets, trays, or racks are performed by using manpower or industrial robots. The industrial robots are used to transfer the parts or the products from different places in the press lines, e.g., between the first station and the first press machine, between the press machines themselves, between the existing conveyer after the last press machine in the press line and the trays or the pallets, as shown in Figure 6:6.

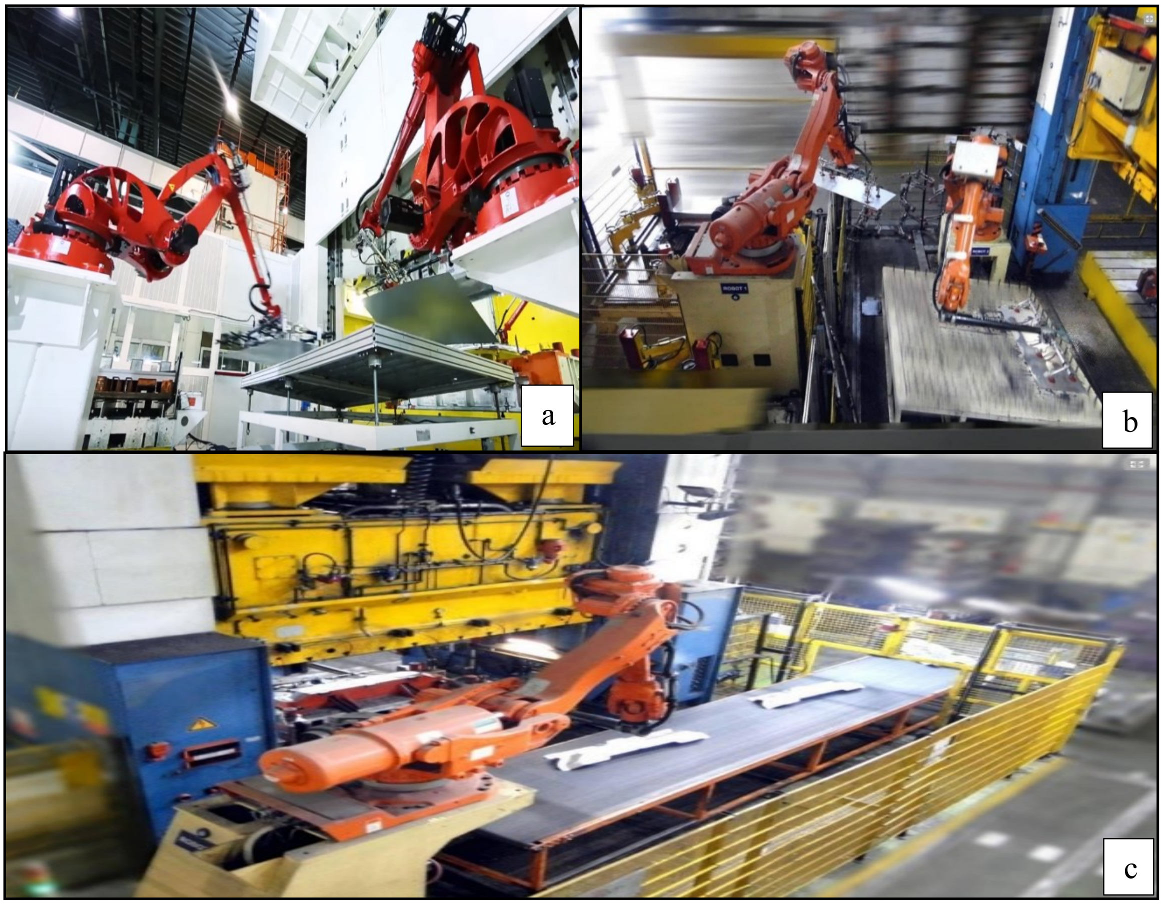


Figure 6:6: The industrial robots are used usually: a) at the start of the press line [Com-16] b) between the press machines [Erm-12] c) at the end of the press line [Erm-12].

Integrating the industrial robots into the press line requires taking into consideration several criteria, e.g., the cycle time for the robot, the cycle time for the press line, the range covered by the robot, the vertical stroke for the robot, the DOFs for the robot, the accessibility, the position accuracy, the repeatability, the shape of the products or the workpieces, the capability of the gripper to grip the products or the workpieces, the masses of the workpieces or the products, and the inertias of the workpieces or the products.

The 2nd prototype of the SAMARA robot has a limited capability to pick and place payloads up to 5.5 kg. Therefore, the decision is made to focus on handling the small products or the small components only at the end of the press line as a case study. The reason for this is because SAMARA is not applicable to transfer the workpieces between the press machines because the trajectory planning algorithm for SAMARA does not have an obstacle avoidance algorithm to avoid obstacles if it exists inside the press machine; the nature of the motion path itself contributes to this. Consequently, the robot could be used to pick small products or small parts from the conveyor and place them into the trays. But, it is also desirable to compare the performance of the 2nd prototype of SAMARA with another handling system for the small and the medium products from Atlas Technologies.

According to [Atl-16], Atlas Technologies provide a solution for the products or the parts racking. The location of this system is at the end of the press line after the last tandem machine, or at the end of the last transfer machine. This system contains three major components which are: multiple output conveyor, two conveyors, and three labor stations as shown in Figure 6:7. The goal of using the output conveyor is to transfer the products or the workpieces between the last press machine and one of the two following conveyors. These two conveyors move horizontally and their target is to distribute the products equally between three stations. In each station, there is a laborer whose job is to relocate the products manually from the station to the racks, as shown in Figure 6:7. This system is controlled automatically to obtain the most reliable performance. The Atlas handling system can handle up to 16 parts per minute in case the parts are manufactured from the transfer press line [Atl-16]. While if this system is integrated at the end of the tandem press line, the system can handle up to 12 parts per minute [Atl-16]. Now, let us focus on integrating the 2nd prototype of SAMARA at the end of the press line to evaluate the performance.

Integrating the 2nd prototype of SAMARA at the end of the press line is required to clarify the cycle time for the press line and the cycle time for SAMARA. If the cycle time for the press line is faster than the cycle time for the robot that means the products accumulate in the exit conveyor, causing a problem and delay. According to [Sch-16a], the production rate per minute for the stamping press line uses 6,3000 kN varies between 34.72 and 41.67 strokes/minute, i.e., the consumed time for each stroke is between 1.73 and 1.44 s. While increasing the utilized forces up to 10,000 kN, which is necessary in several small products, means the production rate per minute reduces between 17.36 and 20.83 strokes/minute, i.e., the consumed time for each stroke is 3.46 and 2.88 s [Sch-16a]. It is observable in the second case that the cycle time for the press line is more than the minimum cycle time achieved for the 2nd prototype of SAMARA robot (the minimum achieved cycle time is 2 s), and therefore, the robot can work in ideal conditions as shown in the third case in Figure 6:8. In cases wherein the required tasks cannot be achieved by using only one robot, two robots are integrated instead of one, as shown in the first and second cases in Figure 6:8.

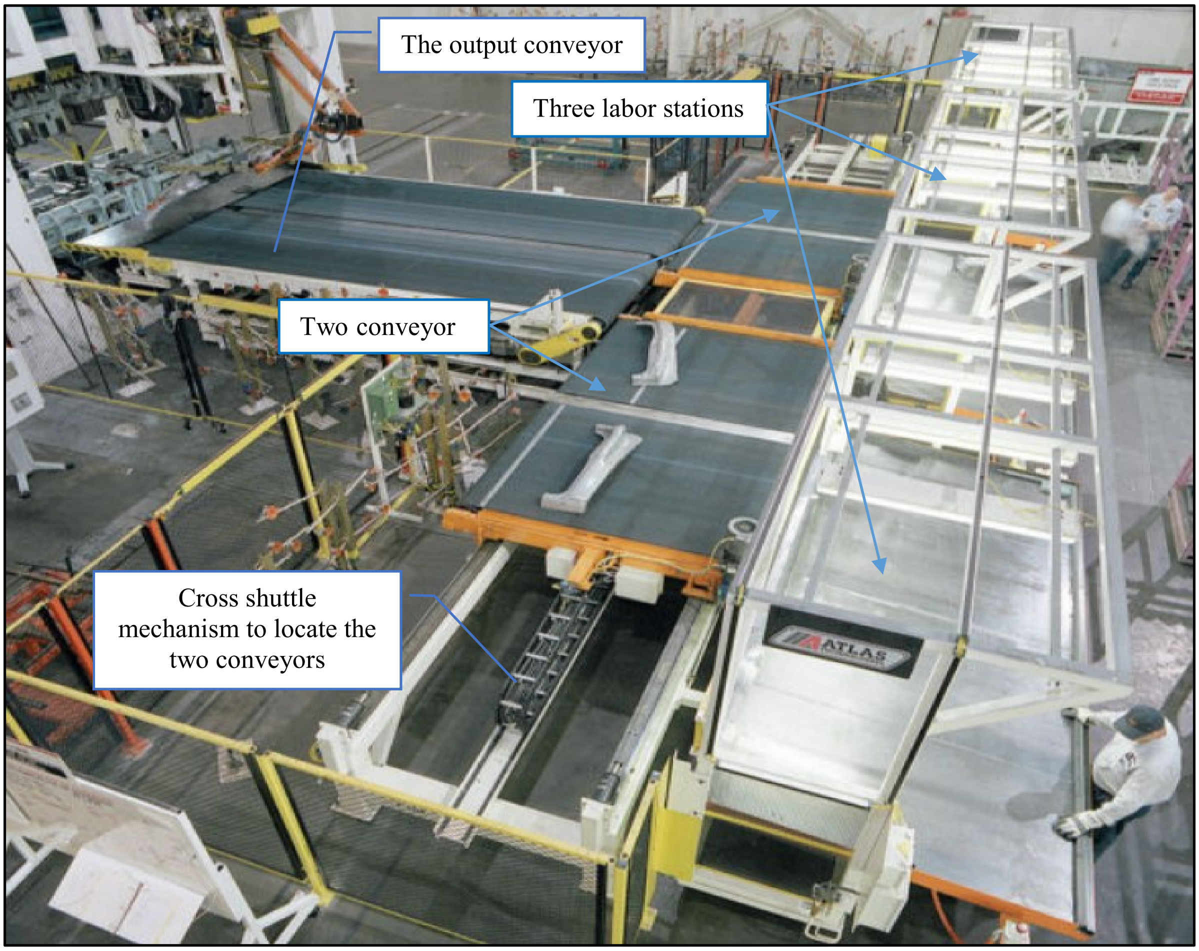


Figure 6:7: The suggested solution from Atlas Company for racking the parts in the end of the press line.

In our case, using the 2nd prototype of the SAMARA robot gives the opportunity to pick and place products up to 5.5 kg in the end of the forming, blanking, or stamping press lines. The 2nd prototype of SAMARA covers a wide range of area without consuming a lot of energy, as clarified before. But an essential point that must be taken into consideration is the ability of the used gripper to grip the product, e.g., the suction cups cannot grip products that do not have a flat surface, and the solution of this problem is to replace the gripper by another gripper which is more adequate.

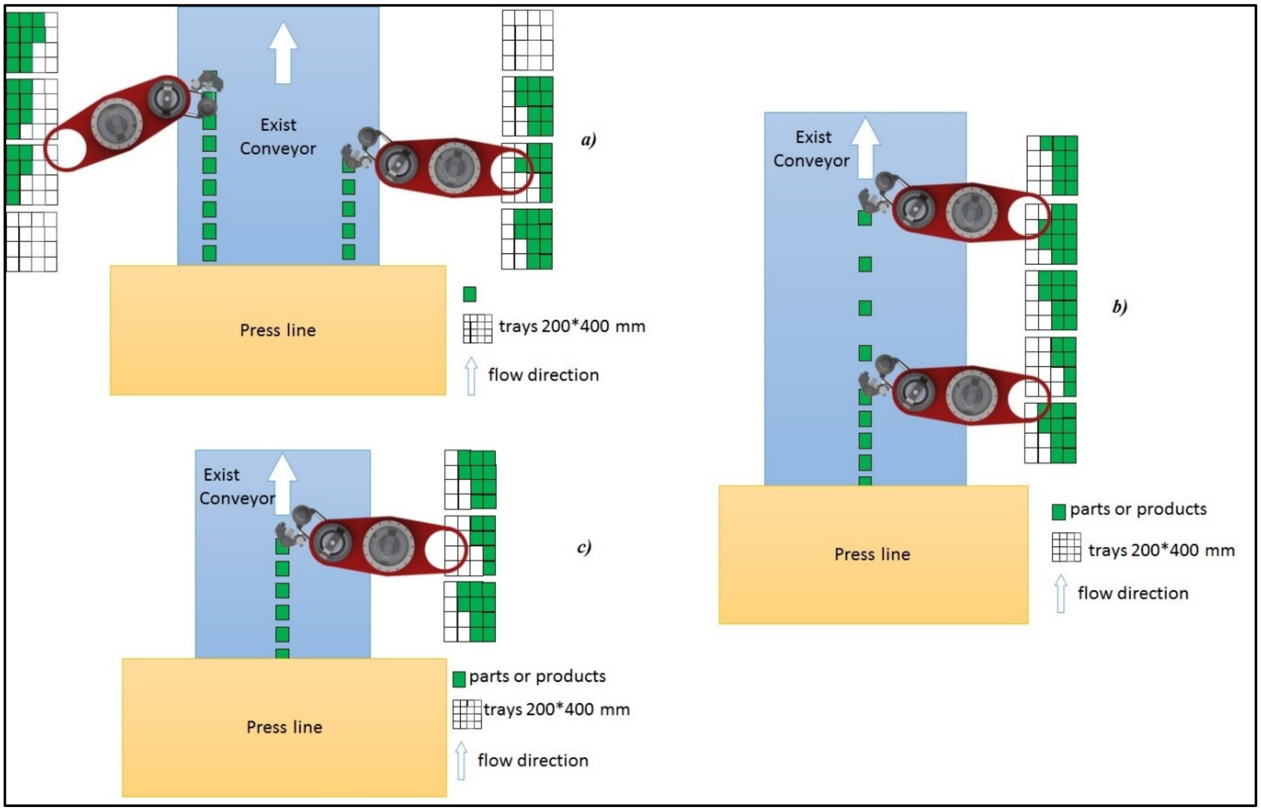


Figure 6:8: Layouts of SAMARA in the end of the press line a) two SAMARA robots are sorting the products into the two sided trays b) two SAMARA are sorting the products into one-sided trays c) single SAMARA is sorting the products.

There are small products or parts which the 2nd prototype of SAMARA can transfer in the end of the press line and these products or parts are used in various fields.

According to [Web-16], kitchen cookware is produced from several materials, e.g., cast iron, aluminum, and stainless steel. Two operations can be used to produce kitchen cookware, which are sheet metal forming or spinning. Here, the focus is on producing kitchen cookware by using the sheet metal forming operation and the selected thickness of these products is usually between 10-22 gauges [Web-16]. Therefore, let us assume the cycle time for the robot is 2 s. The time for each NSM phase is 200 ms and the time for each UAM phase is 0.8 s. Also let us assume there is a sheet metal “A” which is used to form a stock pot, a kitchen pot, or a fry pan, and the selected thickness of this sheet is 2.58826 mm (10 gauge); the selected material is aluminum, so the density of the aluminum is 2700 kg/m^3. Therefore, the area of the sheet metal is 0.28619 m^2 if the mass of the sheet metal is 2 kg. Based on these assumptions, the length of the sheet metal "A" if it is squared is 53.497 cm. While the inertia for this sheet is 9.53976*10^{-2} kg.m^2. It is necessary to mention the ball spline mechanism uses two motors to compensate the orientation error [THK-16]. While it needs one motor to perform the upward motion or the downward motion [THK-16]. Based on that, it is necessary to answer: can motor 4 and motor 5 in the SAS compensate the error of the orientation for the sheet metal "A" occurred during the motion of the robot? This is a critical question otherwise the robot cannot compensate the error of the orientation. To answer this question it is necessary to mention motor 4 and motor 5 are identical motors and both of them must rotate at the same speed and at the same direction to perform a rotational motion [THK-16]. Also, it is worth to know the required parameters for the motors in the SAS, and the required parameters for ball screw mechanism see Table 6:1.

Table 6:1: parameters for the motors in the SAS [Bec-16c], gear ratio for the timing belt, and the ball screw mechanism [THK-16].

Parameters	symbol	value	Parameters	symbol	value
standstill torque	T_{ss}	0.50 Nm	peak torque	T_p	2 Nm
rated torque	T_{rt}	0.50 Nm	the number of teeth on the input gear (motor side)	N_m	72
rated speed	V_{rs}	3000 min^{-1}	the number of teeth on the output gear (ball screw mechanism side)	N_b	24

As mentioned in 2.4.3, the PTP synchronized trajectory planning algorithm is used to compensate the error into the orientation, therefore q_5 in Eq. (2:1) is used to compensate the error into the orientation. The PTP synchronized trajectory planning algorithm usually selects the shortest path to compensate the error also to reduce energy consumption. As a result, the angular positions (q_{SAS}), the angular velocities ($\dot{q}_{SAS}$), and the angular accelerations ($\ddot{q}_{SAS}$) for the SAS have been computed into the payload side. After that, these trajectories are mapped to the motor side by using the gear ratio (G) for the timing belt. In addition, there is a possibility to neglect the inertia of the motors and the inertia of the components of the ball screw mechanism because they are very small in comparison to the inertia of the payload. Based on these assumptions, the required torque to compensate the error of the orientation of the payload on the payload side can be calculated by using Eq. (6:1).

$$T_{req} = J_{payload} * \ddot{q}_{SAS} \qquad (6:1)$$

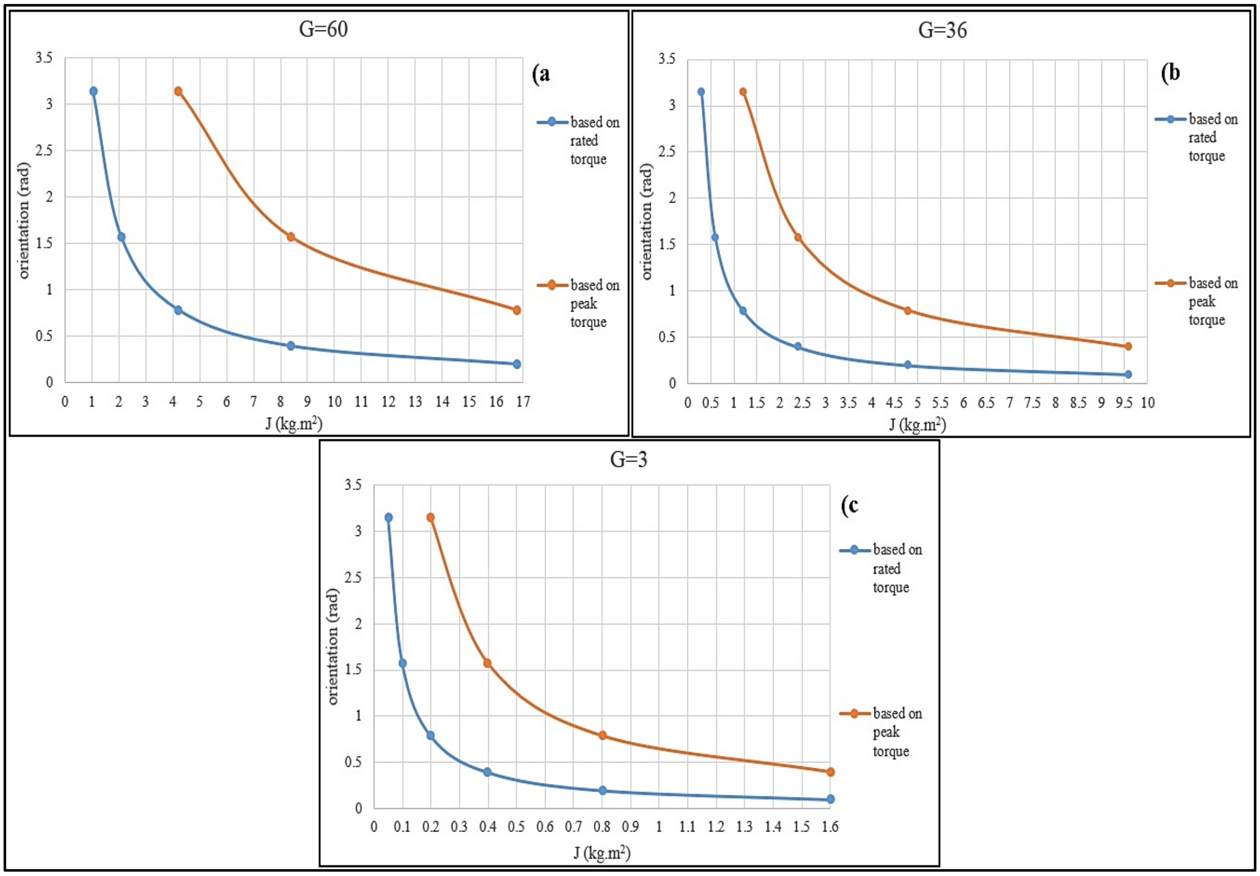


Figure 6:9: Relationship between the orientation correction for the payload and the inertia of the payload when a) $G = 60$, b) $G = 36$, and c) $G = 3$.

To check if motor 4 and motor 5 can compensate the error of the orientation for the sheet metal “A” based on the T_{rt} and T_p limits when $G = 3$, the concept focuses on the simulation of the ability of the end effector to compensate the orientation based on the maximum limits for T_{rt} and T_p respectively as shown in Figure 6:9. As a result, the end effector can compensate the error only by exceeding the limit for T_{rt}. However, this is not a suitable solution consequently there are two solutions have been suggested to solve this problem. The first solution is to replace the existing gears to increase the gear ratio or to integrate a planetary gear e.g. AG2250 planetary gear unit from Beckhoff Company with transmission ratio 12, or 20 [Bec-16b]. This means the gear ratio between each motor and the ball screw mechanism will be 36, or 60 respectively. The effects of these modifications are shown in Figure 6:9. Consequently, selecting $G = 36$ is better solution because the robot will be capable to provide suitable torque below the limit of T_{rt} . But, if the robot would use to handle bigger sheet metal e.g. a sheet metal “B” which is formed to create e.g. a fender for the light car or a reinforcement sheet metal as shown in Figure 6:10. Where the selected thickness of sheet metal “B” is 1.45034 mm (15 gauge); and the selected material is aluminum. Therefore, the maximum area of the sheet metal that the robot can handle it is 1.4045 m^2 and the dimension of the sheet metal “B” if it is squared is 118.51 cm. Therefore, the inertia for this sheet is 1.287 kg.m^2. Therefore, it is recommended to use $G = 60$ because this gear ratio can cover wider range for inertias, see Figure 6:9. However in other cases, the required thickness needed may be more or less depending on the design criteria, or it might need to use another material based on other criteria, e.g., heat conductor, and cost. The second solution to the orientation problem is to replace the existing motors by stronger motors.

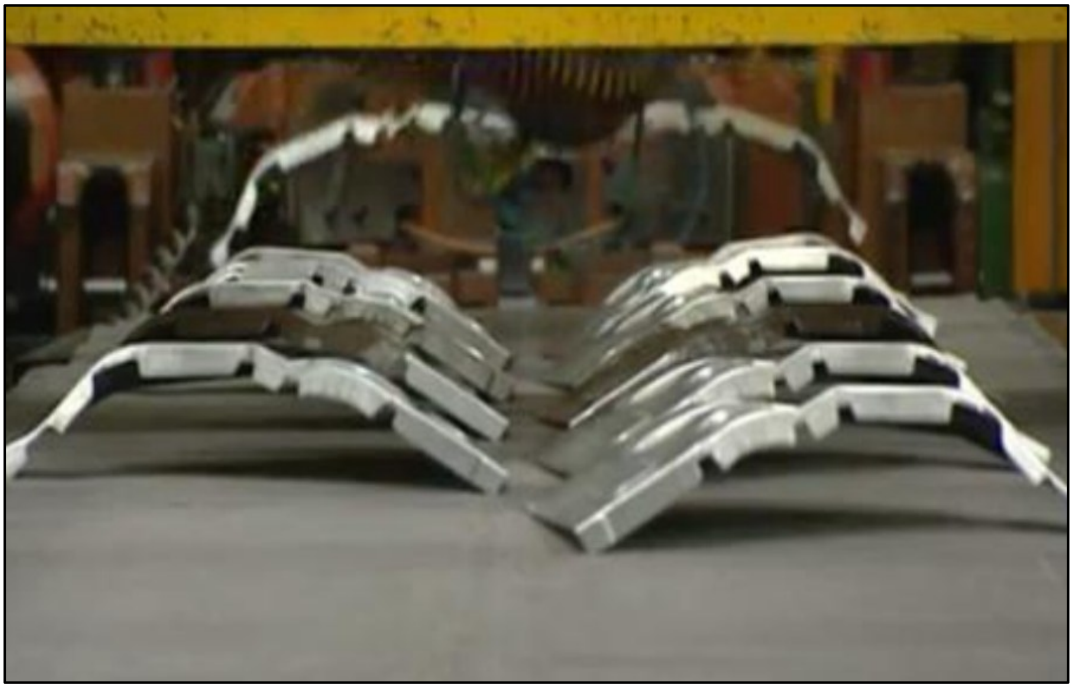


Figure 6:10: Reinforcement sheet metal used in a car.

The robot can also handle some of the other parts that are used in cars, e.g., the hubcaps or the bumper for light cars, because in some cases, the mass of the bumper is less than 5.5 kg [Gra-16]. Cars have 40 to 50 major components formed from the sheet metals, and the robot can transfer some of these components [Roc-13]. There is also a new concern to reduce the weight of the cars by replacing the heavy components by lighter components to reduce the cost of production, cars, and fuel consumption. Consequently, it is expected that the number of the parts or components which the robot can transfer will increase as well.

7 Summary, Outlook, and Recommendations

7.1 Summary and Outlook

This dissertation presents the advantages of using the kinematics of the under-actuated robot as an innovative solution to reduce energy consumption and cost. The kinematics of the robot used the under-actuated and redundancy configurations at the same time in the UAM phase of motion to reduce energy consumption. However, using this kinematics increases the challenge dramatically, because controlling the robot with a passive axis decreases the control space. Also, there is a high coupling for forces and inertias in such a system and this coupling is used to control the passive axis in an indirect way.

Each cycle for SAMARA has four main phases of motion, which they are two NSM and two UAM. The goal of using the NSM is to allow the end effector to pick or place the payloads in or into the processing points respectively, while the goal of using the UAM is to move between the processing points and to consume less energy, because it allows the robot to implement the idea of effective use of the dynamics of the robot. This is achieved because the kinematic configurations for the robot in the UAM phase allows using the centrifugal and the Coriolis forces to reduce the required torque in each axis. In addition, the trajectory planning algorithm for the UAM computes the angular position, the angular velocity, and the angular acceleration trajectories for each axis in the MAS as numerical solutions for the optimization problem. The aim of using the optimization problem is to minimize energy consumption during the UAM in a specified final time. This approach can satisfy the goal of increasing energy saving only; to achieve the goal of decreasing the cycle time while also solving the optimization problem in multiple attempts, in each attempt, the final time is decreased frequently to achieve the required value for the final time if possible. This approach succeeds in increasing the enhancement rate for the cycle time by 48.39% in the simulation bases, but in reality, the enhancement rate for the cycle time was 35.48% only due to the limited capabilities of the actuators. Results-based study summarized that the trajectory planning algorithm minimizes energy when working in short cycle time as compared to standard ones.

In this work, the QL algorithm is used to solve the optimization problem rather than the EA. The solution of the optimization problem is computed based on calculus of variations theories. As a result, other goals for this research have been achieved using the QL algorithm, e.g., the computation time reduced up to one minute instead of two hours. But it is expected that transferring the code to C or C++ will improve the computation time by 10-100 times. On the other hand, using the QL instead of EA decreases the relative error between the simulation and the experimental results from 200-400% to less than 30% in the majority of cases based on using the position control scheme, while it is expected to reduce this error to less than 1% if the optimal control scheme is used.

Two control schemes have been developed to control the robot which are the position control scheme and the optimal control scheme. The position control scheme has the capability to control the position of each axis of the robot as a top-level priority, but it has the ability to control the velocity of each axis and the current as a minor priority. This scheme uses the calculated position trajectories which are calculated by M1 as reference commands, and the target of the controller is to track these reference commands. The optimal control scheme contains the AMPC and successive linearization process to describe the nonlinear models more adequately. Also, it uses the M1 trajectories as reference commands, but it has the capability to predict the behavior of the robot during the prediction horizon to adapt the performance. In addition, the optimal control scheme has the capability to minimize the backlash error, the inputs, and the changes rate of inputs, taking into considerations the maximum allowable input for each axis, and the maximum allowable change of rate for each input.

Three approaches have been developed to analyze the power consumption and the performance of the 2nd prototype of SAMARA. The first approach focuses on comparing the power consumption for the robot if the M1 or M2 are used as trajectory planning algorithms. The achieved result denotes that there is a reduction in the electrical power consumption by 22.54% in average and an enhancement in the cycle time by 9.1%. The second approach focused on comparing the electrical power consumption for the 2nd prototype of SAMARA robot with the SC1 under the same circumstances. The enchantment rate for the handling rate for using 2nd prototype of SAMARA instead of using SC1 was 42.87%. It is also worth mentioning that, at the highest speed for each robot, the 2nd prototype of SAMARA consumed less than 0.45 kW, while the SC1 consumed 0.77 kW. The third approach is summarized by comparing the mechanical power consumption for the 2nd prototype of SAMARA and the SC2 under the same conditions. The results clarified a possibility to save mechanical power up to 65%.

Based on the previous results, it is recommended that the 2nd prototype of SAMARA be used in two application fields based on its features and the requirements of each industry. These application fields are: the food industry, and the industries using press lines for producing, forming, or blanking small parts, workpieces, and products.

The applicability for using the 2nd prototype of SAMARA has been studied for the food industry based on several perspectives, taking into consideration that this industry is classified as a mass production industry. Four solutions have been presented to utilize the robot for pick and place tasks or sorting tasks, especially when wider and taller conveyors are used. Based on that, the robot would be more than applicable in this application, because it is efficient for distances more than 1 m, consuming less energy and with a cycle time less than other industrial robots. On the other hand, integrating the 2nd prototype of SAMARA in the end of press line has been studied to evaluate the applicability of using the robot to execute the pick and place tasks, or the sorting tasks for the small products into trays. Three solutions have been suggested to integrate the robot to achieve the most reliable performance based on the requirements of each industry.

7.2 Recommendations

Based on the knowledge and experience gained, it is recommended that the following tasks be executed:

- Replace the timing belt which connect the 2nd and the 3rd axes by a chain belt to reduce elasticity in the system.
- Reduce the inertia and the mass of the first link or use a stronger motor in this axis to achieve higher velocities.
- Implement the optimal control scheme and test this scheme based on experiments.
- Focus on designing new end effector to transfer multiple parts and products at the same time.

8 References

8.1 Literatures

[Alb-15] Albalasie, A; Glodde, A; Seliger, G; Abu Hanieh, A. Quasi-linearization Approach for the Under-actuated Robots. In: Procedia CIRP, 26, pp. 223-228, 2015.

[Alb-16] Albalasie, A; Seliger, G; Abu Hanieh, A. Using Adaptive Model Predictive Technique to Control Underactuated Robot and Minimize Energy Consumption. In: Procedia CIRP, 40, pp. 407-412, 2016.

[Ane-03] Aneke, N P. Control of Underactuated Mechanical Systems. PhD Thesis, April / 2003.

[Bec-15] Beckhoff Automation GmbH & Co. TwinCAT_3_Matlab/Simulink. Germany, 2015.

[Bon-12] Bona, B. Kinematics. Politecnico di Torino University. Mobile & Service Robotics, 2012. Torino.

[Bre-13] Brett, T. Effective motion design applied to energy efficient handling. PhD Thesis. Institut für Werkzeugmaschinen und Fabrikbetrieb (IWF), 2013.

[Bre-10] Brett, T; Quiel, H; Seliger, G. Eco-efficient path planning for a redundant underactuated assembly robot, In: 3rd CIRP Conference on Assembly Technologies and Systems, Trondeim. 1-3 June, 2010.

[Bre-08] Brett, T; Seliger, G. New kinematics and motion approach for parts handling, In: 40th CIRP International Seminar on Manufacturing Systems, Liverpool. May 30-June 1, 2007, 2008.

[Bur-89] Burdick, J W. On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. In: WALDRON, K J (Ed.). Advanced Robotics: 1989: Proceedings of the 4th International Conference on Advanced Robotics Columbus, Ohio, June 13-15, 1989. Springer, pp. 25–34, 1989.

[Bus-04] Buss, S R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. In: IEEE Journal of Robotics and Automation, 17(16), pp. 1-19, 2004.

[Cam-07] Camacho, E F; Bordons, C. Model predictive control, second edition ed. Springer. London, New York, 2007.

[Cas-07] Castillos, C; Moreno, W; Valavani, K. Unmanned Helicopter Waypoint Trajectory Tracking Using Model Predictive Control, In: Proceedings of the 15th Mediterranean

Conference on Control and Automation, Athens. July 27-29. IEEE, [Piscataway, N.J.], pp. 1–8, 2007.

[Cha-10] Chatterjee, P; Manikrao Athawale, V; Chakraborty, S. Selection of industrial robots using compromise ranking and outranking methods. In: Robotics and Computer-Integrated Manufacturing, 26(5), pp. 483-489, 2010.

[Che-04] Chemori, A; Loria, A. Walking control strategy for a planar under-actuated biped robot based on optimal reference trajectories and partial feedback linearization, In: Robot Motion and Control, 2004. RoMoCo'04. Proceedings of the Fourth International Workshop on. 17-20 June 2004. IEEE, pp. 61–66, 2004.

[Con-97] Conkur, E . S; Buckingham, R. Clarifying the definition of redundancy as used in robotics. In: Robotica, 15(5), pp. 583-586, 1997.

[Duc-08] Duchaine, V; Bouchard, S; Gosselin, C. Motion Control of a Robot Manipulator in Free Space Based on Model Predictive Control. In: Ceccarelli, M (Ed.). Robot manipulators. In-Teh, Croatia, pp. 137–154, 2008.

[Edg-09] Edgar, T F; Hahn, J. Process Automation. In: Nof, S Y (Ed.). Springer Handbook of Automation, First Edition ed. Springer-Verlag Berlin Heidelberg, pp. 529–543, 2009.

[Fah-07] Fahimi, F. Non-linear model predictive formation control for groups of autonomous surface vessels. In: International Journal of Control, 80(8), pp. 1248-1259, 2007.

[Fia-06] Fiacchini, M; Alvarado, I; Limon, D; Alamo, T; Camacho, E F. Predictive control of a Linear Motor for tracking of constant references, In: Proceedings of the 45th IEEE Conference on Decision & Control, Manchester Grand Hyatt Hotel,San Diego, CA, USA. December 13-15, 2006, pp. 4526–4531, 2006.

[Glo-14] Glodde, A; Afrough, M. Energy Efficiency Evaluation of an Underactuated Robot in Comparison to Traditional Robot Kinematics. In: Procedia CIRP, 23, pp. 127-130, 2014.

[Hol-87] Hollerbach, J M; Suh, K C. Redundancy resolution of manipulators through torque optimization. In: Robotics and Automation, IEEE Journal of, 3(4), pp. 308-316, 1987.

[Hua-06] Huang, P; Jie, Y; Yangsheng, X; Wenfu, X; Bin, L. Genetic Algorithms-Based Minimum Torque Path Planning for Space Manipulator, In: Sixth World Congress on Intelligent Control and Automation, Dalian China. June 21-23. IEEE, [Piscataway, N.J.], pp. 3575–3579, 2006.

[Inc-07] Incerti, G. Trajectory tracking for SCARA robots with compliant transmissions: a technique to improve the positioning precision. In: Proceedings of the 12th IFToMM World Congress, 2007.

[Jaz-07] Jazar, R N. Theory of applied robotics − Kinematics, dynamics, and control. Springer. New York, 2007.

[Kal-16] Kaliński, K J; Mazur, M. Optimal control of 2-wheeled mobile robot at energy performance index. In: Mechanical Systems and Signal Processing, 70-71, pp. 373-386, 2016.

[Kir-04] Kirk, D E. Optimal control theory − An introduction. Dover Publications. Mineola, N.Y., 2004.

[LI-90] LI, F; BAINUM, P M. Numerical approach for solving rigid spacecraft minimum time attitude maneuvers. In: Journal of Guidance, Control, and Dynamics, 13(1), pp. 38-45, 1990.

[Luc-95] Luca, A D de; Oriolo, G. Modelling and control of nonholonomic mechanical systems. In: Angeles, J; Kecskeméthy, A (Eds.). Kinematics and dynamics of multi-body systems. Springer-Verlag, Wien, New York, 1995.

[Mar-09] Marcos, M d G; Tenreiro Machado, J A; Azevedo-Perdicoúlis, T P. Trajectory planning of redundant manipulators using genetic algorithms. In: Communications in Nonlinear Science and Numerical Simulation, 14(7), pp. 2858-2869, 2009.

[Mar-08] Marton, L; Hodel, A S; Lantos, B; Hung, J Y. Underactuated Robot Control: Comparing LQR, Subspace Stabilization, and Combined Error Metric Approaches. In: IEEE Transactions on Industrial Electronics, 55(10), pp. 3724-3730, 2008.

[May-14] Mayne, D Q. Model predictive control: − Recent developments and future promise. In: Automatica, 50(12), pp. 2967-2986, 2014.

[Moh-12] Mohan, S; Kim, J; Kim, Y. A Null Space Control of an UnderactuatedUnderwater Vehicle-Manipulator System under Ocean Currents, In: 2012-Yeosu OCEANS. May 21-24. IEEE, Yeosu, pp. 1–5, 2012.

[Nof-09] Nof, S Y. Automation: What It Means to Us Around the World. In: Nof, S Y (Ed.). Springer Handbook of Automation, First Edition ed. Springer-Verlag Berlin Heidelberg, pp. 13–52, 2009.

[Oh-10] Oh, S-R; Sun, J. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. In: Ocean Engineering, 37(2), pp. 289-295, 2010.

[Olf-01] Olfati-Saber, R. Nonlinear Control of Underactuated MechanicalSystems with Application to Robotics andAerospace Vehicles. PhD Thesis. Cambridge, Massachusetts, USA, Feb. / 2001.

[Pae-14] Paes, K; Dewulf, W; Elst, K V; Kellens, K; Slaets, P. Energy Efficient Trajectories for an Industrial ABB Robot. In: Procedia CIRP, 15, pp. 105-110, 2014.

[Pas-07] Passold, F. Tutorial sobre criação de toolboxes no MATLAB/Simulink. Curso de Engenharia Elétrica Grupo de Robótica Móvel, 2007.

[Ray-08] Ray, S. Introduction to materials handling. New Age International (P) Ltd., Publishers. New Delhi, 2008.

[Sch-98] Schuler GmbH. Metal forming handbook. Springer Science & Business Media, 1998.

[Sci-00] Sciavicco, L; Siciliano, B. Modelling and Control of Robot Manipulators, Second edition ed. Springer London. London, 2000.

[Shi-08] Shibata, T; Murakami, T. Motion Behavior of Null Space in Redundant Robotic Manipulators. In: Ceccarelli, M (Ed.). Robot manipulators. In-Teh, Croatia, pp. 413–424, 2008.

[Sid-08] Sidek, S N. Dynamic modeling and control of nonholonomic wheeled mobile robot subjected to wheel slip. PhD Thesis. Vanderbilt University, 2008.

[Spo-94] Spong, M W. Partial feedback linearization of underactuated mechanical systems, In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), Munich, Germany. 12-16 Sept. 1994. IEEE, pp. 314–321, 1994.

[Ted-09] Tedrake, R. Underactuated Robotics: − Learning, Planning, and Control for Efficient and Agile Machines. Massachusetts Institute of Technology, 2009. USA.

[Tok-14] Tokekar, P; Karnad, N; Isler, V. Energy-optimal trajectory planning for car-like robots. In: Autonomous Robots, 37(3), pp. 279-300, 2014.

[Val-99] Valerie, B; Brian, D. Automative and Transportation Applications. In: Nof, S Y (Ed.). Handbook of industrial robotics, 2nd ed. ed. John Wiley, New York, pp. 1069–1080, 1999.

[Wil-10] Wilson, M. Developments in robot applications for food manufacturing. In: Industrial Robot: An International Journal, 37(6), pp. 498-502, 2010.

[Bat-99] Battistella, S; De Pieri, E R; Stemmer, M R. Controle de Posiç ao de um Robô Manipulador SCARA utilizando Redes Neurais Artificiais com Treinamento On-Line, In:. Neural Networks - IV Congresso Brasileiro de Redes Neurais, São José dos Campos - SP - Brazil. July 20-22, pp. 338–343, 1999.

8.2 Internet links

[ABB-13] https://library.e.abb.com/public/73be9c1fdd45bba5c1257c21004696e7/ROB0082EN_G.pdf. Accessed 31 December 2015.

[Atl-16] http://www.atlastechnologies.com/wp-content/themes/atlastechnologies/docs/EndofLineSystems.pdf. Accessed 2 May 2016.

[Aut-16] http://www.autoform.com/en/glossary/sheet-metal-forming/. Accessed 3 May 2016.

[Bec-16a] http://www.beckhoff.de/twincat3/. Accessed 25 April 2016.

[Bec-16b] https://www.beckhoff.com/english.asp?drive_technology/planetary_gears_agxxxx.htm?id=347119897159. Accessed 24 June 2016.

[Bec-16c] http://www.beckhoff.de/default.asp?drive_technology/am8121.htm. Accessed 24 June 2016.

[Bec-16d] http://www.beckhoff.de/default.asp?ethercat/el3403.htm. Accessed 30 July 2016.

[Bec-16e] http://www.beckhoff.de/default.asp?ethercat/el3433.htm. Accessed 31 July 2016.

[Cac-16] http://cache1.asset-cache.net/xd/481863008.jpg?v=1&c=IWSAsset&k=2&d=288666E206AA370DCB495F7B57180B60DD9BE2D9A1DB104CE55667B3A105AEF0FC45D783B2D9F8E7. Accessed 28 April 2016.

[Com-16] https://www.youtube.com/watch?v=vCh0QYWpj6g. Accessed 18 May 2016.

[Den-16] http://www.denso.com.au/var/denso/storage/images/products/aftermarket_industrial_products/robotics/product_range_overview/4_axis/cs/2101-3-eng-AU/cs_large.jpg. Accessed 19 April 2016.

[ADE-16] http://img.directindustry.com/images_di/photo-g/6076-2692867.jpg. Accessed 19 April 2016.

[ISO-12] https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en. 2012. Accessed 8 January 2016.

[Erm-12] http://www.ermetal.com/eng/index-3.php?kid=448#!prettyPhoto. Accessed 3 May 2016.

[Foo-13] http://www.foodmanufacture.co.uk/var/plain_site/storage/images/publications/food-beverage-nutrition/foodmanufacture.co.uk/business-news/suspects-line-up-to-buy-burton-s-biscuits/8377415-1-eng-GB/Suspects-line-up-to-buy-Burton-s-Biscuits.jpg. Accessed 28 April 2016.

[Gra-16] http://www.grantadesign.com/education/InDepth/html/indepth/processinfo/processcasestudy_bumper.htm. Accessed 8 May 2016.

[IAI-16] http://www.pi4.de/fileadmin/material/datenblatt/IX-CARA_V2b_0710_deu.pdf. Accessed 13 April 2016.

[Kaw-16] https://robotics.kawasaki.com/userAssets1/productPDF/YF003N.pdf. Accessed 13 April 2016.

[Kuk-16a] http://www.b4bschwaben.de/cms_media/module_img/212/106081_1_lightbox_KUKA_QUANTEC.jpg. Accessed 19 April 2016.

[Kuk-16b] http://www.kuka-robotics.com/res/sps/9cb8e311-bfd7-44b4-b0ea-b25ca883f6d3_KUKA_PB_KLEINROBOTER_en.pdf. Accessed 4/202016.

[Mat-14] http://de.mathworks.com/help/mpc/ug/adaptive-mpc.html. Accessed 24 February 2015.

[Mat-16a] http://de.mathworks.com/help/mpc/ug/generate-code-and-deploy-controller-to-real-time-targets.html#bu742v8. Accessed 6 April 2016.

[Mat-16b] http://de.mathworks.com/help/mpc/ug/specifying-constraints.html#buj077i. Accessed 14 March 2016.

[Mat-16c] http://de.mathworks.com/products/sl-plc-coder/. Accessed 7 April 2016.

[MBS-16] http://mbs-ag.com.ua/sites/default/files/imagecache/product_1_image_thumb/wsk_60.jpg. Accessed 9 August 2016.

[Mit-16] http://i1-linux.softpedia-static.com/screenshots/SCARA-robot_1.jpg. Accessed 19 April 2016.

[Pbc-16] http://www.pbclinear.com/Images/News/Cartesion-Robotic-System.jpg. Accessed 19 April 2016.

[Pho-16] http://s1277.photobucket.com/user/aquamarin4321/media/aquamarin4321028/IMG_9580_zps805c7f2b.jpg.html. Accessed 9 August 2016.

[Qib-16] http://qibangrobot.com/Public/kindeditor/attached/image/20160224/20160224125100_12394.jpg. Accessed 28 April 2016.

[Roc-13] http://literature.rockwellautomation.com/idc/groups/literature/documents/wp/oem-wp012_-en-p.pdf. Accessed 4 May 2016.

[Sch-16a] https://www.schulergroup.com/technologien/produkte/presswerkplanung_leistungsdaten/index.html?catId=45556429. Accessed 6 May 2016.

[Sch-16b] http://de.schunk.com/de_en/gripping-systems/#/product/28088-0310940-egp-40-n-n-b. Accessed 16 June 1985.

[Sic-16] https://www.sick.com/de/de/motor-feedback-systeme/motor-feedback-systeme-rotativ-hiperface-dsl/eksekm36/eks36-0kf0a020a/p/p231944. Accessed 30 July 2016.

[Sna-16] http://www.snackandbakery.com/ext/resources/Issues/2015/August/automation/0815SFWB_WAREHOUSING7-slideshow.jpg. Accessed 28 April 2016.

[Stä-16] http://www.staubli.com/index.php?eID=tx_nawsecuredl&u=0&g=0&t=1463660531&hash=9427d8b27ce80587df360c75f6b3ebecd248e7ce&file=fileadmin/user_upload/staublicom/robotics/PDF-files/TP80_Robot_EN_022013.pdf. Accessed 3 May 2016.

[Web-16] http://www.webstaurantstore.com/guide/501/types-of-cookware.html. Accessed 8 May 2016.

[WSK-16] http://www.mbs-ag.com/en/pages/Produktprogramm/index.php?navid=1&IdTreeGroup=9&IdProduct=49. Accessed 30 July 2016.

[THK-16] https://tech.thk.com/upload/catalog_claim/pdf/327E_BNS_NS.pdf. Accessed 24 June 2016.

9 Appendix

9.1 Direct kinematics

As described in 2.2 the D-H table calculated in Table 9:1.

Table 9:1: D-H table for SAMARA

Joint index	$\boldsymbol{\theta_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{a_i}$	$\boldsymbol{d_i}$
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	l_3	$-d_3$
4	d_4	0	0	$-d_4$
5	θ_5	0	0	$-d_5$

The homogeneous transformation matrices calculated below based on Figure 2:3

$$T_1^0 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & l_1 * \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & l_1 * \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2 * \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2 * \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & l_3 * \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & l_3 * \sin(q_3) \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} \cos(q_5) & -\sin(q_5) & 0 & 0 \\ \sin(q_5) & \cos(q_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^5 = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5$$

$$T_0^5 = \begin{bmatrix} \cos(q_{1235}) & -\sin(q_{1235}) & 0 & l_1 * \cos(q_1) + l_2 * \cos(q_{12}) + l_3 * \cos(q_{123}) \\ \sin(q_{1235}) & \cos(q_{1235}) & 0 & l_1 * \sin(q_1) + l_2 * \sin(q_{12}) + l_3 * \sin(q_{123}) \\ 0 & 1 & 1 & -d_2 - d_3 - d_4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Finally the orientation and the position for the end effector is shown in Eq. (2:1).

9.2 Dynamics of under-actuated robot

First model

Clarification for Eq. (2:11) where the robot is not gripping a payload is shown below:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

$$\begin{aligned} m_{11} = {} & I_{c1} + I_{c2} + I_{c3} + l_1^2 m_2 + l_1^2 m_3 + l_2^2 m_3 + l_{c1}^2 m_1 + l_{c2}^2 m_2 + l_{c3}^2 m_3 \\ & + 2m_3 l_1 l_{c3} \, cos(q_2 + q_3) + \quad 2l_1 l_2 m_3 \, cos(q_2) + 2l_1 l_{c2} m_2 \, cos(q_2) \\ & + 2l_2 l_{c3} m_3 cos(q_3) \end{aligned}$$

$$m_{13} = m_{31} = I_{c3} + l_{c3}^2 m_3 + l_1 l_{c3} m_3 \, cos(q_2 + q_3) + l_2 l_{c3} m_3 \, cos(q_3)$$

$$m_{22} = I_{c3} + I_{c2} + l_{c3}^2 m_3 + l_2^2 m_3 + m_2 l_{c2}^2 + 2l_2 l_{c3} m_3 cos(q_3)$$

$$m_{23} = m_{32} = I_{c3} + l_{c3}^2 m_3 + l_2 l_{c3} m_3 cos(q_3)$$

$$m_{33} = I_{c3} + l_{c3}^2 m_3$$

$$C(q, \dot{q}) = \begin{bmatrix} C_1(q, \dot{q}) \\ C_2(q, \dot{q}) \\ C_3(q, \dot{q}) \end{bmatrix}$$

$$\begin{aligned} C_1(q, \dot{q}) = {} & -\dot{q}_2^2 l_1 l_{c3} m_3 \, sin(q_2 + q_3) - \dot{q}_3^2 l_1 l_{c3} m_3 \, sin(q_2 + q_3) - \dot{q}_2^2 l_1 l_2 m_3 \, sin(q_2) \\ & - \dot{q}_2^2 l_1 l_{c2} m_2 \, sin(q_2) - \dot{q}_3^2 l_2 l_{c3} m_3 \, sin(q_3) - 2\dot{q}_1 \dot{q}_2 l_1 l_{c3} m_3 \, sin(q_2 + q_3) \\ & - 2\dot{q}_1 \dot{q}_3 l_1 l_{c3} m_3 \, sin(q_2 + q_3) - 2\dot{q}_2 \dot{q}_3 l_1 l_{c3} m_3 \, sin(q_2 + q_3) \\ & - 2\dot{q}_1 \dot{q}_2 l_1 l_2 m_3 \, sin(q_2) - 2\dot{q}_1 \dot{q}_2 l_1 l_{c2} m_2 \, sin(q_2) - 2\dot{q}_1 \dot{q}_3 l_2 l_{c3} m_3 \, sin(q_3) \\ & - 2\dot{q}_2 \dot{q}_3 l_2 l_{c3} m_3 sin(q_3) \end{aligned}$$

$$\begin{aligned} C_2(q, \dot{q}) = {} & \dot{q}_1^2 l_1 l_{c3} m_3 \, sin(q_2 + q_3) + \dot{q}_1^2 l_1 l_2 m_3 \, sin(q_2) + \dot{q}_1^2 l_1 l_{c2} m_2 \, sin(q_2) \\ & - \dot{q}_3^2 l_2 l_{c3} m_3 \, sin(q_3) - 2\dot{q}_1 \dot{q}_3 l_2 l_{c3} m_3 \, sin(q_3) - 2\dot{q}_2 \dot{q}_3 l_2 l_{c3} m_3 sin(q_3) \end{aligned}$$

$$C_3(q, \dot{q}) = m_3 l_{c3} \, l_2 sin(q_3) \, (\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + m_3 l_1 l_{c3} \dot{q}_1^2 sin(q_2 + q_3)$$

$$D(q,\dot{q}) = \begin{bmatrix} D_1(q,\dot{q}) \\ D_2(q,\dot{q}) \\ D_3(q,\dot{q}) \end{bmatrix}$$

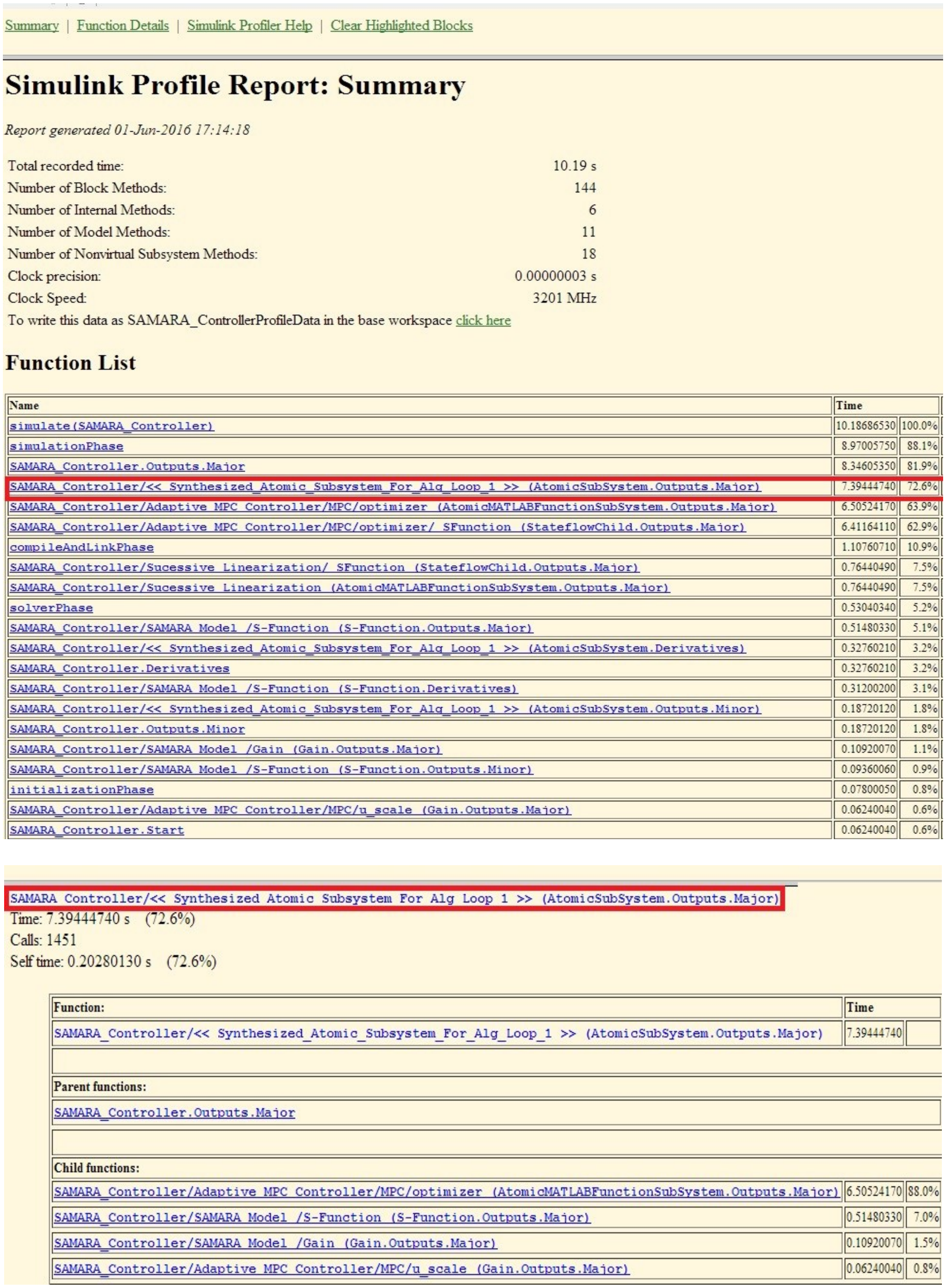
$$D_1(\dot{q}) = k_{d1}\dot{q}_1 + \frac{2k_{r1}\, tan^{-1}(k_w\dot{q}_1)}{\pi}$$

$$D_2(\dot{q}) = k_{d2}\dot{q}_2 + \frac{2k_{r2}\, tan^{-1}(k_w\dot{q}_2)}{\pi}$$

$$D_3(\dot{q}) = k_{d3}\dot{q}_3 + \frac{2k_{r3}\, tan^{-1}(k_w\dot{q}_3)}{\pi}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ 0 \end{bmatrix}$$

9.3 Profile report



Summary | Function Details | Simulink Profiler Help | Clear Highlighted Blocks

Simulink Profile Report: Summary

Report generated 01-Jun-2016 17:14:18

Total recorded time:	10.19 s
Number of Block Methods:	144
Number of Internal Methods:	6
Number of Model Methods:	11
Number of Nonvirtual Subsystem Methods:	18
Clock precision:	0.00000003 s
Clock Speed:	3201 MHz

To write this data as SAMARA_ControllerProfileData in the base workspace click here

Function List

Name	Time	
simulate(SAMARA_Controller)	10.18686530	100.0%
simulationPhase	8.97005750	88.1%
SAMARA_Controller.Outputs.Major	8.34605350	81.9%
SAMARA_Controller/<< Synthesized_Atomic_Subsystem_For_Alg_Loop_1 >> (AtomicSubSystem.Outputs.Major)	7.39444740	72.6%
SAMARA_Controller/Adaptive MPC Controller/MPC/optimizer (AtomicMATLABFunctionSubSystem.Outputs.Major)	6.50524170	63.9%
SAMARA_Controller/Adaptive MPC Controller/MPC/optimizer/ SFunction (StateflowChild.Outputs.Major)	6.41164110	62.9%
compileAndLinkPhase	1.10760710	10.9%
SAMARA_Controller/Sucessive Linearization/ SFunction (StateflowChild.Outputs.Major)	0.76440490	7.5%
SAMARA_Controller/Sucessive Linearization (AtomicMATLABFunctionSubSystem.Outputs.Major)	0.76440490	7.5%
solverPhase	0.53040340	5.2%
SAMARA_Controller/SAMARA Model /S-Function (S-Function.Outputs.Major)	0.51480330	5.1%
SAMARA_Controller/<< Synthesized_Atomic_Subsystem_For_Alg_Loop_1 >> (AtomicSubSystem.Derivatives)	0.32760210	3.2%
SAMARA_Controller.Derivatives	0.32760210	3.2%
SAMARA_Controller/SAMARA Model /S-Function (S-Function.Derivatives)	0.31200200	3.1%
SAMARA_Controller/<< Synthesized_Atomic_Subsystem_For_Alg_Loop_1 >> (AtomicSubSystem.Outputs.Minor)	0.18720120	1.8%
SAMARA_Controller.Outputs.Minor	0.18720120	1.8%
SAMARA_Controller/SAMARA Model /Gain (Gain.Outputs.Major)	0.10920070	1.1%
SAMARA_Controller/SAMARA Model /S-Function (S-Function.Outputs.Minor)	0.09360060	0.9%
initializationPhase	0.07800050	0.8%
SAMARA_Controller/Adaptive MPC Controller/MPC/u_scale (Gain.Outputs.Major)	0.06240040	0.6%
SAMARA_Controller.Start	0.06240040	0.6%

SAMARA Controller/<< Synthesized Atomic Subsystem For Alg Loop 1 >> (AtomicSubSystem.Outputs.Major)

Time: 7.39444740 s (72.6%)

Calls: 1451

Self time: 0.20280130 s (72.6%)

Function:	Time	
SAMARA_Controller/<< Synthesized_Atomic_Subsystem_For_Alg_Loop_1 >> (AtomicSubSystem.Outputs.Major)	7.39444740	
Parent functions:		
SAMARA_Controller.Outputs.Major		
Child functions:		
SAMARA_Controller/Adaptive MPC Controller/MPC/optimizer (AtomicMATLABFunctionSubSystem.Outputs.Major)	6.50524170	88.0%
SAMARA_Controller/SAMARA Model /S-Function (S-Function.Outputs.Major)	0.51480330	7.0%
SAMARA_Controller/SAMARA Model /Gain (Gain.Outputs.Major)	0.10920070	1.5%
SAMARA_Controller/Adaptive MPC Controller/MPC/u_scale (Gain.Outputs.Major)	0.06240040	0.8%

Figure 9:1: a part of the profiler report in the case of E5.

9.4 The simulation process for the SC2

x_i, y_i, z_i, w_i The initial coordinates in the x-axis, in the y-axis, and in the z-axis respectively in the task space [m]. While the initial orientation of the end effector is the w_i in [rad].

x_f, y_f, z_f, w_f The final coordinates in the x-axis, in the y-axis, and in the z-axis respectively in the task space [m]. While the final orientation of the end effector is the w_f in [rad].

Table 9:2: The dynamic parameters for the SC2 which are citied from [Inc-07]:

parameters	value	unit	parameters	value	unit
g	9.81	$[m/s^2]$	Acc_{max}	5	$[m/s^2]$
l_1	0.5	[m]	F_a	[0.3 -0.5 0 0]	[N.m]
l_2	0.4	[m]	F_b	[34 34 35 35]	[N.m.s]

l_{c1}	0.25	[m]	K_r	[-100 0 0 0;
				0 -100 0 0;
				0 0 1.69 0;
				0 0 0 16.2]
					[-]

l_{c2}	0.2	[m]	K_p	10^3*[14.7 12.1 90 4]	[-]
d_1	0.678	[m]	K_d	[280 220 600 320]	[-]
m_1	15	[kg]	K_i	[478 1200 9200 2400]	[-]
m_2	12	[kg]	t_a	V_{max}/Acc_{max}	[s]
m_3	7.8	[kg]	i_3	0.0580	$[kg.m^2]$
$m_{payload}$	input parameter	[kg]	i_4	0.0565	$[kg.m^2]$
m_4	$m_{payload}$+7	[kg]	V_{max}	5	$[m/s]$
i_1	0.313	$[kg.m^2]$	t_f	input parameter	[s]
i_2	0.16	$[kg.m^2]$	t_s	0.001	[s]

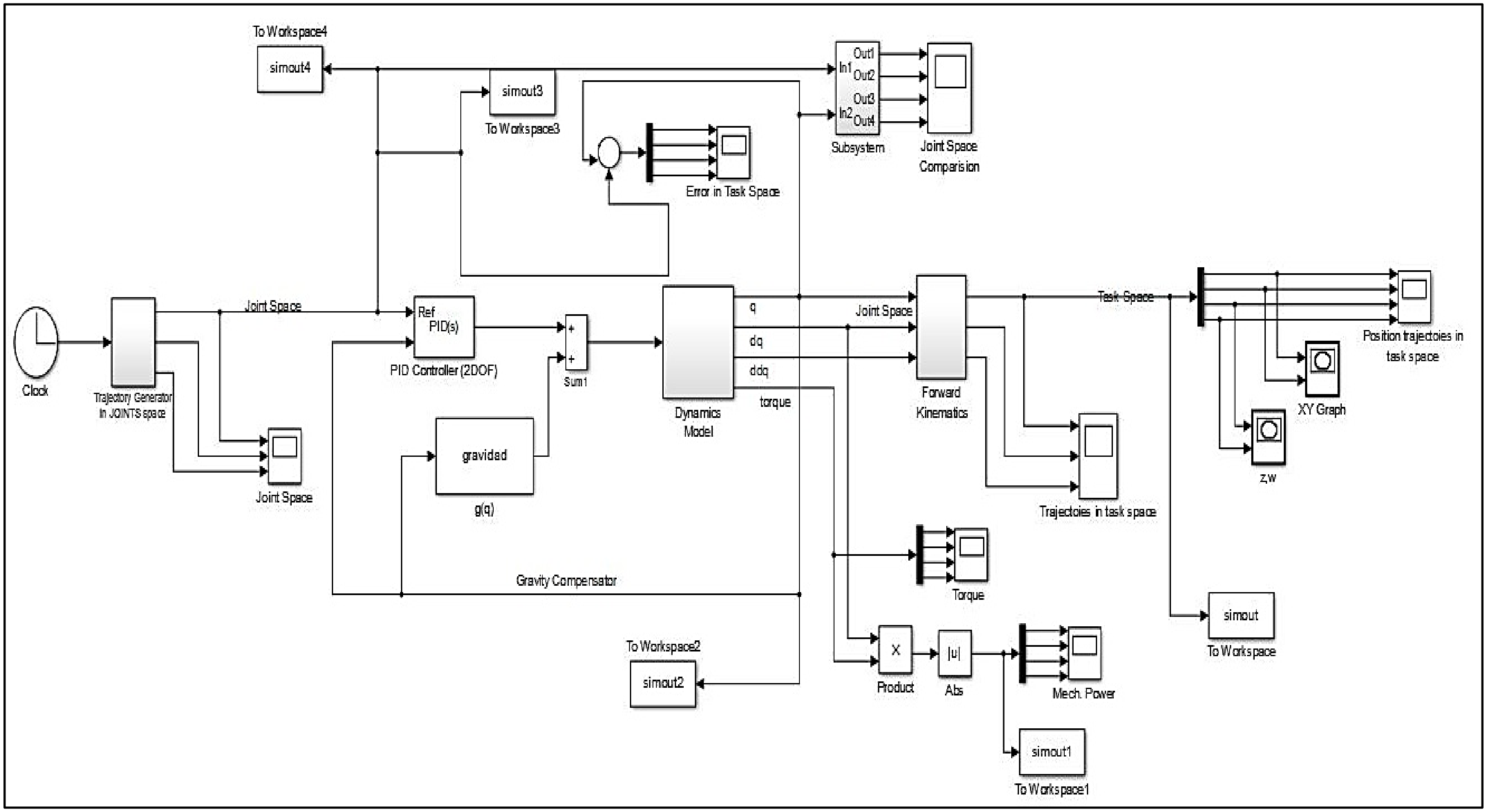


Figure 9:2: The simulation scheme for the SC2.

I) According to [Pas-07], the inverse kinematics calculations:

$$d = \frac{x_i^2 + y_i^2 - l_1^2 - l_2^2}{2l_1 l_2}$$
$$d_f = \frac{x_f^2 + y_f^2 - l_1^2 - l_2^2}{2l_1 l_2}$$
$$q_{2i} = tan^{-1}\left(\frac{\sqrt{1-d^2}}{d}\right)$$
$$q_{1i} = tan^{-1}\left(\frac{y_i}{x_i}\right)\text{-}\ tan^{-1}\left(\frac{l_2 sin(q_{2i})}{l_1+l_2 cos(q_{2i})}\right)$$
$$q_{3i} = d_1 - z_i$$
$$q_{4i} = q_{1i} + q_{2i} - w_i$$
$$q_{2f} = tan^{-1}\left(\frac{\sqrt{1-{d_f}^2}}{d_f}\right)$$
$$q_{1f} = tan^{-1}\left(\frac{y_f}{x_f}\right)\text{-}\ tan^{-1}\left(\frac{l_2 sin(q_{2f})}{l_1+l_2 cos(q_{2f})}\right)$$
$$q_{3f} = d_1 - z_f$$
$$q_{4f} = q_{1f} + q_{2f} - w_f$$

II) According to [Pas-07], the trajectory generator in the joint space based on the inverse kinematics calculations:

The PTP synchronized trapezoidal speed profile is used as a trajectory planning method in the joint space

The process is repeated for each axis for the robot as shown below:

-intervals times
$$t_1 = t_a$$
$$t_2 = t_f - t_a$$
$$t_3 = t_f$$
-initial calculation
$$v_{ap} = \frac{q_{1f} - q_{1i}}{t_f - t_a}$$
$$a_{ap} = \frac{v_{ap}}{t_a}$$
-the acceleration phase ($\leq t_1$)
$$q = q_{1i} + a_{ap}\frac{t^2}{2}$$
$$\dot{q} = a_{ap} t$$
$$\ddot{q} = a_{ap}$$

-the zero acceleration phase ($\leq t_2$)

$$q = q_{1i} + a_{ap}\frac{t_1^2}{2} + v_{ap}(t - t_1)$$
$$\dot{q} = v_{ap}$$
$$\ddot{q} = 0$$

-the deceleration phase ($\leq t_3$)

$$q = q_{1f} - a_{ap}\frac{(t_3 - t)^2}{2}$$
$$\dot{q} = a_{ap}(t_3 - t)$$
$$\ddot{q} = -a_{ap}$$

Else

$$q = q_{1f}$$
$$\dot{q} = 0$$
$$\ddot{q} = 0$$

III) According to [Pas-07] and [Bat-99], The dynamic model for the SCARA robot is shown :

- Compute the $B(q)^{-1} * \mathrm{A}$:-

$$b_0 = m_1 l_{c1}^2 + (m_2 + m_3 + m_4)l_1^2 + m_2 l_{c2}^2 + m_3 + m_4 + i_1 + i_2 + i_3 + i_4 + 2m_2 l_1 l_{c2} cos(q_2) + (m_3 + m_4)2l_1 l_2 cos(q_2)$$

$$b_1 = (m_2 l_{c2} + (m_3 + m_4)l_2)l_1 cos(q_2) + m_2 l_{c2}^2 + (m_3 + m_4)l_2^2 + i_2 + i_3 + i_4$$
$$b_4 = b_1$$
$$b_5 = m_2\, l_{c2}^2 + (m_3 + m_4)l_2^2 + i_2 + i_3 + i_4$$
$$b_{10} = m_3 + m_4$$

$$B = \begin{bmatrix} b_0 & b_1 & 0 & -i_4 \\ b_4 & b_5 & 0 & -i_4 \\ 0 & 0 & b_{10} & 0 \\ -i_4 & -i_4 & 0 & i_4 \end{bmatrix}$$
$$\ddot{q} = B^{-1} * A$$

- Compute $g(q)$:-

$$g_3 = (m_3 + m_4)g$$
$$g_r = \begin{bmatrix} 0 \\ 0 \\ g_3 \\ 0 \end{bmatrix}$$

- Compute the $C * dq$:-

$$K = (m_2 l_{c2} + (m_3 + m_4)l_2)l_1$$

$$C * dq = \begin{bmatrix} -Ksin(q_2)(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ Ksin(q_2)\dot{q}_1^2 \\ 0 \\ 0 \end{bmatrix}$$

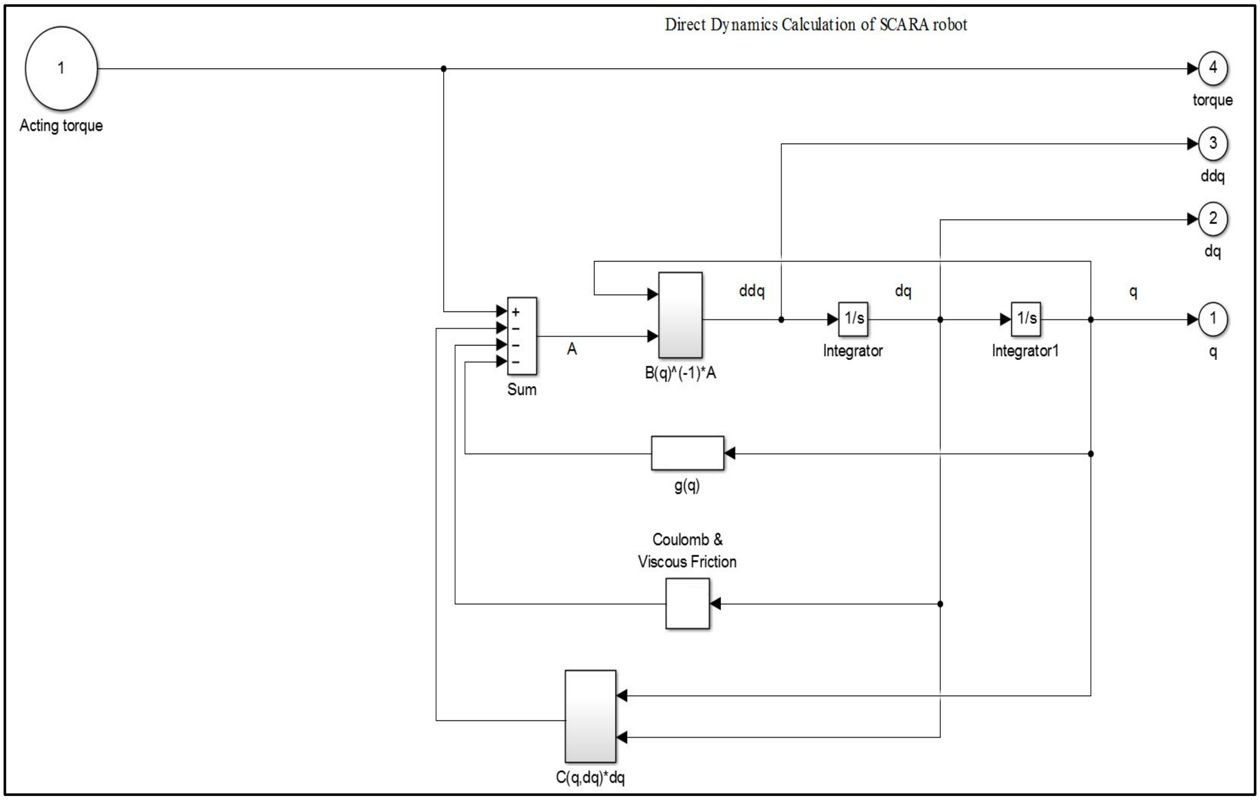


Figure 9:3 : The dynamic model scheme for the SC2 [Pas-07][18].

IV) <u>Forward kinematics to convert the signal from the joint space to task the space:-</u>

Calculate the following terms:

a) desired angular positions in the task space:-

$c_1 = cos(q_1)$

$c_{12} = cos(q_1 + q_2)$

$s_1 = sin(q_1)$

$s_{12} = sin(q_1 + q_2)$

[18] Updated picture.

$x = l_1c_1 + l_2c_{12}$

$y = l_1s_1 + l_2s_{12}$

$z = d_1 - q_3$

$w = q_1 + q_2 - q_4$

b) desired angular velocities in the task space:-

$$J \ = \begin{bmatrix} -l_1s_1 - l_2s_{12} & -l_2s_{12} & 0 & 0 \\ l_1c_1 + l_2c_{12} & l_2c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}$$

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}$$

$\dot{p} = J * \dot{q}$

c) desired angular accelerations in the task space:-

$\dot{q}_{12} = \dot{q}_1 + \dot{q}_2$

$$\dot{J} = \begin{bmatrix} -l_1c_1\dot{q}_1 - l_2c_{12}\dot{q}_{12} & -l_2c_{12}\dot{q}_{12} & 0 & 0 \\ -l_1s_1\dot{q}_1 - l_2s_{12}\dot{q}_{12} & -l_2s_{12}\dot{q}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix}$$

$\ddot{p} = \dot{J}\dot{q} + J\ddot{q}$