People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University 20 August 1955 – Skikda

Ref: D012121002D

Faculty of Technology
Department of Electrical Engineering
Automatic Laboratory of Skikda

Thesis

For the degree of

Doctor of Philosophy (LMD Doctorate)

Domain: Science and Technology
Section: Electrical Engineering
Specialty: Automatic And Industrial Informatics

Presented by

BOUZOUALEGH Samir

Theme

Development Of An Advanced Control Laws For A Mobile Manipulator Robot

Presented publicly on**:** 03/2021

President	Pr. Abderrazek LACHOURI	Full Professor	University of Skikda
Supervisor	Dr. El-Hadi GUECHI	Associate Professor	University of Skikda
Examiners	Pr. Moussa SEDRAOUI	Full Professor	University of Guelma
	Pr. Hadj Ahmed ABBASSI	Full Professor	University of Annaba
	Dr. Lamine MEHENNAOUI	Associate Professor	University of Skikda
	Dr. Fouad INEL	Associate Professor	University of Skikda
Guest	Dr. Karim BELHARET	HDR	University of HEI – France

Year 2021

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 20 Août 1955 - Skikda

Réf : D012121002D

Faculté de Technologie
Département de Génie Electrique
Laboratoire d'Automatique Skikda

THÈSE

En vue de l'obtention du diplôme de

Doctorat LMD

Domaine : **Science et Technologie**

Filière : **Génie Electrique**

Spécialité : **Automatique Et Informatique Industrielle**

Présentée par

BOUZOUALEGH Samir

Thème

Développement Des Lois De Commandes Avancées Pour Un Robot Manipulateur Mobile

Soutenue publiquement le : 03/2021
Devant le jury composé de :

Président	Pr. AbderrazekLACHOURI	Professeur	Université de Skikda
Encadreur	Dr. El-Hadi GUECHI	MCA	Université de Skikda
Examinateurs	Pr. Moussa SEDRAOUI	Professeur	Université de Guelma
	Pr. Hadj Ahmed ABBASSI	Professeur	Université d'Annaba
	Dr. Lamine MEHENNAOUI	MCA	Université de Skikda
	Dr. Fouad INEL	MCA	Université de Skikda
Invites	Dr. Karim BELHARET	HDR	University of HEI – France

Année 2021

Dedicated to

TO my lovely mother and father

To my lovely wife

To my My beloved son

To my brothers and sisters

...

Acknowledgments

This work has been done at the automatic laboratory of Skikda under the supervision of, Dr. Guechi El-Hadi, Professor (Assistant) of University 20 August 1955-Skikda.
First of all, I thank Almighty Allah for always been guiding and helping me during my whole life, and especially to complete this work, and I pray Allah, that this work be useful for future student researchers.
Then, I want to express my honest gratitude and deep appreciation to my supervisor, Dr. Guechi El-Hadi, for directing, supporting and orienting me in my preparation of this work. His availability, feedback, and scientific commitment have greatly supported me throughout my research years. Also, I want to thank Pr. LACHOURI Abderrazek, Full Professor at the University 20 August 1955 of Skikda for the honor, he gave me by accepting to be the head of the jury of my thesis.
I also want to thank Pr. Moussa SEDRAOUI Full Professor at the University of 8 Mai 1945 of Guelma, Pr. Hadj Ahmed ABBASSI Full Professor at the University of Badji Mokhtar of Annaba, Dr. Lamine MEHENAOUI, associate professor at the University 20 August 1955 of Skikda, DR. Fouad INEL , associate professor at the University 20 August 1955 of Skikda and Dr. Karim BELHARET HDR University of HEI – France for having accepted to evaluate this work.
And many thanks to Dr. ZENNIR Youcef, associate professor at Skikda University, for his important advices.
I would as well to thank: Dr. Kelaia Ridha, associate professor of University 20 August 1955-Skikda, for being available and committed to help during these years in my preparation of this work with his advices and experiences and overwhelmed me with his humility.
I would like to acknowledge Dr. Bendib Riad and Dr. Bourahala Fayçal for their valuable advices and there excellent methodology of research. I would also like to express my gratitude to, Dr. Abdelfetah Hentout and Pr. Bouzouia Brahim Research Directors, CDTA, Algiers, for accepting me to visit their Robotic Laboratory and providing a very use full manuals and documents. It is really a pleasure for me to meet them.
I cannot complete my acknowledgements without expressing my respect, my impression and my deep appreciation to Pr. HADDAD Moussa, Ecole Militaire Polytechnique, Algiers.
Further, I want to thank colleagues in the scientific research here at Skikda University who have always been present to discuss ideas with. Special thanks to Ruamel Mohamed, Nafir Noureddine, Sami ALLOU, Abdel Mouneim KHEMISSAT Abderahmene Ganouche, and Bouchabat Rochedi whose support and help have proven to be invaluable.

الملخص -شغل التحكم في الروبوتات المناورة المتحركة اهتمام العديد من الأبحاث في السنوات الأخيرة، نظرًا لاستخدامها المتزايد في البيئات الخطرة أو التي يتعذر الوصول إليها، حيث يصعب على الإنسان التدخل فيها. نماذج هذه الروبوتات غير خطية بشكل كبير، مما يجعل استراتيجية التحكم الخاصة بها صعبة للغاية. هذا ما أجبرنا على اقتراح أوامر متقدمة جديدة للتحكم في هذه الأنواع من الروبوتات، وهذا هو السبب وراء اقتراحنا في هذه الأطروحة طريقة تحكم جديدة تعتمد على نموذج تنبؤي للتحكم في روبوت متحرك ذو أدرع.

منهجية التحكم الجديدة الذي اقترحناه هو مزيج من التحكم الخطي غير الخطي (من أجل تحويل الديناميكيات غير الخطية للنظام إلى ديناميكيات خطية كليًا أو جزئيًا) والتحكم التنبئي القائم على التحكم باستعمال نموذج. (من أجل الحصول على أداء أفضل للنظام). هذه هي المساهمة الرئيسية لهذه الأطروحة.

استراتيجية التحكم في هذا الروبوت هي كما يلي: أولاً، يجب التحكم في القاعدة المتنقلة بحيث تصل إلى موضع بحيث يتناسب الهدف مع مساحة عمل ذراع الروبوت. بعد ذلك، سوف نتحكم في ذراع الروبوت حتى يصل العضو الطرفي إلى هدفه.

في عملنا، للتغلب على مشكلة اللاخطية، استخدمنا التحكم الخطي غير الخطي لذراع الروبوت والتحكم الخطي للمدخلات والمخرجات للقاعدة المتنقلة. بمجرد حصولنا على النماذج الخطية، تم تطوير منهجية التحكم التنبئي القائم على النموذج. حيث قدمنا معيارًا تربيعيًا وتم حساب معاملات التحكم التنبئي ليكون لها سلوك محدد لنظام الحلقة المغلقة من الدرجة الثانية.

يعتمد تصميم منهجية التحكم المقترح على النماذج الديناميكية لهذه الروبوتات (ذراع الروبوت + قاعدة متحركة) حيث تم استخدمناها للتحكم في عدة أنواع من الروبوتات ذوات أذرع كما استخدمناها للتحكم في قاعدة متحركة ذات محرك تفاضلي. أظهرت نتائج المحاكاة التي تم الحصول عليها فعالية منهجية التحكم المقترحة.

الكلمات الرئيسية -التحكم غير الخطي؛ التحكم التنبئي القائم على النموذج؛ روبوت ذو ذراع، قاعدة متحركة ذات محرك تفاضلي

Abstract - In recent years, the control of mobile manipulator robots has been the subject of much research, due to the robots' increasingly frequent use in dangerous or inaccessible environments, where human beings can hardly intervene. These robot models are highly nonlinear which makes the control strategy very difficult. This obliges us to offer new advanced control for controlling this type of robots, hence in this thesis we propose a new control approach based on a model predictive control (MPC) for controlling a manipulator mobile robot. The new proposed approach of control is a combination of a feedback linearization control (in order to transform the nonlinear dynamics of the system to one completely or partially linear) and an MPC control (in order to obtain better performance for the system), which constitutes the main contribution of this thesis. The control strategy of this robot is done in the following way: first we have to control the mobile base, so that it arrives at

a position where the objective fits into the working space of a manipulator arm. In the second step, we must control the manipulator arm so that its end effector achieves its objective.
In our work, to overcome the nonlinearity problem, we have used the feedback linearization control for the manipulator arm and Input-output linearization control for the mobile base. Once, the linear models are obtained, a model predictive control approach is developed. We have introduced a quadratic criterion and these parameters are calculated to have a specific behavior of the closed loop system. The control approach design was based on the dynamic models of the robots and we used it to control several manipulator robots and we also used it to control a differential-drive mobile robot (DDMR). The obtained simulation results show the efficiency of the proposed approach of control.

Keywords - Nonlinear control; Model predictive control (MPC); Manipulator robot; Differential-drive mobile robot (DDMR);

Résumé- La commande des robots manipulateurs mobiles a fait l’objet de nombreuses recherches ces dernières années, en raison de leur utilisation de plus en plus fréquente dans des environnements dangereux ou inaccessibles, où l’être humain peut difficilement intervenir. Les modèles de ces robots sont fortement non-linéaires, ce qui rend leur stratégie de commande très difficile. Cela nous oblige à proposer des nouvelles commandes avancées pour commander ces types de robots, c'est pourquoi dans cette thèse nous proposons une nouvelle approche de commande basée sur un modèle de prédiction pour contrôler un robot mobile manipulateur.
La nouvelle approche de commande que nous avons proposée est une combinaison d'une commande non-linéaire linéarisante (afin de transformer la dynamique non-linéaire du système en une dynamique totalement ou partiellement linéaire) et d'une commande prédictive à base de modèle CPM (afin d'obtenir de meilleures performances pour le système). Ceci constitue la principale contribution de cette thèse.
La stratégie de commande de ce robot se fait de la manière suivante: d'abord, on doit commander la base mobile pour qu'elle arrive à une position où l'objectif rentre dans l'espace de travail du bras manipulateur. Ensuite, on va commander le bras manipulateur pour que son organe terminal atteigne son objectif.
Dans nos travaux, pour surmonter le problème de non-linéarité, nous avons utilisé la commande non-linéaire linéarisante pour le bras manipulateur et le contrôle de linéarisation entrée-sortie pour la base mobile. Une fois les modèles linéaires obtenus, une approche de commande prédictive à base de modèle (CPM) est développée. Nous avons introduit un critère quadratique

Table of Contents

List of Figures

List of Tables

Notations

ANFIS Adaptive Neuro Fuzzy Inference System

ATAN2(y,x) Trigonometric function with two arguments, compute the inverse tangent

COPSO Constrained Particle Swarm Optimization

DDMR Differential drive mobile robot

DH Denavit-Hartenberg

DMC Dynamic Matrix Control

DOF Degree Of Freedom

EPSAC Extended Prediction Self-Adaptive Control

FDM Forward Dynamic Model

FGM Forward Geometric Model

FKM Forward Kinetic Model

FLC Fuzzy Logic control

GPC Generalized Predictive Control

IDM Inverse Dynamic Model

IGM Inverse Geometric Model

IKM Inverse Kinetic Model

Kg Kilogram

LQ linear quadratic

m Meter

MMR Mobile Manipulator Robot

MPC Model Predictive Control

Nm Newton.meter

NNC Neuron Network Control

PD Proportional Derivative

PFC Predictive Functional Control

PID Proportional Integral Derivative

rad radian

s Second

S.I.S.O Single Input and Single Output

TWR Two-Wheeled Robot

UPC Unified Predictive Control

WMR Wheeled Mobile Robot

General Introduction

Due to the continuous development in the production industries and to overcome the difficult and repetitive tasks for human, such as nuclear power plant inspection, welding automation, robotic surgery and automated agriculture, the interest in mobile manipulator arm is increasing significantly day after day. The first challenge for controlling any mobile robot, is the complexity of the mechanical structure of this type of robots (articulated rigid body), which leads a nonlinear models, hence the requirement to convert these complicated nonlinear models to simplified linear one therefore the linear control strategies can be applied, the second challenge is to develop a very challenging control approach capable of providing excellent dynamic performances while respecting the work space constraints, and this developed control strategy has to maintain these features for higher degrees of freedom, which makes the modeling challenge as well as the control part constitutes the big challenges for this thesis.

Due to challenging requirement to get very high dynamic performances, we have used dynamic models for controlling the manipulator arm and the mobile base; both models are strongly nonlinear. It is not possible to simplify these models because, when the manipulator robot moves with a high speed, a coupling effects between the different joints of the robot become no longer ignored, hence the need of using a suitable linearization technique in order to get a complete or partial linear model.
Several approaches for controlling manipulator robots and mobile robots are proposed in the literature, some of them are classical controls (P, PI, PD, PID) and others are modern controls (adaptive control, Linear quadratic control, optimal control, model predictive control, fuzzy control, neuron network control...etc.). The aim of the above control strategies is to calculate the required couple for the different articulation of the manipulator arm to achieve its desired target and for the mobile base to follow a preplanned trajectory.

After reviewing the above control strategies, and looking for their advantages and disadvantages, we have noticed that the model predictive control has a challenging features such as constraint handling, optimization control, multi-variable handling, disturbance cancellation, dynamic performance, and system delay's handling, and due to different challenges associated with the mobile manipulator robot, such as nonlinear dynamics of their models (Kinetic, dynamic), system performances (response time, overshooting and oscillations) and workspace constraints (Maximum torque and shortest trajectory), hence in this thesis we have selected the model predictive control as our contribution field, and proposed a new control approach based on a model predictive control (MPC) for controlling a manipulating mobile robot. The proposed control approach is a combination of a feedback linearization control (in order to transform the nonlinear dynamics of the system to one completely or partially linear) and an MPC control (in order to obtain better performance for the system), which constitutes the main contribution of this thesis. In this work, the proposed control approach is used to control different dynamics models of a

manipulator robots and a dynamic model of a non-holonomic wheeled mobile robot. For the manipulator arm, a feedback linearization technic was enough to convert the dynamic model to a complete one, while the chosen mobile base, which is a differential drive base, and due to its non-holonomic constraints, we had to change the dynamic equations accordingly prior to perform any linearization method, after that and by using input-output linearization technic we could successfully linearize the mobile base dynamic model. After the linearizing phase, the manipulator arm and the mobile base are converted to a linear systems with double integrator which allow us to develop our new MPC control strategy. The optimization of the proposed predictive control law is carried out analytically and by using a constrained particle swarm optimization (COPSO) technic. In order to show the effectiveness of the MPC control whose parameters have been optimized using a constrained particle swarm optimization (COPSO) technic, a comparative study with the classical MPC control was done.

This thesis is organized as follows:
In the beginning of the first chapter, we have listed many control strategies cited in the literature with a brief explanation as a state of the art regarding to the choice of the applied control approach, and highlight their advantages and disadvantages, after that we have given our reasons for selecting the model predictive control approach as our contribution field. Finally we have explained our work objectives and the road map to overcome its associated challenges, with a brief presentation of all our contributions.
In the second chapter, a new model predictive control approach for a dynamic model of a two-link manipulator robot is presented. This technique consists of linearizing a nonlinear dynamic model of the robot by using a feedback linearization control. Once, the linear model has been obtained, a predictive control approach is developed. We have introduced a quadratic criterion and these parameters are calculated to have a specific behavior of the closed loop system. Here, the objective is to control the robot arm from an initial configuration to the final configuration using a predictive control approach and it is obtained by minimizing a quadratic criterion. In order to show the efficiency of the proposed method, some simulation results are given. In addition, for validation purpose, a simulation study was done comparing the proposed MPC control approach with the linear quadratic (LQ) control based on the same feedback linearization, and a control approach proposed in the literature for the same problem. The results showed the efficiency of the proposed method. once we have successfully validate the proposed control strategy, we got a green light to extend it to higher degrees of freedom (dof), we have applied it for controlling a 3 dof manipulator arm with using of a graphic analysis to tuning the obtained controller gains, the simulations results shows the efficiency of the proposed graphical method .
In the third chapter, a model predictive control (MPC) for a differential-drive mobile robot (DDMR) based on the dynamic model is presented. The robot's mathematical model is nonlinear, which is why an input–output linearization technique is used, then based on the obtained linear model, an MPC control was developed. The predictive control law gains were acquired analytically by minimizing a quadratic criterion. In addition, to enable better tuning of the obtained predictive controller gains, a maximum torques limitation as well as the settling time limits were used and via the mobile robot torques and settling time graphs a trad-off gains were computed. To show the efficiency of the proposed approach, some simulation results are provided.
In the fourth chapter, a model predictive control approach is applied for controlling the whole mobile manipulator robot. This robot is composed of a differential-drive mobile

base and a 6 dof manipulator arm. The control strategy of this robot is simple, first we have to control the mobile base, so that it arrives at a position where the objective fits into the working space of a 6 dof manipulator arm. In the second step, we must control the manipulator arm so that its end effector achieves its objective. Here, we have used the constrained particle swarm optimization (COPSO) technic to tune the MPC gains developed in chapter 2 and chapter 3 for controlling a six degrees of freedom (DOF) mobile manipulator robot and meet the required performances. The parameters (h: horizon time of prediction, ρ : weight factor) of the MPC control are computed by identification between the gains found by the COPSO algorithm and those of the MPC. In order to show the efficiency of the proposed tuning method, some simulation results are given.

In the conclusion, we have presented our objectives in this thesis and the road map to overcome its associated challenges, with some future works and perspectives.

Chapter 1

State of the art

1.1 Introduction

The basic problem in controlling any mobile manipulator system is to define a suitable torques control τ to make the manipulator end effector reach a desired position and fulfill the workspace constraints. While for the mobile base is to follow a preplanned desired trajectory with a minimum tracking error.
The selected controller has to deal with non-linear models and its associated constraints and provide good dynamic performances and optimize the required energy.
In this chapter, we will expose the different control strategies that exist in the literature to control mobile manipulator robots. Several strategies control have been studied for manipulating robots, which we will summarize it briefly as below:

1.1.1 Independent Joint Control

This technic is used for manipulator robots, which have an actuator with high reduction ratio and doesn't require a high speed motion, in such way the system present a linear system behavior, therefore a simple PID controller is applied locally for each joint Fig. 1.1, each articulation of the manipulator robot is considered as a single input and single output system (S.I.S.O)[2,3].

The use of these reductions effectively increases the torque or the produced force by the actuators, in order to drive links of considerably large masses[4].
In practice, the tuning of PID controllers is very easy and straightforward. but the main disadvantage of this technic is when the manipulator moves with a high speed, in this

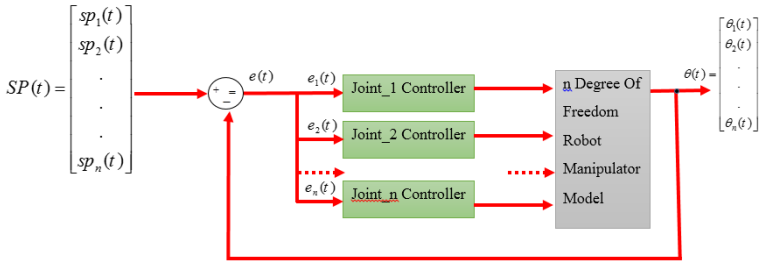


Figure 1.1: Independent Joint Control

case all the coupling effects between joints are no longer ignored and the whole system behavior become a non-linear, therefore the PID will be no more suitable for this control. For instance, in[5] a PID controller is applied for trajectory tracking in robotic arm manipulator systems, where one PID is applied for each joint, while the tuning method of the controller gains was performed by using Artificial Bee Colony optimization algorithm, the obtained results are satisfactory and competitive.
The authors in[6], have used an end-effector base 3 degree of freedom (3-DOF) rehabilitation platform, which uses brushless DC motor as actuators of the two revolute joints. For set point and trajectory tracking of the revolute joints, each joint and corresponding link has been modeled has a SISO system and a PI controller has been employed. The simulation results showed the effectiveness of the proposed control framework.
While in[7], The independent joint control approach control is used for manipulator robot due to the simplicity of the controller implementation's from one side, and from the order one is to suppress the disturbances at each robot joints due to its coupling effect. Therfore they developed simple robust independent joint controllers, which result in linear decoupled joint motions. In this approach, each joint is treated as a simple inertial system plus a disturbance torque representing all the unmolded dynamics. The disturbance is instantly estimated and rejected, thus allowing a Proportional Derivate control to be used. The controllers are inherently robust. Experimental evaluations were done on a PUMA 560 arm with very fruitful results. In case of higher degree of freedoms the coupling equations of the robotic system produces interactions between different controller components, which lead to small changes in controller parameters and unexpected performance. Moreover, the controller parameters are highly coupled with the initial and final conditions of the link joints. On-line tuning of the controller parameters is highly necessary to cover global system operation.

1.1.2 Jacobian control

This technic was used by Whitney,[8,9], he used the inverse Jacobian matrix of the manipulator arm in order to calculate the desired setpoint at each articulation Fig. 1.2
In case of using the kinematic model in the control loop, the inverse of the associated

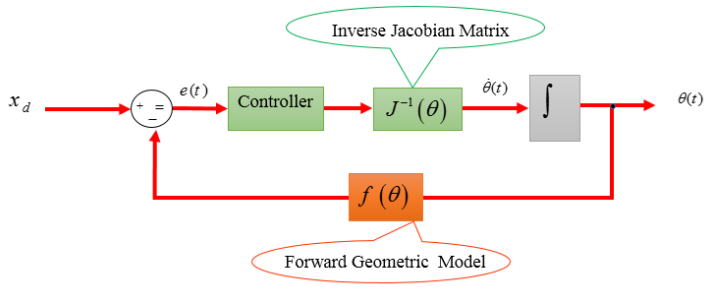


Figure 1.2: Jacobian Control

Jacobian matrix compute the velocities setpoint at each joints, while in case of using the dynamic model, the inverse of the associated Jacobian matrix compute the acceleration setpoint for each joints. The main drawback is the use of the inverse of the Jacobian matrix, which can become singular under certain conditions. The key point of kinematic

control is the solution to the inverse kinematics problem.
In[10], kinematic control has been applied for industrial robot manipulators. The desired end- effector trajectory specified in terms of position, velocity, and acceleration is transformed into joint trajectories by using a second-order Closed Loop Inverse Kinematic algorithm with Damped Least Square inverse to ensure smooth motion in the neighborhood of kinematic singularities. The joint trajectories are then used as references to a computed torque control scheme, which consists of a complete model, based feed forward action and an independent joint control PID action. Note that the damping factor in DLS is aimed at achieving a trade-off between solution accuracy (small end-effector errors) and solution feasibility (limited joint accelerations) which can be fixed by experiments on the specific robot.
Due to uncertain gripping points length, orientation and backload of the objects, which the manipulator robot has to picks up, to overcome these uncertain kinematics, the authors[11] propose a feedback control law for setpoint control without exact knowledge of kinematics, Jacobian matrix, and dynamics. For stability analysis, they used a Lyapunov function. The simulations as well as the experiments results showed that the end-effector's position converges to a desired position in a finite task space even when the kinematics and Jacobian matrix are uncertain.

1.1.3 Nonlinear decoupling control

This technic is also known as dynamic control or computed-torque control[2,12–16]. It is used when the application requires a high-speed motion with dynamic constraints; in this case, the control must take into account the interaction forces between all robot joints. Therfore the need of the robot dynamic model Fig 1.3. Basically, the computed torque

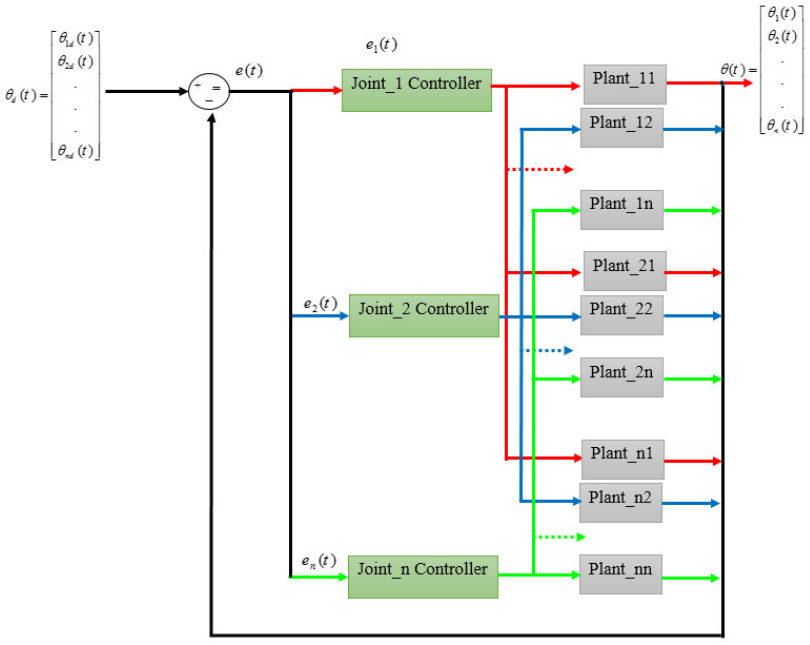


Figure 1.3: Decoupling Control

technique is a feed-forward control with feedback components. The feed-forward control minimizes the nonlinear effects which result from the interaction forces among all the various joints and the feedback components compute the necessary correction torques to

compensate for any deviations from the desired trajectory.
The control law is computed by using the nonlinear decoupling theory. Which leads to centralized non-linear control laws. After that with suitable choices of the feedback vector gains, the overall system becomes a stable decoupled system[4]. This type of technique allows the control in the joints space as well as in the Cartesian space. The main benefit of this technic is to provide full-decoupled joints with ability to compute its associated motion speed with high inertias. The main disadvantage of this method is that, it is strongly depend on the system model, in the presence of uncertainties or unknown parameters, the model may involve an imperfect decoupling. Which is the main drawback of this technic.

1.1.4 Adaptive control

Usually, we never have the exact knowledge of the robot model due to many problems with model formulation. Two common uncertainties that do not allow exact model cancellation in robotic applications are unknown link masses due to payload disturbances and unknown coefficients of friction. The method to deal with these types of parametric uncertainties would be to employ the torque computation controller with some fixed estimate of the unknown parameters instead of the actual parameters. Adaptive controllers are formulated by separating unknown constant parameters from known functions in the robot dynamic equation. The adaptation mechanism continuously reads the trajectory tracking error and update the unknown parameters in such way the system remains stable with a minimum tracking error's[2]. Fig 1.4.

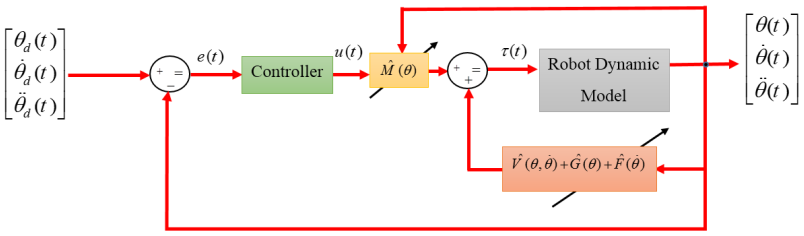


Figure 1.4: Adaptive Control

The authors in[17] proposed a new technique for advanced manipulator control based on adaptive independent joint control. The main features of the proposed control technic are first, due to its adaptive nature, knowledge of manipulator dynamic model and parameter values or the payload parameters are not required. Second, due to its decentralized structure and controller simplicity, the scheme is computationally very fast and is amenable to parallel processing implementation within a distributed computing architecture, with one microprocessor dedicated to each joint.

1.1.5 Lyapunov function based control

The Lyapunov based control is considered as a particular asymptotic stabilizing feedback control solution around a selected equilibrium point[4,18] A stabilizing controller is obtained based on the energy based Lyapunov approach, which exploits the physical properties of the involved mechanical system; it has been successfully applied to robot

manipulators to perform an exact tracking. The method does not seek linearization or decoupling, but only asymptotic convergence of the tracking error the main disadvantage is due to difficulties to find a Lyapunov function with negative derivatives for time varying systems.
The authors aim's in[19] is to stabilize a Two-Wheeled robot (TWR) by using its dynamic model, they developed a controller based on Lyapunov function, in such way the choice of the control law makes the derivate of the chosen Lyapunov function negative along the system trajectories with respect to the time. The simulation results showed that the robot is stable even with various torque disturbances, which proves the robustness and the global asymptotic stability of the closed loop system.

1.1.6 Passive control

This technic look at the robot manipulator as a passive system, which means system which does not create energy[4]. By considering, the desired joint vector's as an equilibrium posture for the system, the used control law has to modify the robot total energy in such way it can reaches the equilibrium posture with a minimum of energy, by using Hamilton formalism, while maintaining a global asymptotic stability.

1.1.7 Model Predictive control

Model Predictive Control (MPC) is an advanced control strategy, it is considered as an optimal control technic based on numerical minimization. it uses a predefined model to predict the system outputs, while the control signal is optimized at regular intervals with respect to a specific objective function. Prediction is to determine the future values of the control output, which lead to a better performance of the control systems, the suggested performance index should be a function of the estimated error over a prediction horizon and the predicted control signal over a control horizon[20–22].Usually it can be formulated as below:

$$J = \delta \sum_{j=N_1}^{N_2} \left[\hat{y}\left(t+j|t\right) - w\left(t+j\right)\right]^2 + \lambda \sum_{j=1}^{N_u} \Delta u^2\left(t+j-1\right) \tag{1.1}$$

where N_1 and N_2 are the minimum and maximum prediction horizons, N_u is the control horizon (typically $N_u = N_2$) and δ and λ are the weighting factors for future errors and control effort.
In the minimization of the cost function, the reference trajectory (if known in advance) must not be necessarily the same as the desired one: usually it is an approximation.
The cost function can be usually a constrained function.
In order to obtain the future control values, $u(t+k|t)$, it is necessary to minimize the functional J by using the prediction model:
- The predicted outputs, $\hat{y}(t+j|t)$, are calculated as a function of past values of inputs and outputs and future control signals, making use of the prediction model Fig 1.5.
- Substitute the compact prediction values in the cost function, J.
- Minimize J with respect to Δu where an analytical solution can be obtained for the quadratic criterion when a linear model is considered and there are not constraints.
- Non-trivial solution,$N_2 - N_1 + 1$ independent variables.
- The structure of the control law depends on the control horizon N_u: $\Delta u(t+j-1)^2$ for $j > N_u$.

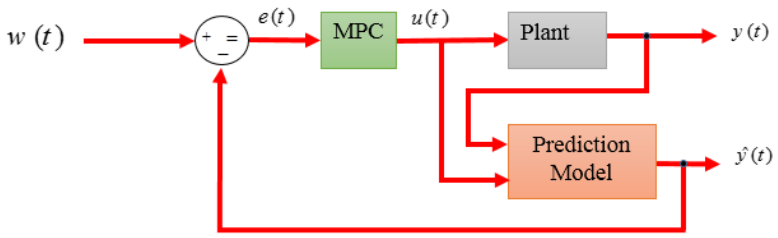


Figure 1.5: Model Predictive Control Closed Loop

- The receding horizon concept is then applied at each sampling period.
The potential advantages of prediction are avoidance of expected constraints, multivariable handling, disturbances cancellation and treating a difficult dynamics such as non-minimum phase behaviours and delays[23].
Many model predictive control algorithms have been used, such as :
- Dynamic Matrix Control (DMC)[24] and[25];
- Generalized Predictive Control (GPC)[26],[27] and[28];
- Extended Prediction Self-Adaptive Control (EPSAC)[29];
- Predictive Functional Control (PFC)[30];
- Unified Predictive Control (UPC)[31];
- Explicit MPC[32];
- Nonlinear MPC[33];
In[34] a model predictive control was developed for an autonomous mobile robot system with a 5-DOF manipulator arm, based on its dynamic model. The main idea consists of considering the state space time-varying matrices as constants within one optimization, and for each prediction time, there are a new matrices and new optimization, this technic converts the model system from a non-linear time-variant model to a non-linear time invariant model. Therfore the estimated system outputs and control inputs are calculated in accordance with the time invariant model formula.
In[35] a model predictive control was applied to solve the problem of the trajectory tracking and path following of an omnidirectional mobile robot. Two different models were considered the kinematic and the dynamic model. The results showed the effectiveness of the dynamic model in comparison with the kinematic one in tracking the trajectory and path without posture to achieve better tracking of the wheeled mobile robot (WMR) path.
The authors in[36] investigates the formation control of multiple differentially driven wheeled mobile robots (WMRs) based on the kinematic model and the leader-follower approach. The linear model of each robot with nonlinear dynamics is found through feedback linearization, while model predictive control is applied to the linear model to perform the formation control. The simulation results show the stability and the good dynamic performances of the proposed algorithm.

1.1.8 Robust control

The robustness study is required in the robots control when disturbances are not negligible or their dynamic models are uncertain which may be caused by the effects of imprecise measurements such as forces, positions or unknown robot load[4]. Therefore the nonlinear decoupling control technic will be no longer applicable, because it leads to unstable

system with the presence of these uncertainties, so the robust or adaptive control technic is required in such cases.

1.1.9 Quadratic Optimal control

This technic is used when control problem has a multiple solutions, and we want to choose the best one among them. Hence the importance of the performance index choice. This last should be a function of the system variable state, the control input and the required time for the system state to achieve some desired terminal state starting for initial one[4,37,38]. The optimal control law is the solution which makes the performance index as minimum as possible, In addition to that, the gains of the feedback control need to be computed in such way it ensures the system stability and the required dynamic behavior.
In[39], a Linear Quadratic Optimal Control is used to control an Inverted Pendulum, while the weighting matrices of the LQ controller are determined by using the Artificial Bee Colony Algorithm the simulation results prove the effectiveness of the LQC especially with the computed weighting matrices compared with trial and error method. In addition, it has been shown that it can optimize multiple control system characteristics such as settling time, overshoot and steady state error, in line with desired results.

1.1.10 Fuzzy Logic control

This technic is used mostly for the nonlinear systems[40] where the mathematical model presents a high complexity and its variables are highly dependent so that the decoupling process becomes very difficult. The fuzzy logic can handle the nonlinear systems due to the fact it has many nonlinear types of membership functions (Gaussian, trapezoidal), interference functions (Max, Min) in addition to defuzzification methods (Max, average, min). The fuzzy logic controller can very well describe the behavior of a desired system and translate human experience in the form of linguistic variables and fuzzy rules (if then), but due to lack of learning ability it cannot adapt to a new environment.
In[41], two independent fuzzy logic controllers were used to control 2 DOF arm manipulator robot. Fuzzy logic has allowed to overcome the difficulty of the mathematical model by using the linguistic model based on expertise. The fuzzy controller has improved the classic PD and PID controllers, it helps significantly to compensate the noise in the input parameters and allows the robot joints to follow a sinusoidal trajectory even in case of an increase in the amplitude or the frequency of the setpoint, it has been able to maintain these performances. It has been noticed as well that the change in the shape of the setpoint in a straight line as well as the sudden change in the mass of the second link of the robot is well supported by the fuzzy controller without the need for modifications on the fuzzy controller.
The authors in[42] used an optimal fuzzy Proportional Derivative (PD) controller for a RPP (Revolute-Prismatic-Prismatic) robot manipulator based on particle swarm optimization and inverse dynamics. The kinematic equations of the manipulator are obtained by using Denavit-Hartenberg approach and the Jacobi method for each of the arms of the robot, while the dynamic equations of motion are determined by using Lagrange method, Hence, the fuzzy PD controller gains are optimized via particle swarm optimization. The obtained results of the optimal fuzzy PD controller based on the inverse dynamics are compared to the outcomes of the PD controller, and it is illustrated that the optimal fuzzy PD controller shows a good controlling performance in comparison with other controller.

1.1.11 Neuron Network Control

The main aim of using the neural networks, in the control of manipulator robots is to improve the performance of the computed torque control, in such way in case of an imprecise computed torque controller we can add a feed forward or feedback neural compensator to refine the command and cancel the possible static deviation, this is in case of a well-known model, but in case of unknown mode, the NNC can be used as the inverse of the robot dynamic model which require an online identification based a specific function cost and back propagation of the error in order to update the network weights in parallel of classical controller[41]. The contribution of this work lies in the development of control methodologies based on neuro-fuzzy approaches. This approach combines the concepts of fuzzy logic and artificial neural networks to elaborate a hybrid intelligent controller called ANFIS (Adaptive Neuro Fuzzy Inference System). The latter is used to control a 6 DOF hydraulic manipulator arm. This work proposes two control techniques based on ANFIS: Adaptive neuro control fuzzy based on torque calculation (PD type), and a neuro-fuzzy adaptive control type PD plus integral correction in order to cancel the tracking error[4].

1.1.12 Sliding mode control

Sliding mode control is one of the robust control for nonlinear systems. Usually used whenever there a discrepancy between the actual plant and it associated mathematical model due to unknown external disturbance, parameter variation of the plant, un-modeled dynamics or due to un-dominant dynamics, which we have to ignore in order to reduce the system order. The advantage of the sliding mode is to reduce the order of the system to first order, robustness and it finite time convergence. The main disadvantage of this control is that the control law contains many shattering in order to maintain the system stable and fulfill the desired dynamic performance[43], [44].

1.2 Selected approach

While studying many control strategies and techniques cited in the literature, and due to the work space and dynamics constraints of the mobile manipulator robots, we have chosed the Model Predictive Control to be our contribution field for the following challenging features:
- Control output signal is an instantaneous optimum solution for the cost function.
- Constraints handling.
- Cancellation of a bounded disturbance.
- Minimizing the steady error with a good dynamic performance.
- Multi Inputs-Outputs systems handling.
- Excellent performance for the delay systems.

1.3 Objectives and Contributions

The aim of our work is to develop a new control law for a mobile manipulator robot. As mentioned previously, we have focused on the model predictive control as a contribution

field for our study. After a deep research, we could introduce a novel model predictive control for mobile manipulator robot and our contributions are as follow:

- Manipulator Arm: we proposed a new control approach based on a model predictive control (MPC) for controlling a different kind of manipulator robots[45],[46], and[47](2 dof, 3 dof and 6 dof). The proposed control approach is a combination of a feedback linearization control (in order to transform the nonlinear dynamics of the system to one completely or partially linear) and an MPC control (in order to obtain better performance for the system). The combination between a nonlinear control and an MPC control did not exist in the literature hence our contribution.

- Mobile base: We have chosen a deferential drive mobile robot (DDMR) as a mobile base for our manipulator arm, by taking the non-holonomic constraints, we could successfully applied our novel proposed approach of control (combination between nonlinear control and an MPC control) and get a good dynamic performance. Especially after tuning the controller gains with a specific graphical analysis[48].

Chapter 2

Model predictive control of Manipulator robots

2.1 Introduction

The interest of arm robotics has increased due to the development of industrial production which require the execution of many difficult and repetitive tasks with high precision during a short time [1-2]. The mechanical structure of this class of robots is complex (articulated rigid body) which makes the task of control more difficult.[49–58]. In this chapter, we have proposed a novel approach for controlling a two-link arm robot. This approach involves determining model predictive control dynamics of a manipulator robot with two degrees of freedom (DOF). The dynamic model of this robot is nonlinear, so a feedback linearization control is applied to the robot dynamic model to make it linear. Next, based on the obtained linear model, an MPC controller is developed and a quadratic criterion is minimized. To prove the efficiency of this new approach we have made a comparison study with a linear quadratic controller (LQC), the simulation results showed that our new approach of MPC has a superior system performance than the LQC approach. After that, we have extended the proposed approach on a three degree of freedom manipulator arm, while the controller tuning parameters were done via a graphical tool (maximum, the minimum torque and the settling time curves for several values of ω_0), the simulation results showed that the proposed graphic tool have provided an excellent tuning to our new approach which is clearly shown in the simulation results.

2.2 Two DOF Manipulator Arm Robot Description's

The manipulator robot shown in Fig.2.1 has two degrees of freedom, where $\theta_{i=1,2}$ are the joints angles, while $M_{i=1,2}$ and $L_{i=1,2}$ are respectively the mass and the length of the first ($i = 1$) and the second ($i = 2$) link of the manipulator arm robot. The variable g denote the gravity force.
The manipulator arm task is a pickup and drop operation for a specific object located in the plan (OXY) from an initial position (x_0, y_0) to a desired one (x_d, y_d).
(OXY) is the robot coordinate system, attached to the frame center O, where the end effector coordinate are denoted as (x, y) and the first link coordinate are denoted as (x_1, y_1).

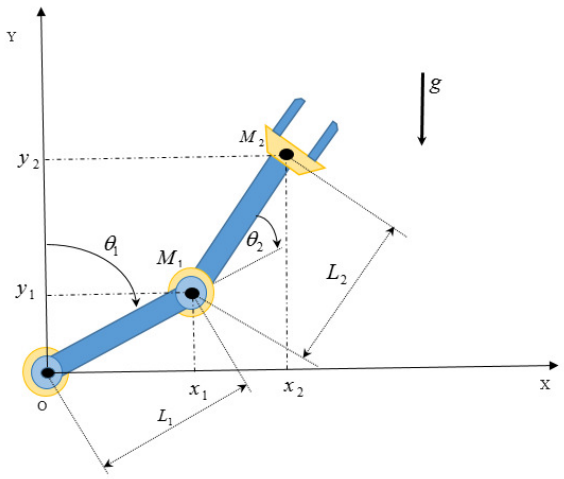


Figure 2.1: Two DOF Manipulator Robot

2.2.1 Manipulator Arm Models

2.2.2 Manipulator Geometrics

Geometric Model is a mathematical model to specify and compute the end effector position and orientation as a function of all prismatic or rotoïdes joints configuration of the robot. There are many methods to define the geometric model[59],[60],[61],[62],[63],[64] and the most used one is Denavit-Hartenberg method.

2.2.2.1 Forward Geometric Model

The Forward Geometric Model (FGM) of a mechanism groups together all the geometric constraints that must be respected by the articular variables θ_i in order to establish the relationship between the configuration of the mechanism defined in the generalized coordinates[65],the Forward geometric model is unique and is given in the form of explicit equations:

$$X = f(\theta) \tag{2.1}$$

Where:
$\theta = [\theta_1\theta_2...\theta_n] \in R^n$: Joint Variables Vector
$X = [x_1x_2...x_n] \in R^n$: Operational variables vector
To calculate the DGM, we assume that the robot links are perfectly rigid and the joints mechanically perfect, there are several methods to determine the FGM, and the most used one is of Denavit-Hartenberg[59]But this method, developed for simple open structures, presents ambiguities when it is applied to robots having closed or tree structures, there is also the Khalil and Kleinfinger notation which allows the homogeneous description with a minimum number of parameters, of simple and complex open architectures[66].

2.2.2.2 Inverse Geometric Model

The Inverse Geometric Model (IGM) allows to calculate the articular coordinates corresponding to a given posture of the end effector in Fig.2.2, the model is written:

$$\theta = f^{-1}(X) \tag{2.2}$$

When it is not possible to find an explicit form of the inverse geometric model (analytical

Figure 2.2: Forward and Inverse Geometric Models

solution) there are several methods to resolve this problem under an explicit form, we present the following three methods:
- Paul method is applied for robots having a simple geometry, where most of the distances (r and d) are zero and most of the angles (theta and alpha) are zero or $\pi/2$, the IGM can be analytically obtained using the Paul method, most available industrial robots can be solved using this method[67] and[68].
- Pieper method :deals with six degree-of-freedom decoupled robots having three prismatic or rotoid joints with a spherical joint, The IGM can be computed by solving two sub-problems, each having three unknowns[69].
- Raghavan-Roth method is suitable for six degree-of-freedom robots with general geometry[70].

2.2.3 Compute Of 2 DOF Manipulator Geometric Models

2.2.3.1 Compute Of the Forward Geometric Model

Based on the Fig.2.1, we can calculate the coordinates of the end effector manipulator robot with respect to the frame (OXY) as follows:

$$\begin{cases} x(t) = L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2) \\ y(t) = L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) \end{cases} \tag{2.3}$$

The equation (2.6) presents the geometric model which gives us the robot end effector coordinates based on its articulations values during the its motion.
We can also obtain the same equations by using Denavit-Hartenberg[59] method as follows:
The homogeneous transformation matrices of the two segments are given by:

Joints	DH Parameters			
n	θ_n	α_n	d_n	a_n
1	$\pi/2 - \theta_1$	0	0	L_1
2	$\pi/2 - \theta_2$	0	0	L_2

Table 2.1: Denavit Hartenberg representation for 2 DOF Manipulator Arm

$$T_{01} = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & L_1 cos(\theta_1) \\ sin(\theta_1) & cos(\theta_1) & 0 & L_1 sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{12} = \begin{bmatrix} cos(\theta_2) & -sin(\theta_2) & 0 & L_2 cos(\theta_2) \\ sin(\theta_2) & cos(\theta_2) & 0 & L_2 sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{02} = T_{01} * T_{12} = \begin{bmatrix} cos(\theta_1 + \theta_2) & -sin(\theta_1 + \theta_2) & 0 & L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) \\ sin(\theta_1 + \theta_2) & cos(\theta_1 + \theta_2) & 0 & L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Forward geometric model, which represent the coordinate of the end effector can be obtained as well from the fourth column of T_{02} as follows:

$$T_{02} = \begin{bmatrix} A_x & B_x & C_x & E_x \\ A_y & B_y & C_y & E_y \\ A_z & B_z & C_z & E_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where: $E = [E_x E_y E_z]^T$ is end effector coordinate vector, therefore:

$$\begin{cases} E_x = x(t) = L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2), \\ E_y = y(t) = L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2), \\ E_z = z(t) = 0 \end{cases} \tag{2.4}$$

2.2.3.2 Compute Of the Inverse Geometric Model

The inverse geometric model, is to find the θ_i as a function of the end effector coordinates. it is easy to calculate θ_2 first as follow:

$$\begin{aligned} x(t)^2 + y(t)^2 &= (L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2))^2 + (L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2))^2 \\ &= L_1^2 + L_2^2 + 2L_1 L_2 cos(\theta_2) \end{aligned}$$

let suppose that:

$$A = (x(t)^2 + y(t)^2 - L_1^2 - L_2^2)/(2L_1 L_2)$$

Therefore:

$$\theta_2 = ATAN2(\pm\sqrt{(1-\tau^2)}, \tau) \tag{2.5}$$

note that, ATAN2(y,x) returns the arc tangent of the two number x and y and determine the quadrant of the result based on the sign of both arguments.

Once the value of θ_2 is known let proceed for the calculation of θ_1 we can rewrite the equation (2.4), as follow:

$$\begin{cases} x(t) = [L_1 + L_2 cos(\theta_2)]sin(\theta_1) + [L_2 sin(\theta_2)]cos(\theta_1), \\ y(t) = [L_1 + L_2 cos(\theta_2)]cos(\theta_1) - [L_2 sin(\theta_2)]sin(\theta_1), \end{cases} \tag{2.6}$$

We can easly prove that the solutions of the above equations are as follow:

$$\begin{cases} cos(\theta_1) & = \frac{y(t)[L_1+L_2 cos(\theta_2)]+x(t)[L_2 sin(\theta_2)]}{x(t)^2+y(t)^2} \\ sin(\theta_1) & = \frac{x(t)[L_1+L_2 cos(\theta_2)]-y(t)[L_2 sin(\theta_2)]}{x(t)^2+y(t)^2} \end{cases} \tag{2.7}$$

where:

$$\theta_1 = ATAN2(sin\theta_1, cos\theta_1) \tag{2.8}$$

2.2.4 Manipulator Kinetic Models

As a general description, Kinematic modelling is the study of the motion of mechanical systems without considering the forces that affect the motion[71], [64], [72] [73], [74], [75], [76], [77]. Kinematics focus one the relationships relating the linear and angular velocities of the end-effector to its associated joint velocities. This velocity relationship is then determined by the Jacobian matrix, The Jacobian matrix is one of the most useful element in the analysis and control of robot motion. Such as identification of the singular configuration, calculation of the inverse geometric model, obtaining of the dynamic model of motion and transformation of torques from the end effector to each actuator associated for all joints[73].

2.2.4.1 Forward Kinetic Model

The forward kinematics problem is to find out the mathematical function between the individual joints of the robot manipulator and the associated position and orientation of the end-effector. Stated more formally, the forward kinematics aim is to obtain the position and orientation of the end-effector, based on given values for the joint variables of the robot. Note that, the joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints[74] .

2.2.4.2 Inverse Kinetic Model

Inverse kinematics goal is to determine all possible and feasible sets of joint variables, which ensure the specified positions and orientations of the manipulator's end-effector with respect to the robot frame. In practice, a robot manipulator control requires precise knowledge of the end-effector position and orientation during its preplanned trajectory for the instantaneous values of all joints. Many industrial applications such as welding

and certain types of assembly operations require that a specific path should be negotiated by the end-effector. To achieve this, it is necessary to find the corresponding motion of each joint, which will produce the desired tip motion. This is a typical case of inverse kinematic application[74] .
To obtain the inverse kinematic model, we reverse the direct kinematic model by solving a system of linear equations. There are two types of solutions either analytical or numerical:
- The analytical solution: the analytical methods are suitable for solving the inverse problem when the Jacobian matrix associated with the mechanism is regular. Their implementation is simplified when the robot wrist has three rotoidal joints with competing axes: the problem is then reduced to the inversion of two regular matrices of order three. It has the advantage of considerably reducing the number of operations, however all the singular cases must be treated separately[78].
- Numerical methods: are more general, the most widespread being based on the notion of pseudo-inverse: the algorithms deal in a unified way with regular, singular and redundant cases. They require a relatively large calculation time.

2.2.5 Compute of 2 DOF Manipulator Kinetic Models

2.2.5.1 Compute of the Forward Kinetic Model

it is easly obtained by derivating the Forward geometric model as follow:

$$\begin{cases} \dot{x}(t) = [L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2)]\dot{\theta}_1 + [L_2 cos(\theta_1 + \theta_2)]\dot{\theta}_2, \\ \dot{y}(t) = [-L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2)]\dot{\theta}_1 - [L_2 sin(\theta_1 + \theta_2)]\dot{\theta}_2, \end{cases} \tag{2.9}$$

where:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) & L_2 cos(\theta_1 + \theta_2) \\ -L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2) & -L_2 sin(\theta_1 + \theta_2) \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}.$$

we can rewrite the kinematic model as below:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = J(\theta_i) \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}.$$

where $J(\theta_i)$ is the Jacobian matrix and its formula is as follow:

$$J(\theta_i) = \begin{pmatrix} L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) & L_2 cos(\theta_1 + \theta_2) \\ -L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2) & -L_2 sin(\theta_1 + \theta_2) \end{pmatrix} \tag{2.10}$$

2.2.5.2 Compute of the Inverse Kinetic Model

The interesting feature of the kinematic model is its linearity with respect to speeds, in such way we can calculate the angular velocities $\dot{\theta}_i$ by using the inverse of the Jacobin

matrix as follow:

$$\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} = J^{-1}(\theta_i)\begin{pmatrix}\dot{x}(t)\\ \dot{y}(t)\end{pmatrix}$$

where:

$$J^{-1}(\theta_i) = \frac{1}{\Delta}\begin{pmatrix}-L_2 sin(\theta_1+\theta_2) & -L_2 cos(\theta_1+\theta_2)\\ L_1 sin(\theta_1)+L_2 sin(\theta_1+\theta_2) & L_1 cos(\theta_1)+L_2 cos(\theta_1+\theta_2)\end{pmatrix} \tag{2.11}$$

And: $\Delta = L_1 L_2 sin(\theta_2)$

Therefore the condition for the Jacobian inverse matrix to be exist is: $sin(\theta_2) \neq 0$
hence, the singularity points for the above 2 dof manipulator robot are: $\theta_2 = k\pi, k = 0, \pm1, \pm2, \ldots$, where the two singularity positions are illustrated in the figures Fig. 2.3 and Fig. 2.4.

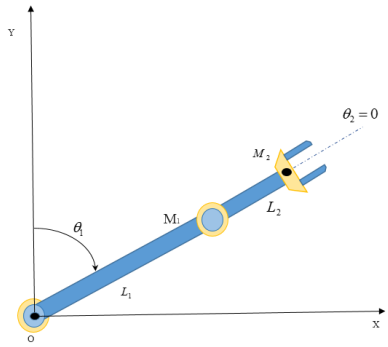


Figure 2.3: singularity at $\theta_2 = 0$

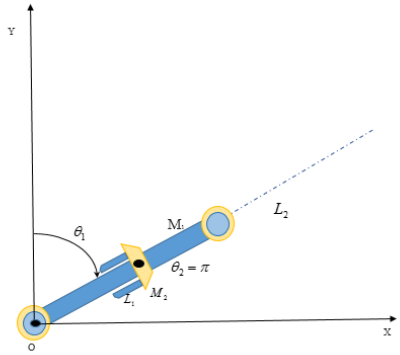


Figure 2.4: singularity at $\theta_2 = \pi$

2.2.6 Manipulator Dynamic Models

Dynamics is the study of mechanical system motion taking into consideration the different forces that affect it. To move a robot's links, it is required to have actuators capable of applying large enough forces and torques on the joints to move them at a desired acceleration and velocity. Otherwise, the joints may not be able to move as fast as needed, therefore, the robot end effector may not maintain its desired positional accuracy. To compute how strong each actuator must be, it is mandatory to determine the dynamic relationships that govern the motions of the manipulator robot. These relationships are the force-mass-acceleration and torque-inertia-angular-acceleration equations. Based on these equations, and considering the external loads on the robot, the designer can calculate the maximum loads to which the actuators may be subjected, thereby designing the actuators that can handle the required forces and torques. In general, the dynamic equations can be used to determine the equations of motion. This means that, by knowing the forces and torques, we can anticipate how a manipulator robot will move. These equations are also used to analyze the effects of different inertial loads on the robot, and depending on the desired accelerations, whether certain loads are allowed or not.In addition to that, the dynamic equations allow the designer to investigate the relationship between different elements of the robot and design its components appropriately. Many techniques such as Newtonian, Lagrangian and Hamiltonian mechanics can be used to find the dynamic equations for manipulator robots. However, due to the fact that Lagrangian formalism is commonly used and the easiest one, we opt to use it as technic to obtain the dynamics of the manipulator robot and analyze its motion in order to design a suitable controller.[77],[79],[80],[68],[81] and[82].

2.2.6.1 Forward Dynamic Model

The forward dynamic model is a function, which maps the the applied torque to the robot manipulator and the joint position, velocity and acceleration, so that we can compute $q(t)$, $\dot{q}(t)$ and $\ddot{q}(t)$ as a function of time.

$$\begin{cases} \ddot{q} &= M(q)^{-1}[\tau - C(q,\dot{q})\dot{q} - g(q)] \\ \dot{q} &= \int \ddot{q} dt \\ q &= \int \dot{q} dt \end{cases} \tag{2.12}$$

2.2.6.2 Inverse Dynamic Model

The inverse dynamic model compute the torques as a function of the positions, velocities and the accelerations of the different joints, that the manipulator robot's motors must deliver in order to bring the robot's end-effector to its desired position while respecting the environment constraints. We can represent the inverse dynamic model as bellow:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \tag{2.13}$$

where:
τ: Applied joint torque
$M(q)$: Inertia
$C(q,\dot{q})$: Coriolis and Centripetal

$g(q)$: Gravity
$\ddot{q}$: Joint acceleration
$\dot{q}$: Joint velocity
q: Joint position

2.2.7 Compute of 2 DOF Manipulator Forward Dynamic Model

Lagrange formalism describes the equations of motion in terms of work and energy of the robot. The general form representing the movement of a manipulator robot constituted of n degrees of freedom and n articulations, is written according to Lagrange:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \tag{2.14}$$

i=1,2,..,n
L; Lagrange Function
q: Manipulator Joints position's
$\dot{q}$: Manipulator Joints Velocity's
note that:

$$L = E_K - E_P \tag{2.15}$$

where:
E_K: Total Kinetic Energy of the Manipulator Robot
E_P: Total Potential Energy of the Manipulator Robot
In case of two dof manipulator robot with two rotoïde joints therefore $q_i = \theta_i$, thus the Lagrange equation will be written as follow:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \tag{2.16}$$

i=1,2

In order to calculate the Lagrangian of the two dof manipulator robot, we need first to calculate its total kinetic and potential energies.

$$\begin{cases} E_K = E_{K1} + E_{K2} \\ E_{K1} = \frac{1}{2} M_1 V1^2 \\ E_{K2} = \frac{1}{2} M_2 V2^2 \end{cases} \tag{2.17}$$

where:

$$\begin{cases} V_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \\ V_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \end{cases} \tag{2.18}$$

From the manipulator robot shown in Fig.2.1, the coordinate of the first and the second links are as follow:

$$\begin{cases} x_1(t) = L_1 S_1 \\ y_1(t) = L_1 C_1 \end{cases} \tag{2.19}$$

and:

$$\begin{cases} x_2(t) = L_1 S_1 + L_2 S_{12}, \\ y_2(t) = L_1 C_1 + L_2 C_{12}, \end{cases} \tag{2.20}$$

where:

$$\begin{cases} cos\theta_1 = C_1 \\ sin\theta_1 = S_1 \\ cos(\theta_1 + \theta_2) = C_{12} \\ sin(\theta_1 + \theta_2) = S_{12} \end{cases} \tag{2.21}$$

therefore:

$$\begin{cases} \dot{x}_1(t) = L_1 \dot{\theta}_1 C_1 \\ \dot{y}_1(t) = -L_1 \dot{\theta}_1 S_1 \end{cases} \tag{2.22}$$

and:

$$\begin{cases} \dot{x}_2(t) = L_1 \dot{\theta}_1 S_1 + L_2(\dot{\theta}_1 + \dot{\theta}_2) C_{12}, \\ \dot{y}_2(t) = -L_1 \dot{\theta}_1 S_1 - L_2(\dot{\theta}_1 + \dot{\theta}_2) S_{12}, \end{cases} \tag{2.23}$$

we can easly prove that the total kinetic energy of the manipulator robot is as follow:

$$\begin{aligned} E_K = \tfrac{1}{2}(M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} M_2 L_2^2 \dot{\theta}_1^2 + M_2 L_2^2 \dot{\theta}_1 \dot{\theta}_2 \\ + \tfrac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_2 L_2 (\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2) C_2 \end{aligned} \tag{2.24}$$

and the potential energy is given by the following formula:

$$E_P = g M_1 L_1 C_1 + g M_2 (L_1 C_1 + L_2 C_{12}) \tag{2.25}$$

finally the Lagrangian of the two dof manipulator robot is given as below:

$$\begin{aligned} L = \tfrac{1}{2}(M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} M_2 L_2^2 \dot{\theta}_1^2 + M_2 L_2^2 \dot{\theta}_1 \dot{\theta}_2 + \tfrac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 \\ + M_2 L_2 L_2 (\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2) C_2 - g M_1 L_1 C_1 - g M_2 (L_1 C_1 + L_2 C_{12}) \end{aligned} \tag{2.26}$$

By replacing the formula of L in (2.26) and resolving the equation (2.16) the dynamic model of a robotic arm with two degrees of freedom is given by the following formula:

$$\begin{cases} M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \\ Y = \theta \end{cases} \tag{2.27}$$

where:
$\theta = [\theta_1 \theta_2]^T$: is the Joints position Vector
$\tau = [\tau_1 \tau_2]^T$: is the Joints Torque Vector

Y: is the Output Vector

$G(\theta) = \begin{bmatrix} -(M_1 + M_2)gL_1S_1 - M_2gL_2S_{12} \\ -M_2gL_2S_{12} \end{bmatrix}$: is the gravity torques Vector's

$C(\theta, \dot{\theta}) = \begin{bmatrix} -M_2L_1L_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2)S_2 \\ -M_2L_1L_2\dot{\theta}_1\dot{\theta}_2S_2 \end{bmatrix}$: is the Coriolis and centrifugal forces Vector's;

$M(\theta) = \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix}$: is the inertia matrix;

with:
$D_1 = (M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2C_2$
$D_2 = M_2L_2^2 + M_2L_1L_2C_2$
$D_3 = D_2$
$D_4 = M_2L_2^2$

2.3 Control Algorithm

In this section, a predictive control of a robotic arm with two DOF is developed. For that, we consider the nonlinear dynamic model given by Eq. (2.27). First, we determine a feedback linearization control to make the model (2.27) linear. Once the linear model has been obtained, a model predictive control will be designed in the second step.

2.3.1 Feedback Linearization Control

The main idea of this technique is to transform the nonlinear dynamics of the system to one completely or partially linear, such that linear control approaches can be applied to stabilize it[83],[84] Here, the control approach with feedback linearization is developed for a dynamic model (2.27) of the two-link robot arm. So, we differentiate the output Y until the control input τ appears. In our case, the control input τ appears in the second derivative of the output Y. This implies that the relative degree is equal to two. The second derivative of Y is given by the following formula:

$$\ddot{Y} = \ddot{\theta} = M(\theta)^{-1}[-C(\theta, \dot{\theta})\dot{\theta} - g(\theta) + \tau] = v \tag{2.28}$$

where:
$v = [v_1 v_2]^T$: is the synthetic control vector
Finally, from (2.28) we get the feedback linearization control as follow:

$$\tau = M(\theta)v + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{2.29}$$

the Fig. 2.5 illustrate the the feedback linearization control loop.

Applying the control law given by (2.29) to the nonlinear system (2.27), the dynamic model of the manipulator robot with two DOF, becomes a linear system double integrator. The relative degree is equal to two. This means that by using the control law (2.29), we obtain a complete linearization of the nonlinear system (2.27) and we get a linear system for each joint variable.

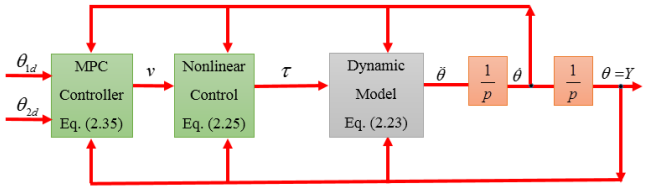


Figure 2.5: MPC and Feedback linearization Close-loop diagram

$$\begin{cases} \frac{\theta_1(p)}{v_1(p)} = \frac{1}{p^2} \\ \frac{\theta_2(p)}{v_2(p)} = \frac{1}{p^2} \end{cases} \tag{2.30}$$

where p is a Laplace variable. The linearization of the nonlinear system has been done. So,we can develop minimum time control for the two-link robot arm which will be the goal of the next paragraph.

2.3.2 Model Predictive Control

In the case of a robot arm with two DOF and after application of the feedback linearization (2.29) to the nonlinear system (2.27), we obtain the following two decoupled linear systems.

$$\begin{cases} \ddot{\theta}_1 = v_1 \\ \ddot{\theta}_2 = v_2 \end{cases} \tag{2.31}$$

Considering the first equation of (2.31). This system can be rewritten in the state-space form:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = v_1(t) \\ y(t) = x_1(t) \end{cases} \tag{2.32}$$

where:

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \theta_1 & \dot{\theta}_1 \end{bmatrix}$$

v_1 is a synthetic control of the first link of the robot and Y is the output.
Now, we are going to develop a model predictive controller (MPC)[85],[86] for the first link of the robot arm. The MPC controller for the second link of the robot will be developed in the same way as the first.
Assuming $v_1(t) = v_1$ constant in the time interval [t t+h] , where h is horizon time of prediction.

By using the equation 2.31, and the equation (2.32), we get the prediction model as follows:

$$\begin{cases} \dot{\theta}_1(t+h) = v_1 h + \dot{\theta}_1(t) \\ \theta_1(t+h) = \frac{1}{2} v_1 h^2 + \dot{\theta}_1(t) h + \theta_1(t) \end{cases} \tag{2.33}$$

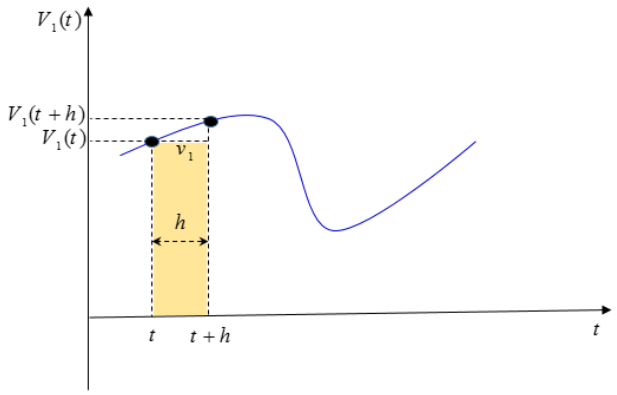


Figure 2.6: Prediction Horizon

,
the desired angle of the first joint θ_{1d} is constant as well as the second joint.

Let define the cost function J for one-horizon time which stabilize the first joint θ_1 as follow:

$$J = e_1^2(t+h) + \rho \dot{e}_1^2(t+h) \tag{2.34}$$

,

where: $e_1(t+h)$ is the predicted joint position error $\dot{e}_1(t+h)$ is the predicted joint velocity error as follow:

$$\begin{cases} e_1(t+h) = \theta_{1d} - \theta_1(t+h) \\ \dot{e}_1(t+h) = -\dot{\theta}_1(t+h) \end{cases} \tag{2.35}$$

The horizon time h and the weight factor ρ are both positive parameters to be determined later.
By substituting the prediction model(2.33) into (2.34) the quadratic cost function will be:

$$\begin{aligned} J = (\rho h^2 + 0.25h^4)v_1^2 + (\theta_1 h^2 - \theta_{1d} h^2 + \dot{\theta}_1 h^3 + 2\rho h \dot{\theta}_1)v_1 \\ + \theta_{1d}^2 + \theta_1^2 - 2\theta_{1d}\dot{\theta}_1 h - 2\theta_{1d}\theta_1 + 2\dot{\theta}_1\theta_1 h + \dot{\theta}_1^2 h^2 + \rho\dot{\theta}_1^2 \end{aligned} \tag{2.36}$$

By minimizing the criterion J with respect to v_1, the new approach of the model predictive controller is as follow:

$$v_1(t) = k_3\theta_{1d} - k_1\theta_1(t) - k_2\dot{\theta}_1(t) \tag{2.37}$$

where the control law gains are as follow:

$$\begin{cases} k_1 = \frac{2}{h^2+4\rho} \\ k_2 = \frac{2h^2+4\rho}{h^3+4\rho h} \\ k_3 = k_1 \end{cases} \tag{2.38}$$

The block diagram of the closed-loop system is presented as depicted in 2.7:

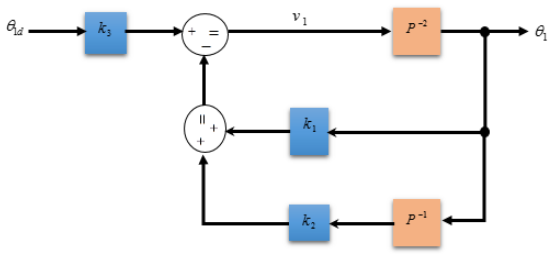


Figure 2.7: Closed-Loop Transfert Function

Regarding the response of the system, we would like to have a behavior similar to a system of the second order:

$$\frac{\omega_0^2}{p^2 + 2\zeta\omega_0 p + \omega_0^2} \tag{2.39}$$

where ζ is a damping factor and ω_0 is a natural frequency.
The transfer function of the system presented in Fig 2.7 is given as below:

$$\frac{\theta_1(p)}{\theta_{1d}(p)} = \frac{k_3}{p^2 + k_2 p + k_1} \tag{2.40}$$

Taking into account (2.38),(2.39) and (2.40) we obtain:

$$2\zeta\omega_0 = \frac{2h^2 + 4\rho}{h^3 + 4\rho h} \tag{2.41}$$

$$\omega_0^2 = \frac{2}{2h^2 + 4\rho} \tag{2.42}$$

based on equation.2.42 we can easly prove that:

$$\rho = \frac{2 - (\omega_0 h)^2}{4\omega_0^2} \tag{2.43}$$

Substituting equation (2.43) into equation (2.41),we get the following second order equation:

$$\omega_0^2 - 4\zeta\omega_0 h + 2 = 0 \tag{2.44}$$

where h is the variable to look for. Choosing $\zeta = \frac{\sqrt{2}}{2}$ and for different values of ω_0,we determine the parameter h from (2.44). Now, the horizon time is obtained, we use the Eq.(2.43) to determine the weight factor with h and ρ being both real and positive.

ω_0(rad/s)	1	4	8	9
h(s)	1.41	0.35	0.17	0.15
ρ	$1.82x10^{-8}$	$1.14x10^{-9}$	$2.85x10^{-10}$	$1.66x10^{-10}$
τ_{1min}	-26.28	-120.50	-422.01	-529.16
τ_{1max}	21.15	53.66	170.77	212.24
τ_{2min}	-1.76	-28.53	-115.68	-145.80
τ_{2max}	1.21	19.23	78.32	99.21
$t^{\theta_1}_{r\pm5\%}$(s)	6.19	1.55	0.77	0.68
$t^{\theta_2}_{r\pm5\%}$(s)	6.19	1.55	0.77	0.68
D_{θ_1} (%)	13.56	13.58	13.58	13.58
D_{θ_2} (%)	13.56	13.58	13.58	13.58

Table 2.2: System performances for each value of h and ρ

The performances of the system are computed for each determined value of h and ρ and they are summarized in Table (2.2)

where $t^{\theta_i}_{r\pm5\%}$ and $D_{\theta_i}(\%)$ i=1,2 are respectively the settling time within the band (±5%) of the final value and the overshoot in (%) of first joint angle (i=1) and the second joint angle (i=2).
Analyzing the Table 2.2, we observe that the horizon time h has an influence on the variation range of the control signals (robot torques). When h is small then the variation range of the robot torques is big.
For an acceptable control signals range ($max|\tau_i|$ for i=1,2)[3], we choose the horizon time h=0.35 (s) and the weight factor $\rho = 1.14x10^{-9}$ which corresponds to $\omega_0 = 4$ (rad/s). With these chosen parameters, we obtain a better system performance than in[3].

2.3.3 Simulations and Results

In order to show the efficiency of the proposed approach, some simulation results are given. For simulation purpose, we assume that the mass and the length of the first and the second links of the robot arm are $M_{i=(1,2)} = 1$ (Kg) and $L_{i=(1,2)} = 1$ (m), respectively. The initial and the desired orientations of the first and the second links of the robot arm are $\theta_1(0) = \pi/2$, $\theta_2(0) = -\pi/2$,$\theta_{1d} = -\pi/2$ and $\theta_{2d} = \pi/2$, respectively. According to the obtained horizon time and the weight factor,the gains of the MPC controller are:
$k_1 = 15.99$, $k_2 = 5.65$.

Fig.2.8 represents the followed trajectory of the end effector of the robot arm to reaches the desired position of both joints. Fig.2.9 and Fig.2.10 present the convergence of the joint angles θ_1 and θ_2 to their reference values. We can notice an asymptotic convergence of both joint variables. In Fig. 2.12 the robot synthetic controls v_1 and v_2 given by the Eq. (2.37) are shown. As we can see, the synthetic controls reach zero when the end effector of the robot reaches its objective. Fig. 2.13 shows the robot torques that can be obtained from synthetic controls using Eq. (2.28) angles errors e_1 and e_2 of the two-link of the robot arm towards zero using the proposed approach of control is depicted in Fig. 2.11.

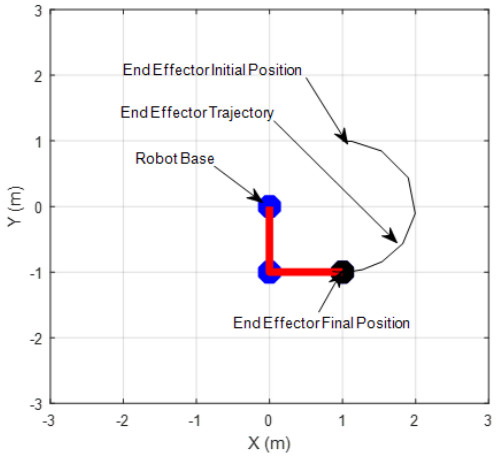


Figure 2.8: Final situation:the robot end effector reaches its objective point

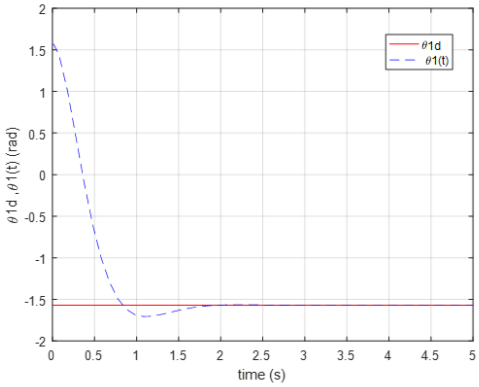


Figure 2.9: Real and the desired orientation of the first link of the robot

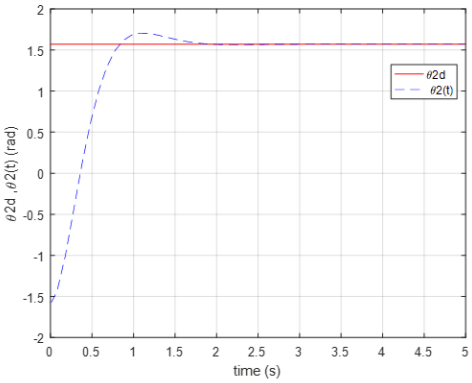


Figure 2.10: Real and the desired orientation of the second link of the robot

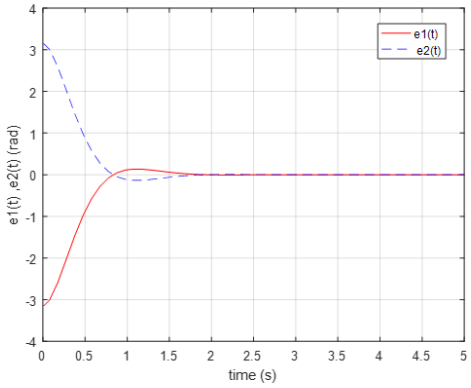


Figure 2.11: Errors in the joint angle

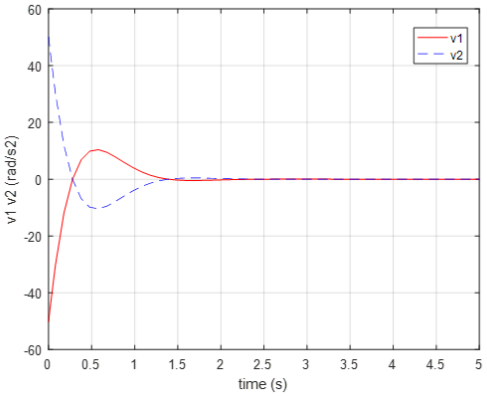


Figure 2.12: Robot synthetic controls

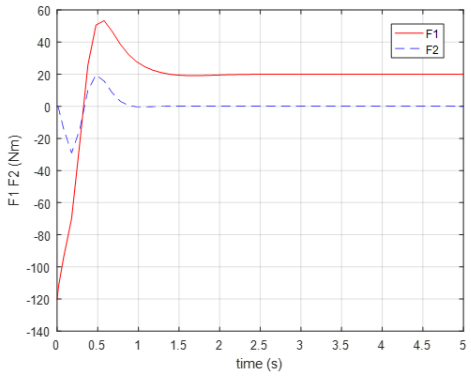


Figure 2.13: Robot torques

2.4 MPC Control and LQ Optimal Control of A Two-Link Robot Arm: A Comparative Study

2.4.1 Linear Quadratic Controller

In order to challenge the new control approach of a model predictive control, we have made a comparative study with the LQ optimal control approach and a control approach

proposed in the literature for the two-link robot arm[3]. The linear model given by the Eq.(2.12) is used to find linear quadratic (LQ) optimal control[87–89] and[90] for the two-link robot arm. The error e_i between the actual angle θ_i and the desired angle θ_{id} is defined as:

$$e_i = \theta_{id} - \theta_i \tag{2.45}$$

with $i = 1, 2$.

The desired angle θ_{id} $i = 1, 2$ is constant. Differentiating the Equation (2.45) twice, the following equation is obtained:

$$\ddot{e}_i = -\ddot{\theta}_i = -v_i \tag{2.46}$$

with $i = 1, 2$. let consider the follwing single decoupled linear system:

$$\ddot{e}_1 = -v_1 = v_1^* \tag{2.47}$$

The state space representation of the system (2.47) is given by:

$$\begin{cases} \dot{z} = Az + Bv_1^* \\ \psi = Cz \end{cases} \tag{2.48}$$

where:
$z = [z_1 z_2]^T = [e_1 e_2]^T \in \mathbb{R}^n$: is a state vector;
$\psi \in \mathbb{R}^m$: is the output vector;
v_1: is the synthetic control of the first joint of the robot;
A = $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, B = $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ And C = $\begin{bmatrix} 1 & 0 \end{bmatrix}$
If the objective cost function is considered:

$$J = \int_0^\infty (z^T Q z + v_1^{*T} R v_1^*) dt \tag{2.49}$$

where:

- $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is a symmetric positive semi-definite matrix,
- and R is a positive constant.

The objective cost function is minimized using the following linear quadratic optimal control:

$$v_1^* = -(R^{-1})^T P z(t) \tag{2.50}$$

where: P = $\begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}$ is the solution to the following so-called Riccati Equation:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{2.51}$$

The solution of the Riccati Equation (2.51) is:

$$P = \begin{bmatrix} \sqrt{2}\sqrt[4]{R} & \sqrt{R} \\ \sqrt{R} & \sqrt{2R\sqrt{R}} \end{bmatrix} \tag{2.52}$$

Therefore, the linear quadratic optimal control $v_1 = -v_1^*$ is given by the following control law:

$$v_1 = R^{-1}[P_2 z_1(t) + P_3 z_2(t)] = k_1 e_1(t) + k_2 \dot{e}_1(t) \tag{2.53}$$

where k_1=$R^{-1}\sqrt{R}$ and k_2=$R^{-1}\sqrt{2R\sqrt{R}}$.
Note that the control law for the second joint is obtained following the same steps.
Table (2.3)

R	1	1/100	1/150	1/200
τ_{1min}(Nm)	-21.15	-39.46	-44.88	-49.19
τ_{1max}(Nm)	26.28	82.83	96.95	108.85
τ_{2min}(Nm)	-1.21	-12.21	-14.75	-17.09
τ_{2max}(Nm)	1.76	17.80	21.48	25.03
$t^{\theta_1}_{r\pm5\%}$(s)	6.19	1.96	1.78	1.64
$t^{\theta_2}_{r\pm5\%}$(s)	6.19	1.96	1.78	1.64
$D_{\theta_1}(\%)$	13.56	13.56	13.56	13.56
$D_{\theta_2}(\%)$	13.56	13.56	13.56	13.56

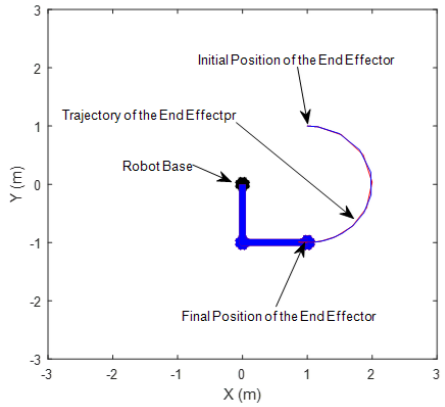
Table 2.3: System performance for different values of R

Comparing Tables 2.2 and 2.3, we have noticed that, the proposed model predictive control approach, provides a better system performance with out overshooting and fast asymptotic conversion of both joint variables to its desired positions was obtained compared with LQ optimal control approach. In addition, we have noticed as well that by using the MPC control and the LQ optimal control, a better system performance was obtained compared with PID control approach presented by David and Robles[3]. Furthermore, the mathematical development of the proposed LQ optimal control approach was only valid in the case where the desired angle was a constant or was ramp-like. The proposed MPC control approach, however, was valid for any desired angle.

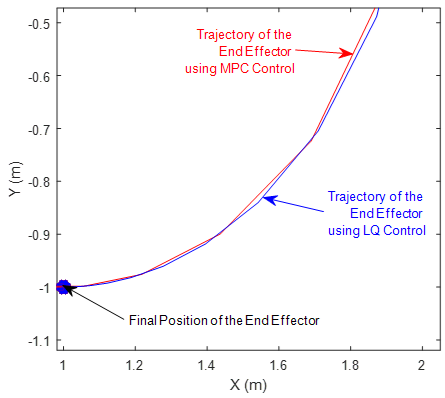
2.4.2 Comparison Study Between LQ and MPC Controllers

For comparison reason, we have maintained the same properties of the manipulator arm, but we have changed the horizon time h and the weight factor ρ of the MPC controller so that the maximum torque was strictly less than and as close as possible to 100 Nm, This condition was met in the case of the LQ optimal control approach for $R = 1/150$ with $max(|\tau|) \approx 97$ Nm.The same condition was met in the case of the MPC control approach for (h = 0.1961) (s) and (ρ = 0.0312) with $max(|\tau|) \approx 97$ Nm. which implies the MPC gains are :k_1 = 12.25 and k_2=6.30, while the LQ Gains are :k_1 = 12.24 and k_2=4.94.

Figure 2.14a represents the trajectory of the end effector of the robot arm with the final position of both joints depicted. Figure 2.14b presents a zoom image of the end-effector of the robot in its final position. The trajectory (full line) computed using the MPC control is observed arriving directly to the goal, unlike the trajectory (dotted-line) computed using the LQ control.

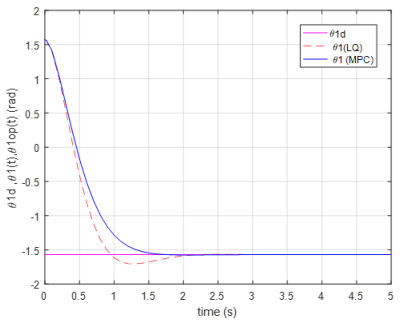


(a) Final position: the robot end-effector reaches its objective point

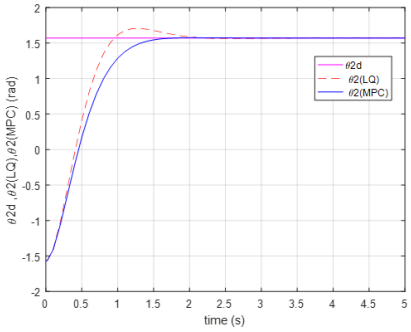


(b) Zoom image of the end-effector in its final position

Figure 2.14: Robot's end effector trajectories



(a) Real and the desired orientations of the second link of the robot



(b) Real and the desired orientations of the first link of the robot

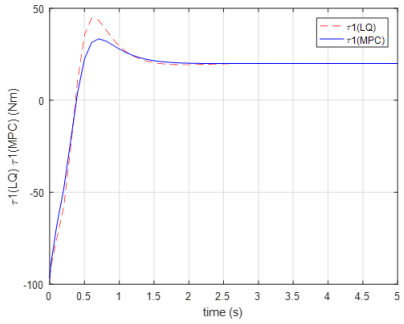
Figure 2.15: Comparison between linear quadratic (LQ) and model predictive control (MPC) controls of a real and reference orientations

Figure 2.15b , 2.15a represent the comparison between the convergence of the joint angles θ_1 and θ_2,respectively, to their reference values, using the MPC control and the LQ control. It is observed that the MPC control approach results in a fast and asymptotic convergence of both joint variables and without overshooting.

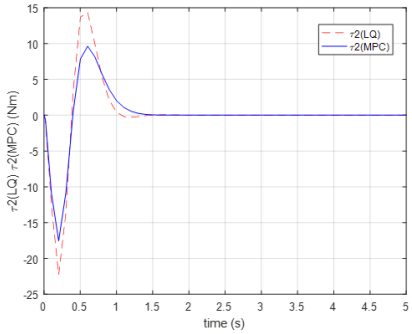
Figure 2.16a,2.16b show the comparison between the robot torques $\tau 1$ and $\tau 2$, respectively, that have been obtained, using the MPC control and the LQ optimal control. It is observed that by using the MPC control approach, the energy consumption is lower than by using the LQ optimal control.

Figure 2.17a,2.17b depict the robot synthetic controls v_1 and v_2 given by Equation (2.37). As can be seen, the synthetic controls reach zero when the end-effector of the robot reaches its objective.

The convergence of the joint-angle errors e_1 and e_2 of the two-link robot arm towards zero using the two proposed approaches of control are depicted in Figure 2.18a,2.18b, respectively.

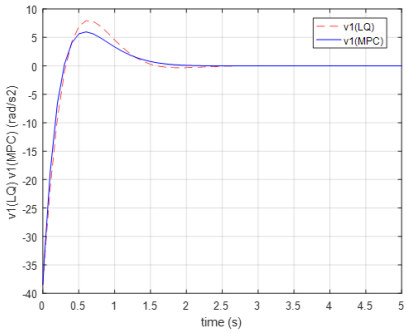


(a) Torques of the robot's first link

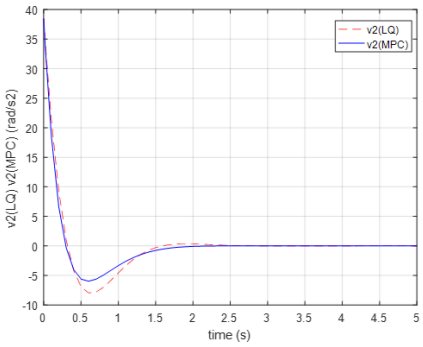


(b) Torques of the robot's second link

Figure 2.16: Comparison between LQ and MPC controls of the robot torques

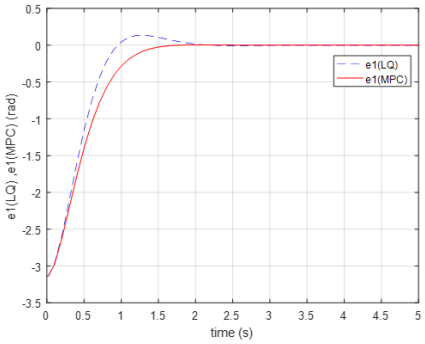


(a) Synthetic controls of the robot's first link

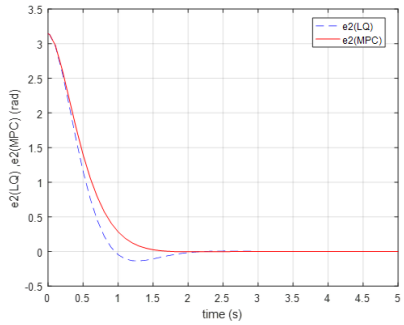


(b) Synthetic controls of the robot's second link

Figure 2.17: Comparison between LQ and MPC controls of the robot synthetic controls



(a) Error of the robot's first link



(b) Error of the robot's second link

Figure 2.18: Comparison of the errors in the joint angles, using LQ and MPC controls

2.4.3 Conclusion

In order to demonstrate the effectiveness of the proposed approach, we have performed a comparative study with the LQ control approach. From the results obtained and presented in the previous section, it can be stated that the proposed MPC control approach gives a better system performance than the LQ optimal control approach. In addition, both proposed approaches (MPC control and LQ control) give a better system performance than the PID control technique proposed by David and Robles[3].

2.5 Graphic base tuning of MPC controller for Three DOF Manipulator Arm

2.5.1 Three DOF Manipulator Arm Robot Description's

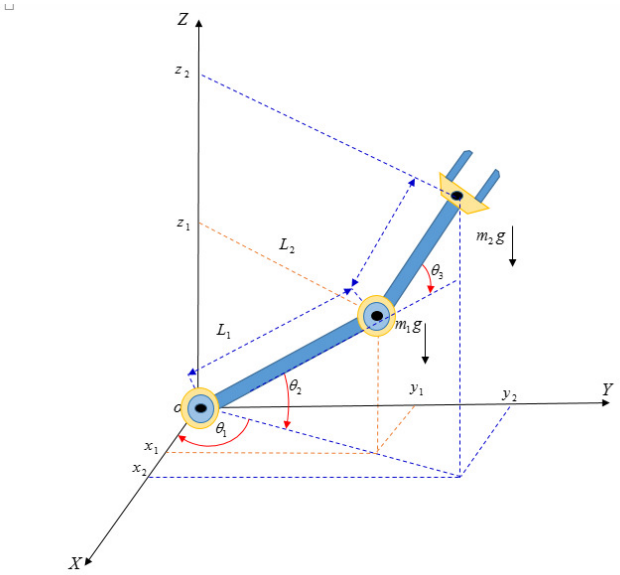


Figure 2.19: Three DOF Manipulator Robot

The manipulator robot shown in the Fig. 2.19 has three degrees of freedom, where $\theta_{i=1,2,3}$ are the joints angles, while $L_{i=1,2}$ and $M_{i=1,2}$ are respectively the length and the mass of the first (j=1) and the second (j=2) link of the robot. The variable g denote the gravity force.

2.5.2 Compute Of the Three DOF Manipulator Geometric Models

2.5.2.1 Compute Of the Forward Geometric Model

Based on the Fig. 2.19, we can calculate the coordinates of the end effector manipulator robot with respect to the frame (OXY) as follow:

$$\begin{cases} x(t) = [L_1 cos(\theta_2) + L_2 cos(\theta_2 + \theta_3)]cos(\theta_1), \\ y(t) = [L_1 cos(\theta_2) + L_2 cos(\theta_2 + \theta_3)]sin(\theta_1) \\ z(t) = L_1 sin(\theta_2) + L_2 sin(\theta_2 + \theta_3) \end{cases} \tag{2.54}$$

The equation (2.54) presents the geometric model which gives us the robot end effector coordinates based on its articulations values during its motion. We can also obtain the same equations by using Denavit-Hartenberg method as follow: The homogeneous transformation matrices of the three segments are given by:

$$T_{01} = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joints	DH Parameters			
n	θ_n	α_n	d_n	a_n
1	θ_1	$\pi/2$	0	0
2	θ_2	0	0	L_1
3	θ_3	0	0	L_2

Table 2.4: Denavit-Hartenberg representation for 3 DOF Manipulator Arm

$$T_{12} = \begin{bmatrix} C_2 & -S_2 & 0 & L_1C_2 \\ S_2 & C_2 & 0 & L_1S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{23} = \begin{bmatrix} C_3 & -S_3 & 0 & L_2C_3 \\ S_3 & C_3 & 0 & L_2S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{03} = T_{01} * T_{12} * T_{23} = \begin{bmatrix} C_{23}C_1 & -S_{23}C_1 & S_1 & [L_1C_2 + L_2C_{23}]C_1 \\ C_{23}S_1 & -S_{23}S_1 & -C_1 & [L_1C_2) + L_2C_{23}]S_1 \\ S_{23} & C_{23} & 0 & L_1S_2 + L_2S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The forward geometric model, which represent the coordinate of the end effector is also obtained as well from the fourth column of T_{03} as follow:

$$T_{03} = \begin{bmatrix} A_x & B_x & C_x & E_x \\ A_y & B_y & C_y & E_y \\ A_z & B_z & C_z & E_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where: $E = [E_xE_yE_z]^T$ is end effector coordinate vector, therefore:

$$\begin{cases} E_x = x(t) = [L_1C_2 + L_2C_{23}]C_1 \\ E_y = y(t) = [L_1C_2) + L_2C_{23}]S_1 \\ E_z = z(t) = L_1S_2 + L_2S_{23} \end{cases} \tag{2.55}$$

2.5.2.2 Compute Of the Inverse Geometric Model

From the Equation (2.55) we can rewrite it as follow:

$$\begin{cases} \frac{x_2(t)}{C_1} = L_1C_2 + L_2C_{23} \\ \frac{y_2(t)}{S_1} = L_1C_2 + L_2C_{23} \end{cases} \tag{2.56}$$

therefore :

$$\theta_1 = ATAN2(x_2(t), y_2(t)) \tag{2.57}$$

let suppose that:

$$\begin{cases} \frac{x_2(t)}{C_1} = x(t) \\ \frac{y_2(t)}{S_1} = y(t) \end{cases} \tag{2.58}$$

Consequently we can write the following equation:

$$\begin{cases} x(t) = L_1C_2 + L_2C_{23} \\ y(t) = L_1C_2 + L_2C_{23} \end{cases} \tag{2.59}$$

the equation (2.59) is similar to the two DOF forward geometric model, then θ_2 and θ_3 can be obtained as follows:

$$\theta_3 = ATAN2(\pm\sqrt{(1-\tau^2)}, \tau) \tag{2.60}$$

where:$\tau = (x(t)^2 + y(t)^2 - L_1^2 + L_2^2)/(2L_1L_2)$
and:

$$\begin{cases} cos(\theta_2) & = \frac{y(t)[L_1+L_2cos(\theta_2)]+x(t)[L_2sin(\theta_2)]}{x(t)^2+y(t)^2} \\ sin(\theta_2) & = \frac{x(t)[L_1+L_2cos(\theta_3)]-y(t)[L_2sin(\theta_3)]}{x(t)^2+y(t)^2} \end{cases} \tag{2.61}$$

therefor:

$$\theta_2 = ATAN2(sin\theta_2, cos\theta_2) \tag{2.62}$$

2.5.3 Compute Of the Three DOF Kinetic Models

2.5.3.1 Compute Of the Forward Kinetic Model

By deriving the forward geometric model of the three manipulator arm, the forward kinetic model is obtained as below:

$$\begin{cases} \dot{x}(t) = -[L_1C_2) + L_2C_{23})]S_1\dot{\theta}_1 - [L_1S_2) + L_2S_{23})]C_1\dot{\theta}_2 - L_2S_{23}S_1\dot{\theta}_3 \\ \dot{y}(t) = -[L_1C_2) + L_2C_{23})]C_1\dot{\theta}_1 - [L_1S_2) + L_2S_{23})]S_1\dot{\theta}_2 - L_2S_{23}S_1\dot{\theta}_3 \\ \dot{z}(t) = -[L_1C_2) + L_2C_{23})]\dot{\theta}_2 + L_2C_{23}\dot{\theta}_3 \end{cases} \tag{2.63}$$

where:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} -[L_1C_2) + L_2C_{23})]S_1 & -[L_1S_2) + L_2S_{23})]C_1 & -L_2S_{23}S_1 \\ -[L_1C_2) + L_2C_{23})]C_1 & -[L_1S_2) + L_2S_{23})]S_1 & -L_2S_{23}S_1 \\ 0 & -[L_1C_2) + L_2C_{23})] & L_2C_{23} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{2.64}$$

we can rewrite the kinematic model as below:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = J(\theta_i) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{2.65}$$

where:
(θ_i) is the Jacobian matrix and its formula is as follow:

$$J(\theta_i) = \begin{bmatrix} -[L_1C_2 + L_2C_{23}]S_1 & -[L_1S_2 + L_2S_{23}]C_1 & -L_2S_{23}S_1 \\ -[L_1C_2 + L_2C_{23}]C_1 & -[L_1S_2 + L_2S_{23}]S_1 & -L_2S_{23}S_1 \\ 0 & -[L_1C_2 + L_2]C_{23} & L_2C_{23} \end{bmatrix} with : i = 1, 2, 3 \tag{2.66}$$

2.5.3.2 Compute Of the Inverse Kinetic Model

The inverse kinetic model of three DOF manipulator arm if given via the Jacobin matrix inverse as follow:

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix} = J^{-1}(\theta_i)\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{z}(t)\end{bmatrix} with: i = 1,2,3 \tag{2.67}$$

where:

$$J^{-1}(\theta_i) = \frac{1}{\Delta}Adj[J(\theta_i)] \tag{2.68}$$

and:

$$\Delta = L_1L_2[L_1C_2S_3 - L_2S_2 + L_2C_3^2S_2 + L_2C_2C_3S_3] \tag{2.69}$$

Therefore the condition for the Jacobian inverse matrix to be exist is:

$$L_1C_2S_3 - L_2S_2 + L_2C_3^2S_2 + L_2C_2C_3S_3 \neq 0$$

The singularity positions for the above three dof manipulator robot are:

$$\begin{cases}\theta_3 = k\pi, k = 0, \pm1, \pm2, ...\\ L_1C_2 + L_2C_{23} = 0\end{cases} \tag{2.70}$$

$L_1C_2 + L_2C_{23} = 0 \Rightarrow x(t) = y(t) = 0,$

where the singularity positions are illustrated in the figures Fig.2.20, Fig.2.21 and Fig. 2.22.

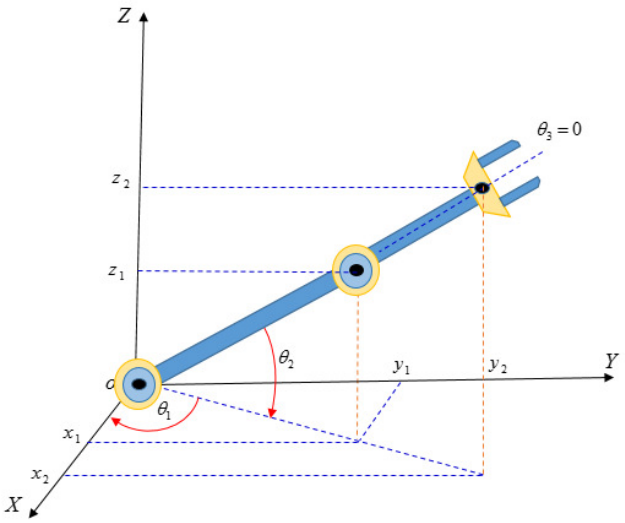


Figure 2.20: Three DOF Manipulator Robot singularity at $\theta_3 = 0$

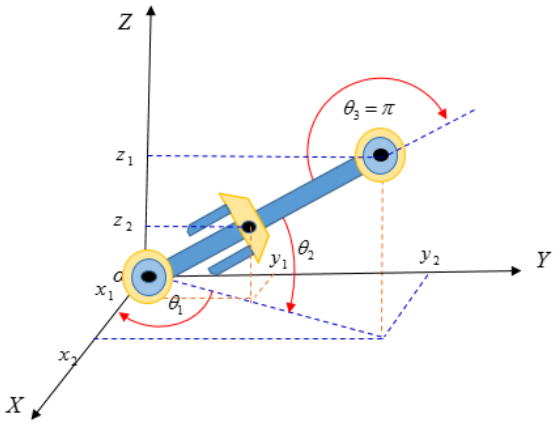


Figure 2.21: Three DOF Manipulator Robot singularity at $\theta_3 = \pi$

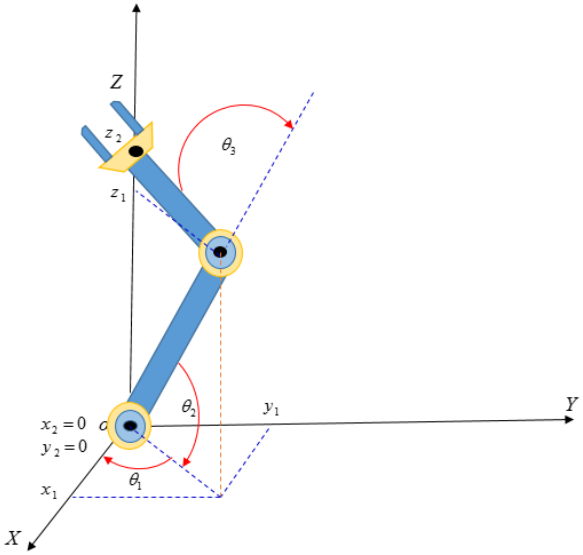


Figure 2.22: Three DOF Manipulator Robot singularity at xy=0

2.5.4 Compute Of the Dynamic Model

By using the equations (2.54) and (2.64) the total energy of the three DOF manipulator arm is given as below:

$$\begin{aligned} E_K = {} & \tfrac{1}{2}[M_1 L_1^2 C_2^2 + \tfrac{1}{2} M_2 (l_1 C_2 + l_1 C_{23})^2]\dot{\theta}_1^2 \\ & + \tfrac{1}{2}[M_1 L_1 + M_2 (L_1^2 + L_2^2 + 2 L_1 L_2 C_3)\dot{\theta}_2^2 \\ & + M_2 L_2^2 \dot{\theta}_3^2 + M_2 (L_2^2 + L_1 L_2 C_3)\dot{\theta}_2 \dot{\theta}_3 \end{aligned} \tag{2.71}$$

and the potential energy is given by the following formula:

$$E_P = g M_1 L_1 S2 + g M_2 (L_1 S_2 + L_2 S_{23}) \tag{2.72}$$

Finally the Lagrangian of the two dof manipulator robot is given as below:

$$\begin{aligned} L = & \tfrac{1}{2}[M_1 L_1^2 C_2^2 + \tfrac{1}{2} M_2 (l_1 C_2 + l_1 C_{23})^2]\dot{\theta}_1^2 \\ & + \tfrac{1}{2}[M_1 L_1 + M_2 (L_1^2 + L_2^2 + 2L_1 L_2 C_3)\dot{\theta}_2^2 \\ & + M_2 L_2^2 \dot{\theta}_3^2 + M_2 (L_2^2 + L_1 L_2 C_3)\dot{\theta}_2 \dot{\theta}_3 \\ & + g M_1 L_1 S2 + g M_2 (L_1 S_2 + L_2 S_{23}) \end{aligned} \tag{2.73}$$

By using the Lagrangian Formalism, it is easy to prove that the dynamic model of the three DOF manipulator arm Fig.2.19is given by the following formula:

$$\begin{cases} M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \\ Y = \theta \end{cases} \tag{2.74}$$

where:
$\theta = [\theta_1 \theta_2 \theta_3]^T$: is the joints position vector
$\tau = [\tau_1 \tau_2 \tau_3]^T$: is the joints torque vector
Y: is the output vector

$G(\theta) = [0 g_2 g_3]^T$: is the gravitational vector's

$C(\theta, \dot{\theta}) = [C_1 C_2 C_3]^T$: is the Coriolis and centrifugal forces vector's;

$M(\theta) = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & D_{23} \\ 0 & D_{32} & D_{33} \end{bmatrix}$: is the inertia matrix;

with:

$G_2 = M_1 g L_1 C_2 + M_2 g (L_1 C_2 + L_2 C_3)$

$G_3 = M_2 g L_2 C_{23}$

$C_1 = 2[M_1 L_1^2 S_2 C_2 + M_2 (L_1 S_2 + L_2 S_{23})]\dot{\theta}_1 \dot{\theta}_2 - 2M_2 L_2 S_{23} \dot{\theta}_1 \dot{\theta}_3$

$C_2 = -M_2 L_1 L_2 S_3 \dot{\theta}_3^2 - M_2 L_1 L_2 S_3 \dot{\theta}_2 \dot{\theta}_3 + [M_1 L_1^2 S_2 C_2 + M_2 (L_1 S_2 + L_2 S_{23})((L_1 C_2 + L_2 C_{23})]\dot{\theta}_1^2$

$C_3 = [(M_2 L_2 S_{23})(L_1 C_2 + L_2 C_{23})]\dot{\theta}_1^2 + M_2 L_1 L_2 S_3 \dot{\theta}_2^2$

$D_{11} = M_1 L_1^2 C_2^2 + M_2 (l_1 C_2 + l_1 C_{23})^2$

$D_{22} = M_1 L_1^2 + M_2 (L_1^2 + L_2^2 + L_1 L_2 C_3)$

$D_{23} = M_2 (L_2^2 + L_1 L_2 C_3)$

$D_{32} = D_{23}$

$D_{33} = M_2 L_2^2$

2.6 Three DOF Model Predictive Controller

2.6.1 Feedback Linearization Control

By using the same idea used for two DOF manipulator arm ,[83] and[84] we found as well, that the control input τ appears in the second derivative of the output Y. and the second derivative of Y is given by the following formula:

$$\ddot{Y} = \ddot{\theta} = M(\theta)^{-1}[-C(\theta, \dot{\theta})\dot{\theta} - G(\theta) + \tau] = v \tag{2.75}$$

where:
$v = [v_1 v_2 v_3]^T$: is the Synthetic Control Vector
Finally, from (2.75) we get the feedback linearization control as follow:

$$\tau = M(\theta)v + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{2.76}$$

the Fig. 2.23 illustrate the the feedback linearization control loop.

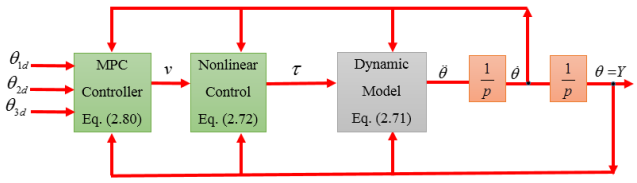


Figure 2.23: MPC and Feedback linearization Close-loop diagram

Applying the control law given by (2.76) to the nonlinear system (2.74), the dynamic model of the manipulator robot with three DOF, becomes a linear system double integrator. The relative degree is equal to two. This means that by using the control law (2.76), we obtain a complete linearization of the nonlinear system (2.74) and we get a linear system for each joint variable.

$$\begin{cases} \frac{\theta_1(p)}{v_1(p)} = \frac{1}{p^2} \\ \frac{\theta_2(p)}{v_2(p)} = \frac{1}{p^2} \\ \frac{\theta_3(p)}{v_3(p)} = \frac{1}{p^2} \end{cases} \tag{2.77}$$

where p is a Laplace variable. The linearization of the nonlinear system has been done. So,we can develop minimum time control for the three DOF robot arm which will be the aim of the next subsection

2.6.2 Model Predictive Control

In the case of a robot arm with three DOF and after application of the feedback linearization (2.76) to the nonlinear system (2.74), we obtain the following two decoupled linear

systems.

$$\begin{cases} \ddot{\theta}_1 = v_1 \\ \ddot{\theta}_2 = v_2 \\ \ddot{\theta}_3 = v_3 \end{cases} \tag{2.78}$$

Considering the first equation of (2.78). each equation from the above system can be rewritten in the state-space form:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = v_1(t) \\ y(t) = x_1(t) \end{cases} \tag{2.79}$$

where:

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \theta_1 & \dot{\theta}_1 \end{bmatrix}$$

v_1 is the synthetic control of the first link of the robot and Y is the output.
Now, we are going to develop a model predictive controller (MPC)[85], [86] for the first link of the robot arm. The MPC controller for the second and the third link of the robot will be developed in the same way as the first.
By Assuming $v_1(t) = v_1$ constant in the time interval [t t+h] , where h is horizon time of prediction. we can write the following equations as developed for the two-link robot arm:

$$\theta_1(t+h) = \frac{1}{2} v_1 h^2 + \dot{\theta}_1(t)h + \theta_1(t) \tag{2.80}$$

the desired angle of the first joint θ_{1d} is constant as well as the second and the third joint. We want to minimize not only the articulation deviation error but also the energy needed by the manipulator arm to reach the final position, the reason why we choose the following cost function:

$$J = e_1^2(t+h) + \rho \int_t^{t+h} v_1^2 dt \tag{2.81}$$

,

where: $e_1(t+h)$ is the predicted joint position error,v_1 is the synthetic control and:

$$e_1(t+h) = \theta_{1d} - \theta_1(t+h) \tag{2.82}$$

The horizon time h and the weight factor ρ are both positive parameters to be optimized later.
By substituting the prediction model(2.82) into (2.81) the quadratic cost function will be:

$$J = (\frac{1}{4}h^4 + \rho h)v_1^2 - (\theta_{1d} - \theta_1 - \dot{\theta}_1 h)h^2 v_1 + (\theta_{1d} - \theta_1 - \dot{\theta}_1 h)^2 \tag{2.83}$$

By minimizing the criterion J with respect to v_1, the new approach of the model predictive controller is as follow:

$$v_1(t) = \alpha_1(\theta_{1d} - \theta_1(t)) - \alpha_2\dot{\theta}_1(t) \tag{2.84}$$

where the control law gains are as follow:

$$\begin{cases} \alpha_1 = \frac{2h}{h^3+4\rho} \\ \alpha_2 = \frac{2h^2}{h^3+4\rho} \end{cases} \tag{2.85}$$

The block diagram of the closed-loop system is presented as depicted in 2.24:

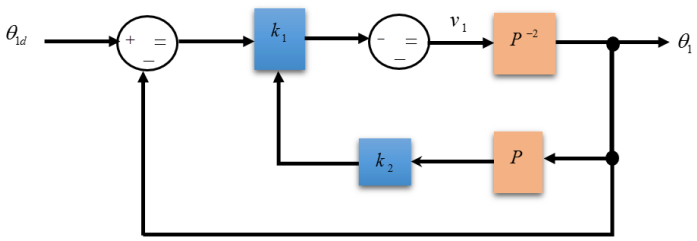


Figure 2.24: Three DOF Close Loop Transfert Function

Regarding the response of the system, we would like to have a behavior similar to a system of the second order:

$$\frac{\omega_0^2}{p^2 + 2\zeta\omega_0 p + \omega_0^2} \tag{2.86}$$

where ζ is a damping factor and ω_0 is a natural frequency.
The transfer function of the system presented in Fig 2.24 is given as below:

$$\frac{\theta_1(p)}{\theta_{1d}(p)} = \frac{\alpha_1}{p^2 + \alpha_2 p + \alpha_1} \tag{2.87}$$

Taking into account (2.85),(2.86) and (2.87) we obtain:

$$2\zeta\omega_0 = \frac{2h^2}{h^3 + 4\rho} \tag{2.88}$$

$$\omega_0^2 = \frac{2h}{h^3 + 4\rho} \tag{2.89}$$

based on equation.2.89 we can easly prove that:

$$h = \frac{2\zeta}{\omega_0} \tag{2.90}$$

and:

$$\rho = \frac{\zeta}{\omega_0^3}(1 - 2\zeta^2) \tag{2.91}$$

The weight factor ρ is a positive number, therefore from the equations (2.89) and (2.90), the damping factor ζ must be less than $\frac{1}{\sqrt{2}}$. Hence, in order to get a good damping, we have to choose the value of ζ to be as near as possible to $\frac{1}{\sqrt{2}}$, let suppose : $\zeta = \frac{0.999}{\sqrt{2}}$

2.6.3 Optimization and Constraints Handling based on graphics creteria

The aim of this section is to find h and ρ values in such way our system fulfills the following conditions:

$$\begin{cases} |\tau_i| \leq 100Nm \\ T_{r\pm 5\%} \leq 1.7S \\ i = 1, 2, 3 \end{cases} \tag{2.92}$$

The idea is to draw the maximum, the minimum torque values (for each arm articulations) as well as the settling time values for several values of ω_0 (from 1 rad/s up to 10 rad/s) (see Fig.: 2.25, 2.26 and 2.27. Then it will be very easy to find a trade-off which satisfy the above constraints conditions. Note that the the joints torques are given by the equation (2.75) while the setting time is given by the following formula[91]:

$$h = -\frac{\ln(0.05\sqrt{1-\zeta^2})}{\zeta\omega_0} \tag{2.93}$$

From Fig. 2.25, in order to have a maximum torque less than or equal to 100 Nm, the natural frequency ω_0 should be less than or equal to 2.8 rad/s, and from Fig. 2.26, in order to have a minimum torque greater than or equal to -100 Nm, the natural frequency ω_0 should be less than or equal to 5.3 rad/s. which means ω_0 should be less or equal to 2.8rad/s. From Fig. 2.27. To get a settling time $T_{r5\%}$ less than or equal to 1.7 s, the natural frequency ω_0 should be greater than or equal to 2.7 rad/s. Finally the natural frequency has to verify the following inequality:

$$2.7rad/s \leq \omega_0 \leq 2.8rad/s \tag{2.94}$$

Let's take ω_0=2.79 rad/s, Using the equations. (2.90) and (2.91), the prediction horizon time h and the weight factor ρ are:$h = 0.5064$ and $6.5021 * 10^{-5}$. Using the equation system's (2.85), the values of the predictor control law gains are: $\alpha_1 = 7.78$ and α_2=3.94.

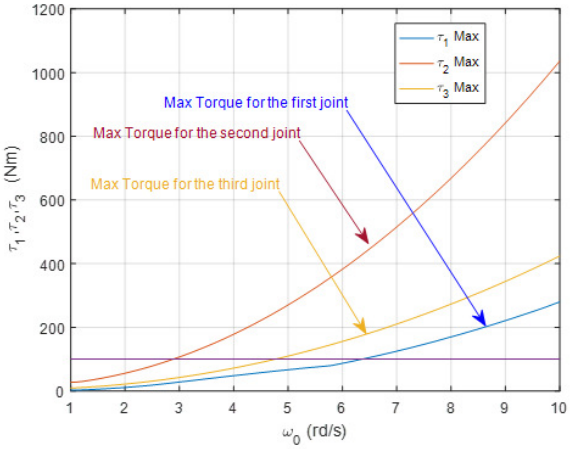


Figure 2.25: The variation of the maximum torques as a function of ω_0

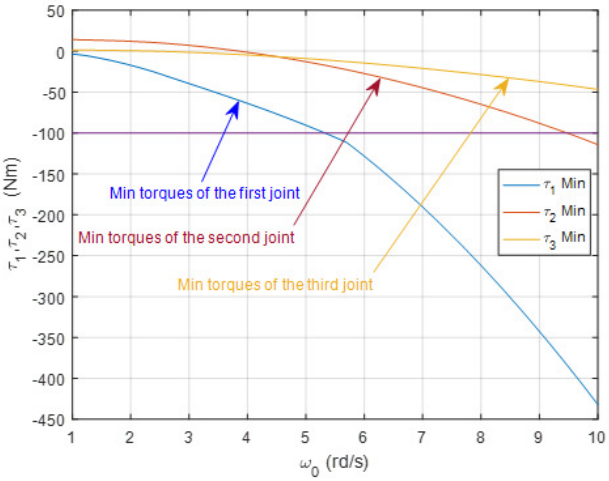


Figure 2.26: The variation of the minimum torques as a function of ω_0

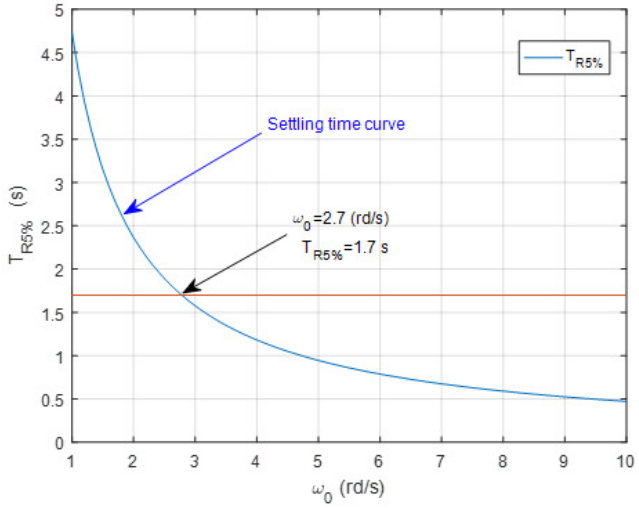


Figure 2.27: The variation of the settling time as a function of ω_0

2.6.4 Simulation and Results

In order to show the efficiency of the proposed graphical based tuning for the MPC controller.For simulation purpose, we assume that the mass and the length of the first and the second links of the robot arm are $M_{i=(1,2)} = 1$ (Kg) and $L_{i=(1,2)} = 1$ (m), respectively. The initial and the desired orientations of the first and the second links of the robot arm are $\theta_1(0) = \pi/4$, $\theta_2(0) = \pi/3$,$\theta_3 = 0$,$\theta_{1d} = -\pi/2$,$\theta_{2d} = \pi/4$and $\theta_{3d} = \pi/6$, respectively.

Fig.2.28 shows the manipulator arm end effector trajectory from the initial conditions to the desired configurations. Fig.2.29,2.30 and 2.31 illustrate the convergence of the joint angles θ_1,θ_2 and θ_3 ,respectively, to their reference values. with settling time $T_{r5\%}$ less than 1.7 s. In Fig. 2.33 the robot synthetic controls v_1, v_2 and v_3 given by the Eq. (2.84) are shown. As we can see, the synthetic controls reach zero when the end effector of the robot reaches its objective. Fig. 2.34 shows the robot torques that can be obtained from synthetic controls using Eq. (2.84). The angles errors e_1,e_2 and e_3 of the three-link of the robot arm converges towards zero using the proposed approach of control are depicted in Fig. 2.32.

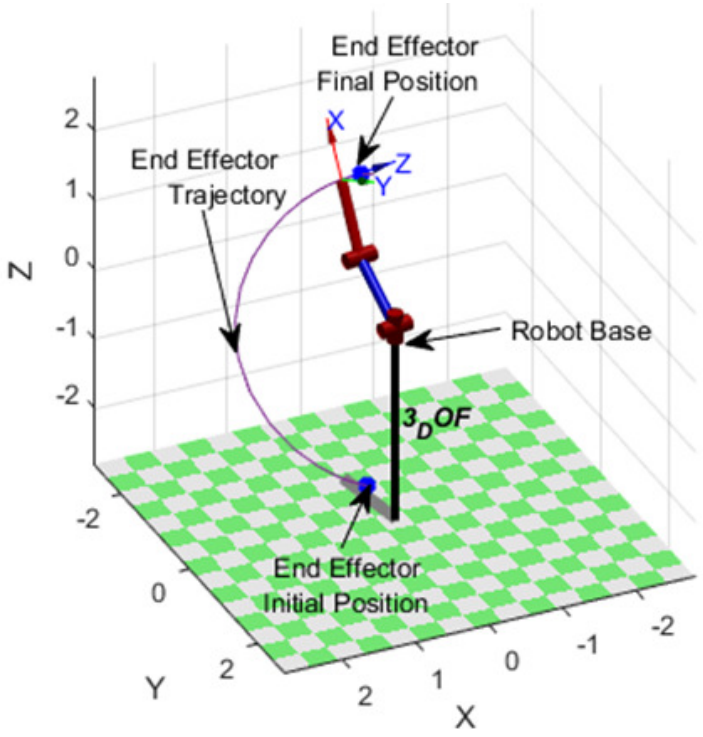


Figure 2.28: Arm manipulator end-effector in the final position

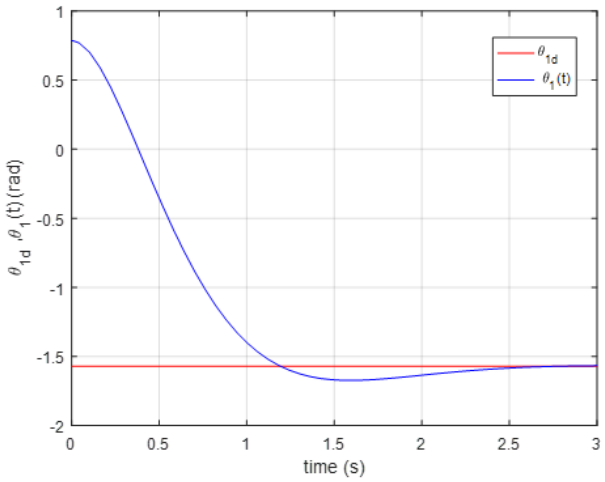


Figure 2.29: Real and the desired orientation of the first link of the 3 DOF robot

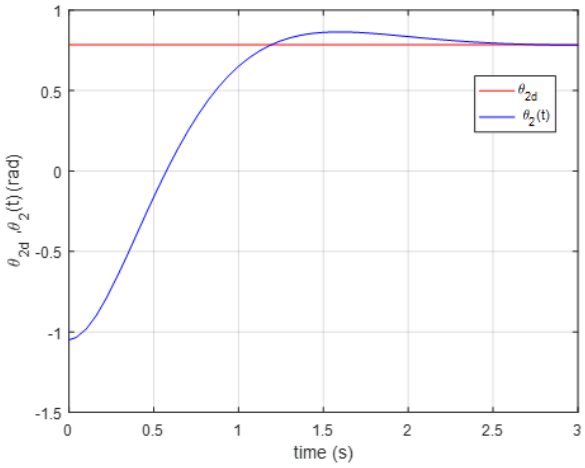


Figure 2.30: Real and the desired orientation of the second link of the 3 DOF robot

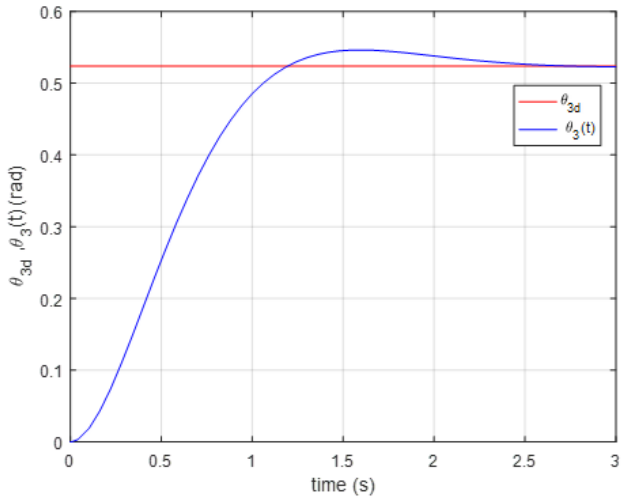


Figure 2.31: Real and the desired orientation of the third link of the 3 DOF robot

2.6.5 Conclusion

A new MPC control approach for a two-link robot arm with two degrees of freedom is proposed in this chapter. The technique consisted of linearizing a nonlinear dynamic model of the robot by using a feedback linearization control. Next, based on the obtained linear model, an MPC controller was developed that was tuned by choosing the parameters h and ρ. In order to demonstrate the effectiveness of the proposed approach, we performed a comparative study with the LQ control approach. From the results obtained and presented in this chapter, it can be stated that the proposed MPC control approach gives a better system performance than the LQ optimal control approach. In addition, both proposed approaches (MPC control and LQ control) give a better system performance than the PID control technique proposed by[3]. In this chapter we have also developed an MPC control for controlling a 3 DOF manipulator robot. The simulation results obtained show the effectiveness of the proposed control approach. The research carried out in this chapter has been published in an international journal and in two IEEE international

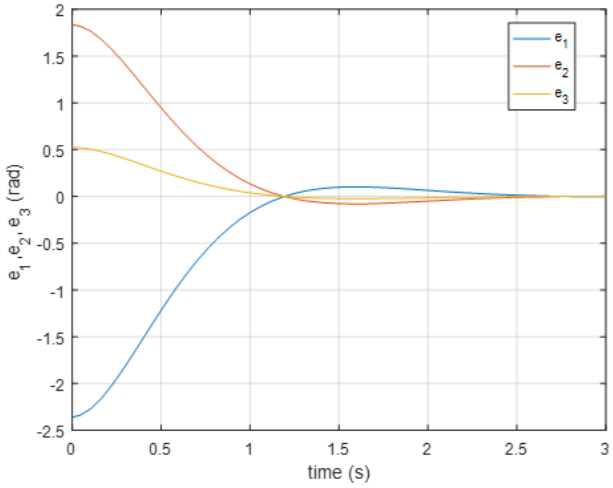


Figure 2.32: Errors in the joint angle

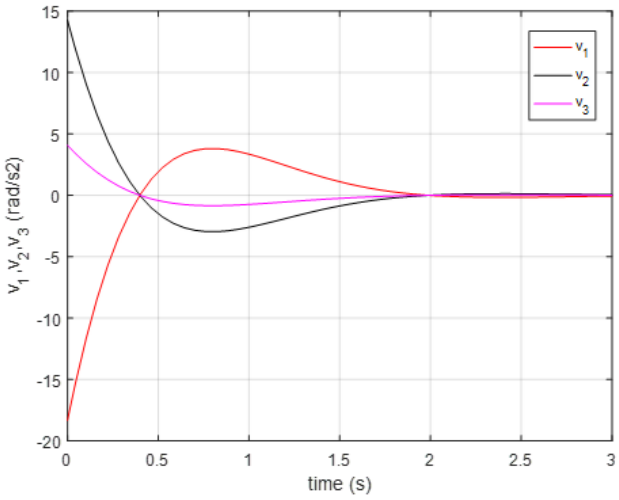


Figure 2.33: Robot synthetic controls

conference[45], [46] and[47].

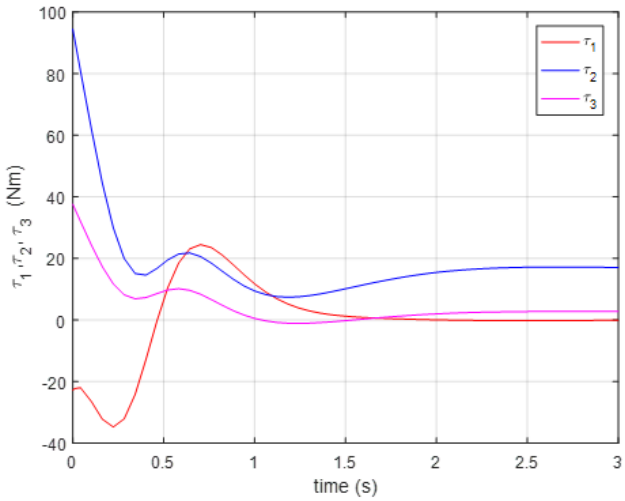


Figure 2.34: Robot torques

Chapter 3

Differential-Drive Mobile Robot Control

3.1 Introduction

Today the articulated robot arm constitute a common auxiliary machine used for wide range of applications such as manipulation e.g. dangerous materials, chemical substances, explosive materials e.g. welding, painting or assembling. These robots are constructed by a revolute joints and a rigid arms, and the manipulator robot workspace can only be defined based on the number of degrees of freedom (DOF).To extend the manipulator robot workspace, the manipulator robot is fixed to a mobile platform allowing the motion in additional axis, and create a mobile manipulator robot[35,36,45,92–97].
Our main contribution of this section is to apply a model predictive control for controlling a differential drive mobile robot and use the wheels torques and settling time graphs as a tool to better tune predictive controller gains. Finally, we tried to find a compromise between the dynamic performance and the required energy needed to perform any requested task from the DDMR.

3.1.1 Mobile Base Classifications

Wheeled Mobile Robots can be classified according to their drive system to:
1- Differential drive WMRs:A differential wheeled robot is a mobile robot whose movement is based on two separately driven wheels placed on either side of the robot base. Its direction can be changed by varying the relative rate of rotation of its wheels and hence does not require an additional steering motion. For balance reason, additional wheels may be required.
2- Car-type WMRs: A car-type mobile robot consists of a two drive wheels at the rear of the main body and two differential steering wheels at the front[98].
3- Omnidirectional WMRs: Omnidirectional robots are capable to heading for any direction[99],[100]. They can change direction without changing the position or orientation of robots, thus being capable of easily performing tasks in congested environments when facing static and dynamic obstacles and narrow aisles.
4- Synchro drive WMRs: In the synchro-drive (synchronous drive) mobile robot, the wheels are rotated together, and therefore they always point to the same driving direction (all wheels steer and drive synchronously). All wheels turn and drive in unison[101],[102] and[103].The mobile base does not rotate as the wheels steer; thus, it remains in the same

orientation regardless of its direction of movement. This can be done by using a single motor and a chain for steering and a single motor for driving all wheels[104].

3.2 DDMR Geometric Description

Fig. 3.1shows the differential drive mobile robot subject of our study.

Figure 3.1: 3D View of a mobile manipulator robot with 3DOF

3.2.1 Coordinates Systems

In order to describe the position of the mobile manipulator robot in its environment, we have defined the following different coordinate systems:

1. Global Coordinate System: This coordinate system is a global frame which is fixed in the environment or plane in which the DDMR moves in. Moreover, this frame is considered as the reference frame and is denoted as (X_g, Y_g) .

2. Robot Coordinate System: This coordinate system is a local frame attached to the DDMR, and thus, moving with it. This frame is denoted as (X_r, Y_r) .

The two defined frames are shown in Fig.3.2 the origin of the robot frame is defined to be the mid-point A on the axis between the wheels. The center of mass C of the robot is assumed to be on the axis of symmetry, at a distance d from the origin A. The mobile robot position and orientation in the global frame can be defined as:

$$q^g = |x_a y_a \theta|^T \tag{3.1}$$

3.2.2 DDMR Non-holonomic Constraints

Before discussing the non-holonomic constraints, let put a clear definition about it, and keep it in our mind as we advance in our development and get the finalized dynamic model of the DDMR. Holonomic or omnidirectional mobile robot is a robot that has three degrees of freedom:
- A translation along X (T_x) moves forward or backward

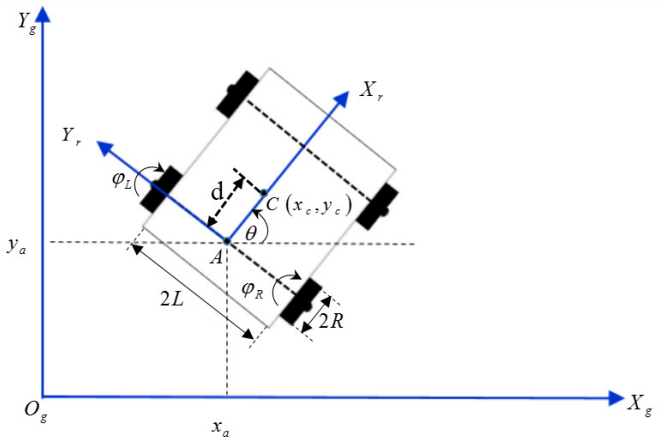


Figure 3.2: Unicycle mobile robot

- A translation along Y (T_y) goes to the right or to the left
- A rotation along Z (T_z) turns to the right or to the left

Any movement performed in a plane can be broken down into a maximum of two pure

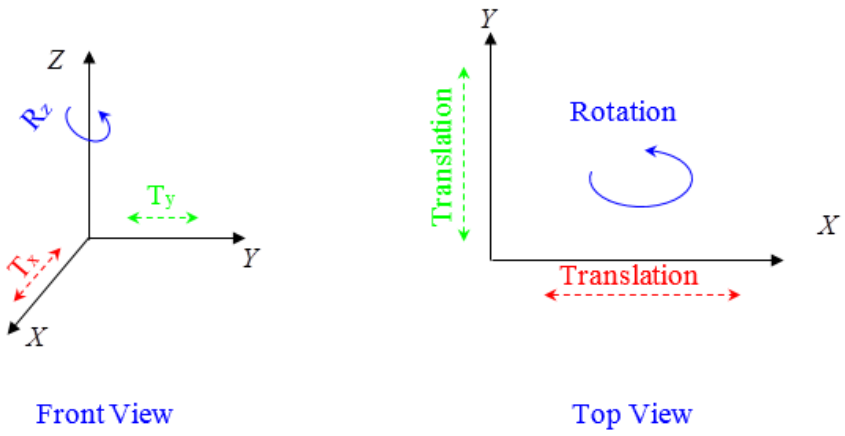


Figure 3.3: Three Degree Of Freedom Mobile Base

translational movements performed respectively with respect to the X and Y axes, plus one pure rotational movement around the axis perpendicular to the plane as illustrated in Fig. 3.3. The holonomic robot therefore is able to move in any direction whatever its orientation.

The holonomic kinematic constraints contains positional parameters without its derivatives with respect to time can be written under the following form:

$$\Lambda\left(x_i, y_i, z_i, q_i\right) = 0 \tag{3.2}$$

The robots which have less than three DOF are non-holonomic robots.

The DDMR motion is characterized by the non-holonomic constraints equations, which are based on the following assumptions[71]:

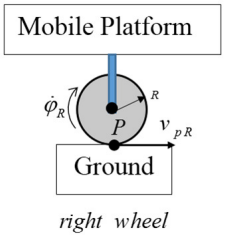


Figure 3.4: Rolling motion constraints

- No lateral slip:

$$-x_a \cos\theta + y_a \sin\theta = 0 \tag{3.3}$$

-Pure rolling:

$$\begin{cases} \dot{x}_{PR}\cos\theta + \dot{y}_{PR}\sin\theta = R\dot{\phi}_R \\ \dot{x}_{PL}\cos\theta + \dot{y}_{PL}\sin\theta = R\dot{\phi}_L \end{cases} \tag{3.4}$$

And:

$$\begin{cases} \dot{x}_{pR} = \dot{x}_{pL} = \dot{x}_a + L\dot{\theta}\cos\theta \\ \dot{y}_{pR} = \dot{y}_{pL} = \dot{y}_a + L\dot{\theta}sin\theta \end{cases} \tag{3.5}$$

where: $\dot{\phi}_R$ and $\dot{\phi}_L$ are the right and left wheels angular velocities respectively, R is the wheels radius, L is the distance between the driving wheels and the axis of symmetry and d is the distance between the points A and C as illustrated in Fig. 3.2 and Fig. 3.4.
Taking Eq. (3.5) into account, Eq.(3.4) becomes the following:

$$\begin{cases} \dot{x}_a\cos\theta + \dot{y}_a\sin\theta + L\dot{\theta} - R\dot{\phi}_R = 0 \\ \dot{x}_a\cos\theta + \dot{y}_a\sin\theta - L\dot{\theta} - R\dot{\phi}_L = 0 \end{cases} \tag{3.6}$$

The constraint equations (3.3) and (3.4) can be presented as follow:

$$\Lambda(q)\,\dot{q} = 0 \tag{3.7}$$

where:

$$\Lambda(q) = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & L & -R & 0 \\ \cos\theta & \sin\theta & -L & 0 & -R \end{bmatrix} \tag{3.8}$$

And:

$$\dot{q} = \begin{bmatrix} \dot{x}_a & \dot{y}_a & \dot{\theta} & \dot{\phi}_R & \dot{\phi}_L \end{bmatrix}^T \tag{3.9}$$

3.3 DDMR Models

3.3.1 Kinematic Model

Kinematic modelling is the study of the motion of mechanical systems without considering the forces that affect the motion ; we can calculate the linear and angular velocities

for the driving wheels in the robot frame as below:

$$\begin{cases} v = \frac{v_{pR}+v_{pL}}{2} \\ w = \frac{v_{pR}-v_{pL}}{2L} \end{cases} \tag{3.10}$$

where v_{pR} and v_{pL} Are the right and left linear velocities of the contact point P.
Therefore, the platform centre A velocities in the robot and global frames are as follows:

$$\dot{q}^r = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ w \end{bmatrix} = \frac{R}{2} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix}, \tag{3.11}$$

And:

$$\dot{q}^g = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ w \end{bmatrix} = \frac{R}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.12}$$

3.3.2 Dynamic Model

The study of mechanical system motion taking into consideration the different forces that affect it. The dynamic model of the DDMR is essential for simulation analysis of its motion and for the design of various motion control algorithms using the Lagrange formula:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) + \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)\,\lambda \tag{3.13}$$

where:

$$L = T - V \tag{3.14}$$

T is the kinematic energy, V is the potential energy of the mobile platform, q_i are the generalized coordinates, F is the generalized force vector,$\wedge$ is the constraint matrix, and λ is the Lagrange multiplier vector associated with the constraints. In addition, knowing the DDMR kinematic energy, which is the sum of T_c , the kinematic energy of the robot platform without wheels, plus T_{wR},T_{wL} , the kinematic energy of the wheels, note that their formulas are as follows[71]:

$$\begin{cases} T_c = \frac{1}{2}m_c v_M^2 + \frac{1}{2}I_c\dot{\theta}^2 \\ T_{wR} = \frac{1}{2}m_w v_{wR}^2 + \frac{1}{2}I_m\dot{\theta}^2 + \frac{1}{2}I_w\dot{\phi}_R^2 \\ T_{wL} = \frac{1}{2}m_w v_{wL}^2 + \frac{1}{2}I_m\dot{\theta}^2 + \frac{1}{2}I_w\dot{\phi}_L^2 \end{cases} \tag{3.15}$$

where m_c and I_c are the mass and the moment of inertia of the platform without the driving wheels, respectively, and m_w and I_w are the mass and the moment of inertia of each driving wheel plus the rotor of its motor.
The dynamic model of the non-holonomic DDMR with n generalized coordinates $(q_1, q_2, ..., q_n)$ and subject to m constraints can be described by the following equation of motion[1]:

$$M(q)\,\ddot{q} + V(q,\dot{q}) = B(q)\tau - \Lambda^T(q)\,\lambda \tag{3.16}$$

where:
$M(q)$: The inertia moment matrix, symmetric positive definite matrix.

$V(q,\dot{q})$:The coriolis matrix.
$B(q)$:Input matrix.
τ:Input vector.
In addition:

$$M(q)=\begin{bmatrix} m_T & 0 & -m_T d\sin\theta & 0 & 0 \\ 0 & m_T & m_T d\cos\theta & 0 & 0 \\ -m_T d\sin\theta & m_T d\cos\theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \tag{3.17}$$

$$V(q,\dot{q})=\begin{bmatrix} m_T & -m_T d\dot{\theta}\cos\theta & 0 & 0 & 0 \\ 0 & -m_T d\dot{\theta}\sin\theta & m_T d\cos\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.18}$$

$$B(q)=\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}^T \tag{3.19}$$

Note that:

$$\begin{cases} m_T = m_c + 2m_w, \\ I = I_c + m_c d^2 + 2m_w L^2 + 2I_m \end{cases} \tag{3.20}$$

For the purpose of control and simulation and because the Lagrange multipliers λ_i are unknown, it is more convenient to eliminate the constraint term $\Lambda(q)^T\lambda$ in Eq.(3.16). Furthermore, since the constrained velocity is always in the null space of $\Lambda(q)$, it is possible to define (n-m=2) velocities $\eta(t)=[\eta_1\eta_2]$, such that[1]:

$$\dot{q}=S(q)\,\eta(t) \tag{3.21}$$

We can verify that $S(q)$ is also the null space of the constraint matrix λ_i , which means:

$$S(q)\,\Lambda(q)=0 \tag{3.22}$$

then, it is easy to verify that:

$$S(q)=\begin{bmatrix} \frac{R}{2L}(L\cos\theta-d\sin\theta) & \frac{R}{2L}(L\cos\theta+d\sin\theta) \\ \frac{R}{2L}(L\sin\theta+d\cos\theta) & \frac{R}{2L}(L\sin\theta-d\cos\theta) \\ \frac{R}{2L} & -\frac{R}{2L} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.23}$$

Therefore based on $S(q)$ matrix choice, and the state variable

$$q=\begin{bmatrix} x_a & y_a & \theta & \phi_R & \phi_L \end{bmatrix}^T$$

we have

$$\eta(t)=[\dot{\phi}_R\dot{\phi}_L]$$

By differencing the Eq.(3.22) and substituting the expression of $\ddot{q}$ into Eq.(3.16) and multiplying it by S^T, The dynamic model equation is as follow[71]:

$$\bar{M}\dot{\eta} + \bar{V} = \bar{B}\tau = \tau \tag{3.24}$$

Such as:

$$\begin{cases} \bar{M} = S^T(q)M(q)S(q) \\ \bar{V} = S^T(q)\left(M(q)\dot{S}(q) + V(q,\dot{q})\right) \\ \bar{B} = S^T(q)B(q) = I_{2\times 2} \end{cases} \tag{3.25}$$

The Eq. (3.24), shows that the DDMR dynamic model is a function only of the right and left wheel angular velocities($\dot{\phi}_R, \dot{\phi}_L$) , the robot angular velocity $\dot{\theta}$ and the driving motor torques (τ_R, τ_L) .

3.4 Control Algorithm

In this part, we will develop a predictive control law for the differential-drive mobile robot. First, we consider the nonlinear dynamic model given by Eq. (3.24) Then, we convert the nonlinear dynamic model into a completely linear model in which we can apply our linear control approach. Once the linear model has been obtained, a model predictive control will be designed as the second step.

3.4.1 Input-Output Linearization

Let consider the state space vector below[1]:

$$x = [q^T \eta^T]^T = [x_c, y_c, \theta, \phi_R, \phi_L, \dot{\phi}_R, \dot{\phi}_R]^T \tag{3.26}$$

We can write the state space dynamic model as follows:

$$\dot{x} = \begin{bmatrix} S\eta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{M}^{-1}\left(\tau - \bar{V}\right) \end{bmatrix} = \begin{bmatrix} S\eta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{2x2} \end{bmatrix} \bar{M}^{-1}\left(\tau - \bar{V}\right) \tag{3.27}$$

We can rewrite the above state space equation as follow:

$$\dot{x} = f(x) + g(x)\ u \tag{3.28}$$

where:

$$\begin{cases} u = \bar{M}^{-1}\left(\tau - \bar{V}\right) \\ f(x) = \begin{bmatrix} S\eta \\ 0 \end{bmatrix} \\ g(x) = \begin{bmatrix} 0 \\ I_{2\times 2} \end{bmatrix} \end{cases} \tag{3.29}$$

From the Eq. (3.28), we could assume a new input u which could linearize (3.27), but the challenge now is to find another equation that links the new input u with the robot's position in such a way that we can compute the wheels' torques based on this new input.

Let us suppose $y = \Upsilon(q) = \begin{bmatrix} x_c & y_c \end{bmatrix}^T$ as the output signals that we need to control, as shown in Fig.**??**:

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_a \\ y_a \end{bmatrix} + \begin{bmatrix} x_c^a & -y_c^a \\ x_c^a & y_c^a \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{3.30}$$

x_c^a, y_c^a are the coordinates of point C in the robot frame system. Therefore:

$$\dot{y} = {}^{\partial\Upsilon}/_{\partial t} = \left({}^{\partial\Upsilon}/_{\partial q}\right)\dot{q} = J_f[S(q)\eta(t)] = [J_f S(q)]\eta(t) = \Gamma(q)\eta(t) \tag{3.31}$$

where: Γ : 2x2 decoupling matrix, such as :

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \tag{3.32}$$

where:

$$\begin{bmatrix} \Gamma_{11} \\ \Gamma_{12} \\ \Gamma_{21} \\ \Gamma_{22} \end{bmatrix} = \frac{R}{2L} \begin{bmatrix} L - y_c^a & -d - x_c^a \\ L + y_c^a & d + x_c^a \\ L - y_c^a & d + x_c^a \\ L + y_c^a & -d - x_c^a \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{3.33}$$

The second derivate of Eq.(3.31), is obtained as follows:

$$\ddot{y} = \dot{\Gamma}\eta + \Gamma\dot{\eta} \tag{3.34}$$

As developed in[1], to make Eq. (3.34) linear we have to do the following substitution:

$$\begin{cases} \ddot{y} = v \\ \dot{\eta} = u \end{cases} \tag{3.35}$$

which is the I/O feedback linearisation, therefore the second input u will be:

$$v = \dot{\Gamma}\eta + \Gamma u \Rightarrow u = \Gamma^{-1}(v - \dot{\Gamma}\eta) \tag{3.36}$$

By substituting the second input u of the Eq.(3.36) into Eq. (3.24), we can compute the DDMR wheels' torques as follows:

$$\tau = \bar{M}u + \bar{V} \tag{3.37}$$

Now the only variable to determine is v, note that:

$$\ddot{y} = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{3.38}$$

Where: $v = [v_1, v_2]$ is the synthetic control vector. Which will be the subject of the following subsection.
The figure 3.5 presents an overall control loop of our DDMR:

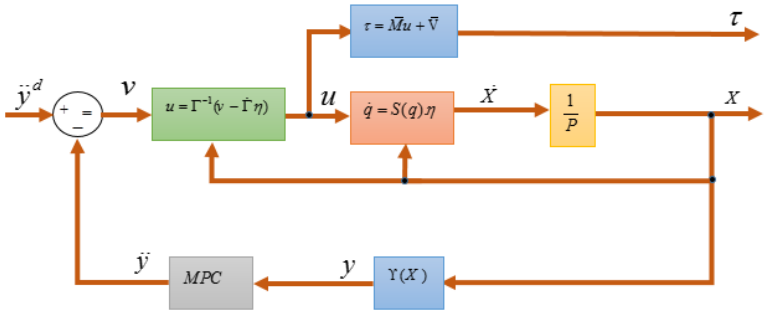


Figure 3.5: DDMR-Close-Loop

3.4.2 Model Predictive Control Law

Following the I/O feedback linearization, let us apply the model predictive control to compute the first input v_1.
where v_i is the synthetic control vector. Now let us develop the predictive control law for the x_c coordinate, which will be similar to the y_c coordinate.

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = v_1(t) \\ z(t) = x_1(t) \end{cases} \tag{3.39}$$

Where:
$\begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} x_c & \dot{x}_c \end{bmatrix}^{\mathrm{T}}$, $v_1$is the synthetic control of the x_c variable, while $z(t) = x_c(t)$ is the output signal.
Let consider that $v(t) = v_1$ is constant in the time interval $[t\ t+h]$[85],[105],[86], where h is the prediction horizon time, and by using the Eq.(3.39), we can formulate the prediction model as follows:

$$x_c(t+h) = \frac{1}{2}v_1h^2 + \dot{x}_c(t)h + x_c(t) \tag{3.40}$$

We want to minimize not only the articulation deviation error but also the energy needed by the manipulator arm to reach the final position, which is why we choose the following cost function[48]:

$$J = e_1^2(t+h) + \rho \int_t^{t+h} v^2 dt \tag{3.41}$$

where: $e_1(t+h) = x_{cd} - x_c(t+h)$ is the prediction error and x_{cd} is the desired value generated by the referenced trajectory. The horizon time h and the weight factor ρ are both positive

parameters to be computed later. By replacing the predicted value of $x_c(t+h)$ given by Eq. (3.40) into Eq.(3.41), therefore the criterion J becomes:

$$\begin{aligned} J = {} & x_{cd}^2 - x_{cd} v_1 h^2 - 2h x_{cd} \dot{x}_c(t) - 2 x_{cd} x_c(t) + \tfrac{1}{4} v_1^2 h^4 + \\ & v_1 h^3 x_{cd} + v_1 h^2 x_c(t) + \dot{x}_c^2(t) h^2 + 2h \dot{x}_c(t) x_c(t) + x_c^2(t) + \rho h v_1^2 \end{aligned} \tag{3.42}$$

Therefore: $\min \{J\}_{v_1}$ v with subject to

$$h > 0$$

and

$$\rho > 0$$

$$\Rightarrow v_1(t) = k_1(x_{cd} - x_c(t)) - k_2 \dot{x}_c(t) \tag{3.43}$$

where the predictive control law gains are as follow:

$$k_1 = \frac{2h}{h^3 + 4\rho}$$

and

$$k_2 = \frac{2h^2}{h^3 + 4\rho}$$

. The block diagram of the closed-loop system can be presented as shown in Fig. 3.6.

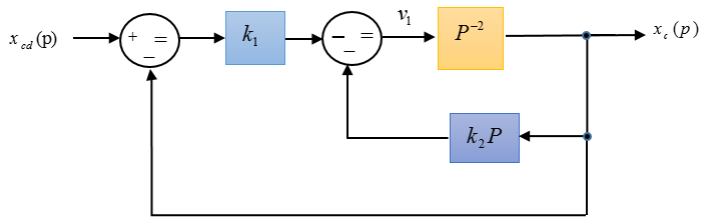


Figure 3.6: DDMR-2 Order Close Loop system

Regarding the response of the equivalent system, we want it to adopt similar behavior to the second-order system:

$$\frac{\omega_0^2}{p^2 + 2\zeta\omega_0 p + \omega_0^2}$$

where ζ and ω_0 are respectively the damping factor and the natural frequency.
The close loop transfer function of the system shown in Fig. 3.6 is given by:

$$\frac{x_c(p)}{x_{cd}(p)} = \frac{k_1}{p^2 + k_2 p + k_1} \tag{3.44}$$

we can verify that:

$$\begin{cases} 2\zeta w_0 = \frac{2h^2}{h^3+4\rho} = hk \\ \omega_0^2 = \frac{2h}{h^3+4\rho} \end{cases} \tag{3.45}$$

Therefore:

$$\begin{cases} h = \frac{2\zeta}{\omega_0} \\ \rho = \frac{\zeta}{\omega_0^3}\left(1 - 2\zeta^2\right) \end{cases} \tag{3.46}$$

The weight factor $\rho > 0$; therefore, from Eq. (3.46), as the damping factor ζ must be less than $\frac{1}{\sqrt{2}}$ to obtain good damping, we have to choose ζ as near as possible to $\frac{1}{\sqrt{2}}$; let us suppose $\zeta = \frac{0.999}{\sqrt{2}}$.

3.4.3 Graphic Base Tuning of MPC Gains for DDMR

We want to find h and ρ values in such a way that our system fulfils the following conditions:

$$\begin{cases} |\tau_i| \leq 150\,Nm\,, i = 1, 2 \\ T_{r\pm5\%} \leq 1\ s \end{cases} \tag{3.47}$$

with[91]:

$$T_{r\pm5\%} = \frac{-\ln(0.05\sqrt{1-\xi^2})}{\xi w_0} \tag{3.48}$$

Hence, we calculate and draw the maximum and minimum torque values (for each arm articulation) and the settling time values for several values of ω_0 (from 1 rd/s to 10 rd/s), with a Matlab script we could obtained the following graphs:

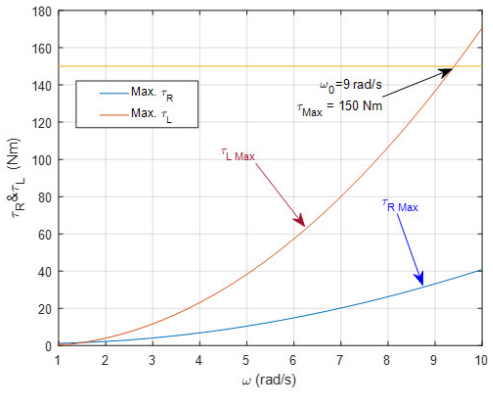


Figure 3.7: Variation of τ_{iMax}as function of ω

From Figure 3.7, to have the maximum wheel torque less than or equal to 150 Nm, the natural frequency ω_0 should be less than or equal to 9 rd/s. From Figure 3.8, to have the minimum wheel torque greater than or equal to 150 Nm, the natural frequency ω_0 should be less than or equal to 9 rd/s. In addition, from Figure 3.9, to have a settling time $T_{r\pm5\%}$ less than or equal to 1 s, the natural frequency should be greater than or equal to 4.8 rd/s.

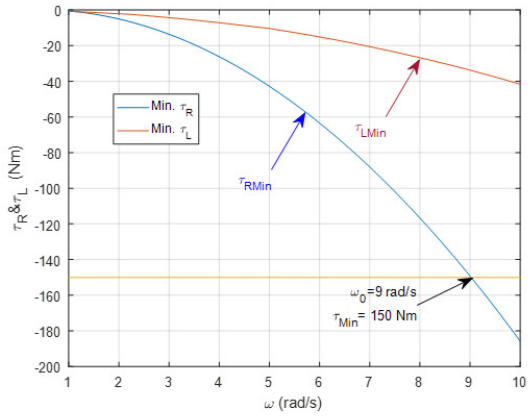


Figure 3.8: Variation of τ_{iMin}as function of ω

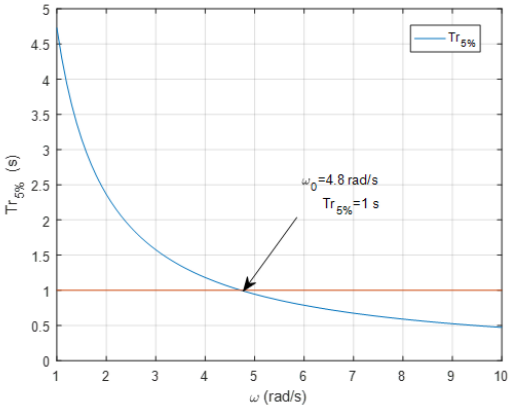


Figure 3.9: Variation of settling time as function of ω

Finally, the natural frequency ω_0 should verify the following inequality:

$$4.8\, rd/s \leq \omega_0 \leq 9\, rd/s \tag{3.49}$$

Let us take $\omega_0 = 9$ rd/s, which means that $h = 0.157$ s,$\rho = 1.9\ .10^{-6}$ and the predictor control law gains values are $k_1 = 72.26$ and $k_2 = 12$.

3.5 Simulations and Results

To show the suitability of our tinning parameters, k_1 and k_2, let us consider $m_c = 1$ (Kg) and $I_c = 1$ (Nm^2) the mass and the moment of inertia of the platform without the driving wheels. $m_w = 0.1$ (Kg) and $I_w = 0.1$ (Nm^2) : the mass and the moment of inertia of each driving wheel plus the rotor of its motor. $I_m = 0.1$ (Nm^2) :the moment of inertia of each wheel and the motor rotor about a wheel diameter. For trajectory generation, we show that the DDMR moves from an initial point $P_{init} = (1, 1)$ to a desired position $P_{dis} = (40, 60)$ during a time period of 60 s, following a short distance, which will be a straight line; note that we suppose that there is no obstacle between P_{init} and P_{dis}.

Fig.3.10,3.11,3.12,3.13,3.14 and 3.15 represent the trajectory of Xc ,Yc and (Xc,Yc) coordinates of the DDMR, from the initial position to the final one; we noticed fast asymp-

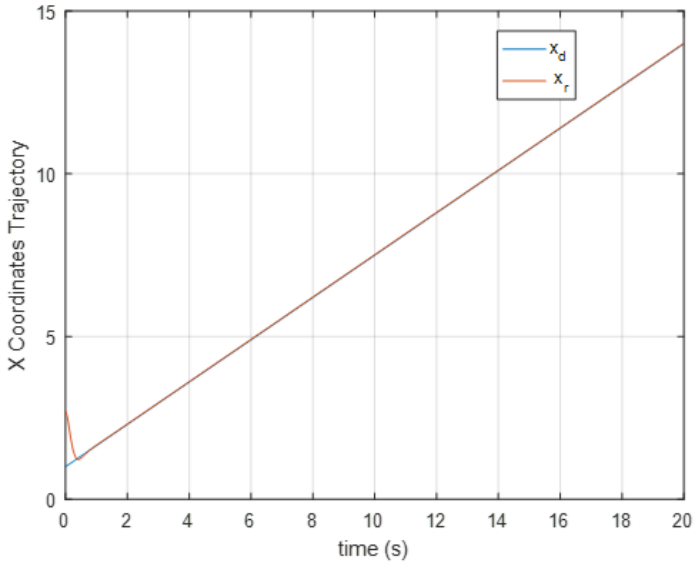


Figure 3.10: Desired and Real Xc with MPC

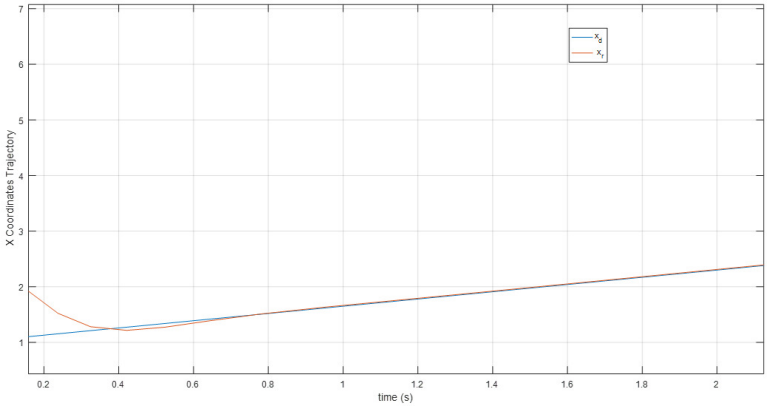


Figure 3.11: Desired and Real Xc with MPC (Zoom)

totic convergence of the two coordinates without oscillations comparing with Yoshio work and his team[1] illustrated in Fig. 3.16,they have used a proportional derivate controller ($K_p = 2$ and $K_d = 5$) for DDMR in order to follow a strait line. it is observed that the imposed settling time $T_{r\pm5\%}$ limit (less than 1s) is well respected. Fig. 3.17 shows using the proposed control approach the convergence of the trajectory tracking error towards a very small value. In Fig.3.43 the DDMR synthetic controls v_1, v_2 given by Eq.(3.43) are shown. As we can notice, the synthetic controls do converge as well to almost to zero when the succeed to follow the desired trajectory. Fig.3.19 shows that the DDMR torques obtained from the synthetic controls using Eq.(3.37), respect the imposed torque limitations . The Fig. 3.20, shows that the DDMR follows the desired trajectory rapidly without any overshoot or oscillations.

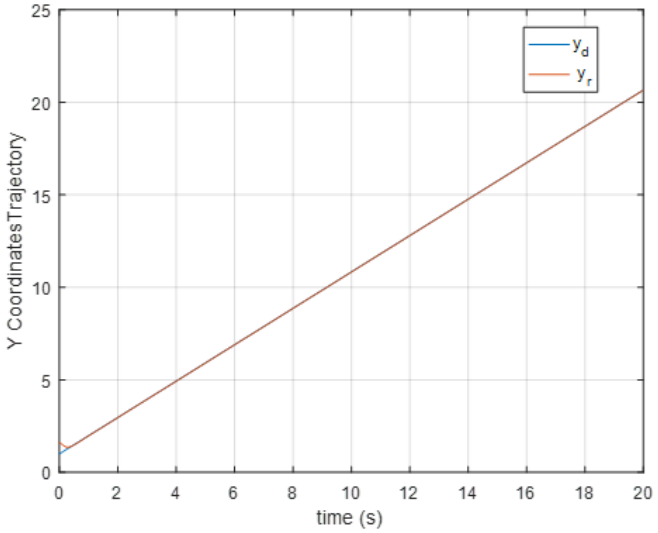


Figure 3.12: Desired and Real Yc with MPC

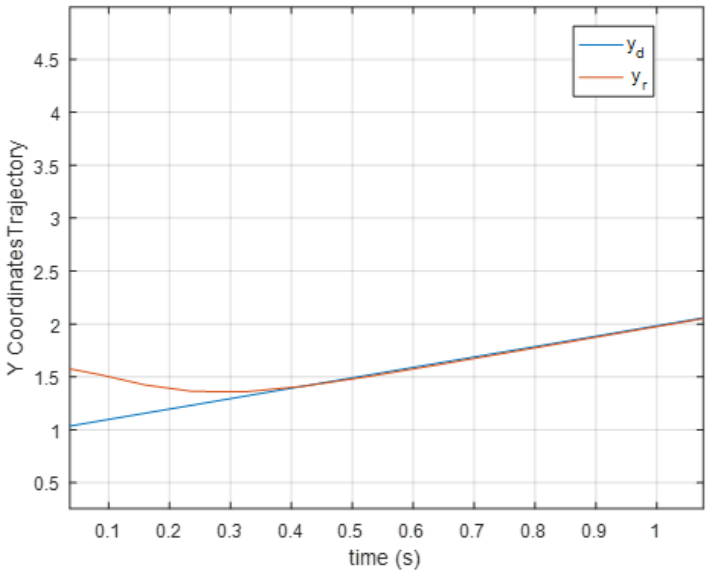


Figure 3.13: Desired and Real Yc with MPC (Zoom)

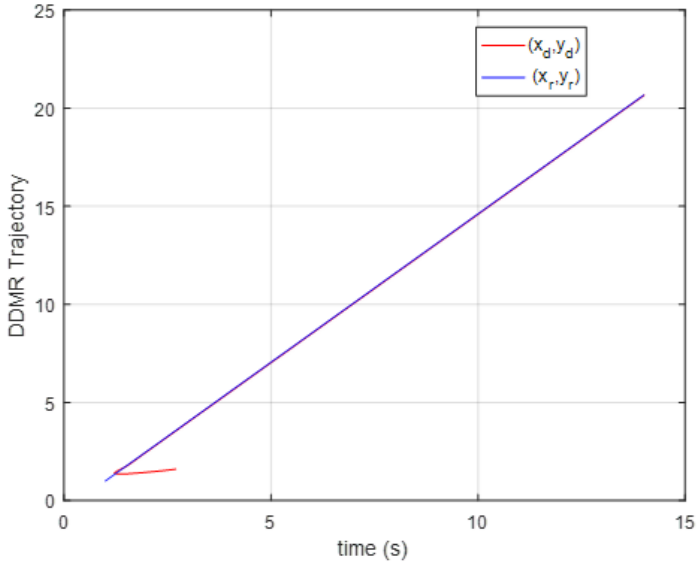


Figure 3.14: Desired and Real Trajectory with MPC

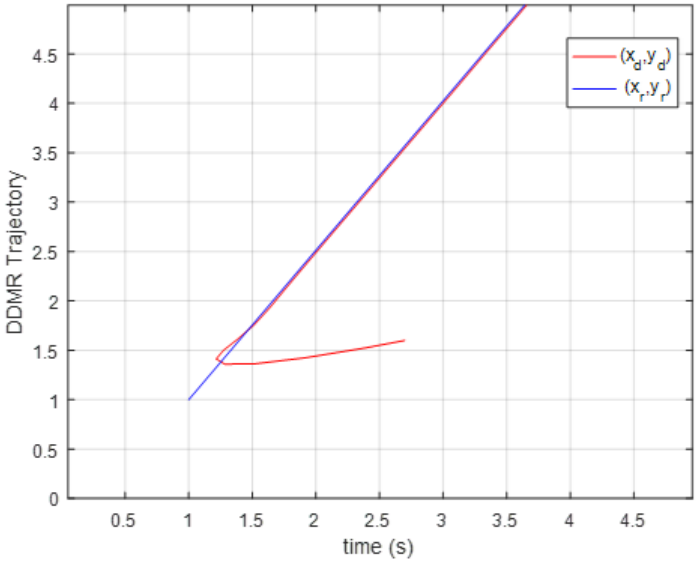


Figure 3.15: Desired and Real Trajectory with MPC (Zoom)

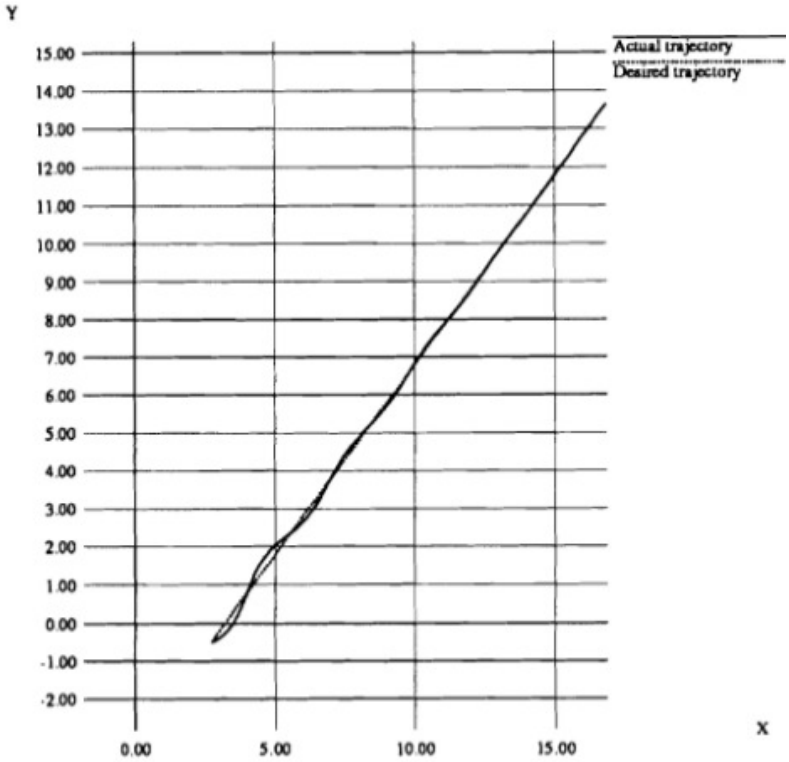


Figure 3.16: Desired and Real Trajectory PD[1]

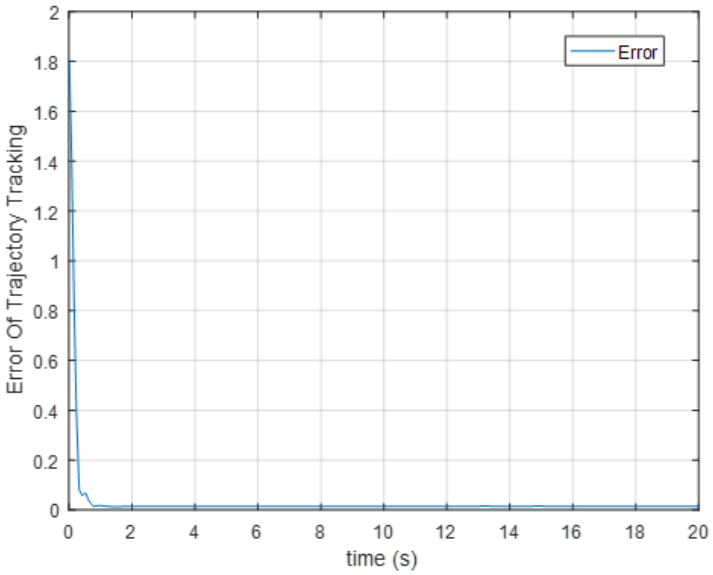


Figure 3.17: Trajectory tracking with MPC

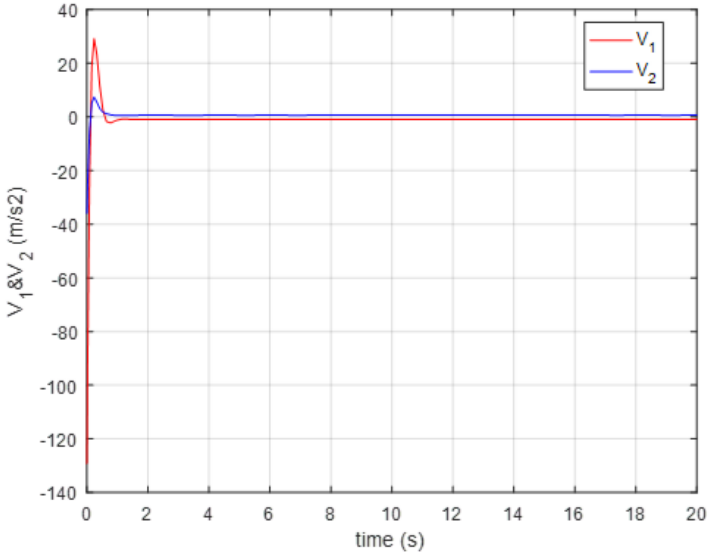


Figure 3.18: DDMR Synthetic control with MPC

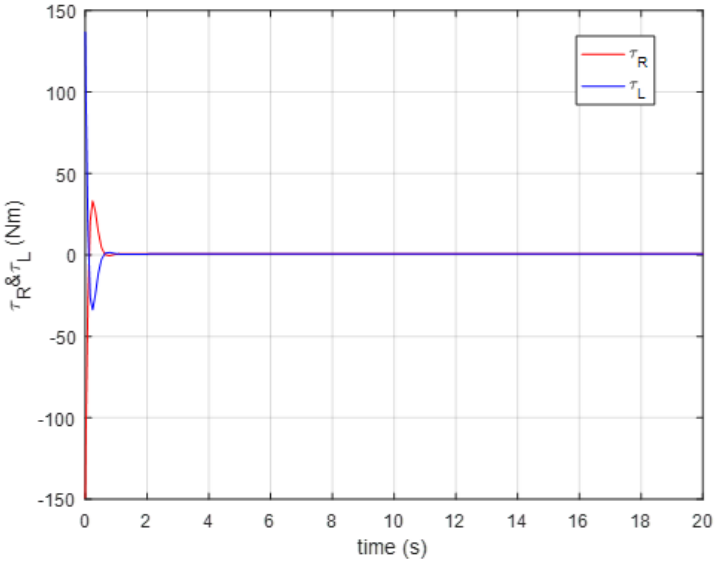


Figure 3.19: Wheels acting Torques with MPC

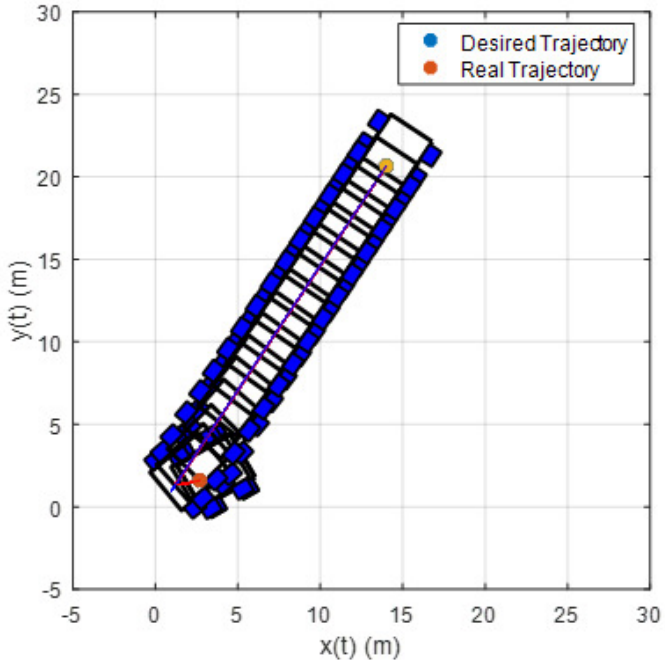


Figure 3.20: DDMR Trajectory simulation with MPC

3.6 Conclusion

In this chapter, we proposed a model predictive control for a DDMR. The control strategy started with an input–output linearization technique to linearize the nonlinear dynamic model of the DDMR, then, based on the obtained linear model, a predictive control law was developed. The prediction horizon time h and the weight factor ρ were tuned based on torques and settling time graphics analysis; the simulation results showed that we could find a compromise between the settling time and the energy required for the mobile robot to follow the desired trajectory. In addition, the proposed approach of control produces a better system performance than the PD control technique proposed by Yoshio and Xiaoping[1]. The research carried out in this chapter has been published in an international journal[48] .

Chapter 4

PSO based tuning of Predictive Controller for 6 DOF Mobile Manipulator Robot

4.1 Introduction

In the present section we have used constrained particle swarm optimization (COPSO) technic to tune the model predictive control (MPC) approach developed in our previous work[45] and[48] for controlling a six degrees of freedom (DOF) mobile manipulator robot and meet the required performances. After getting linearized dynamic models for the mobile manipulator robot by using a feedback linearization control, MPC control is developed with minimizing a quadratic criterion which is a function of the predicted error and its derivate. Next, the parameters (h: horizon time of prediction, ρ: weight factor) of the MPC control are computed by identification between the gains found by the COPSO algorithm and those of the MPC. In order to show the efficiency of the proposed tuning method, some simulation results are given.

4.2 6 DOF MMR Geometric Description

The manipulator robot shown in the Fig. 4.1 has six degrees of freedom, where $\theta_{i=1,2,3,4,5,6}$ are the joints angles, while $L_{j=1,2,3}$ and $M_{j=1,2,3}$ are respectively the length and the mass of the first (j=1), the second (j=2) link and the third link (j=3) of the robot. $I_{j=4,5,6}$ are respectively the wrist inertia moments, the variable g denote the gravity force.

4.3 6 DOF Manipulator Arm Models

4.3.1 Geometric Model

Based on the Fig. 4.1, we can calculate the coordinates of the end effector manipulator robot with respect to the frame (o, X, Y, Z) as follow:

$$\begin{cases} x_2 = (l_2c_2 + l_3c_{23})c_1 \\ y_2 = (l_2c_2 + l_3c_{23})s_1 \\ z_2 = l_1 + l_2s_2 + l_3s_{23} \end{cases} \tag{4.1}$$

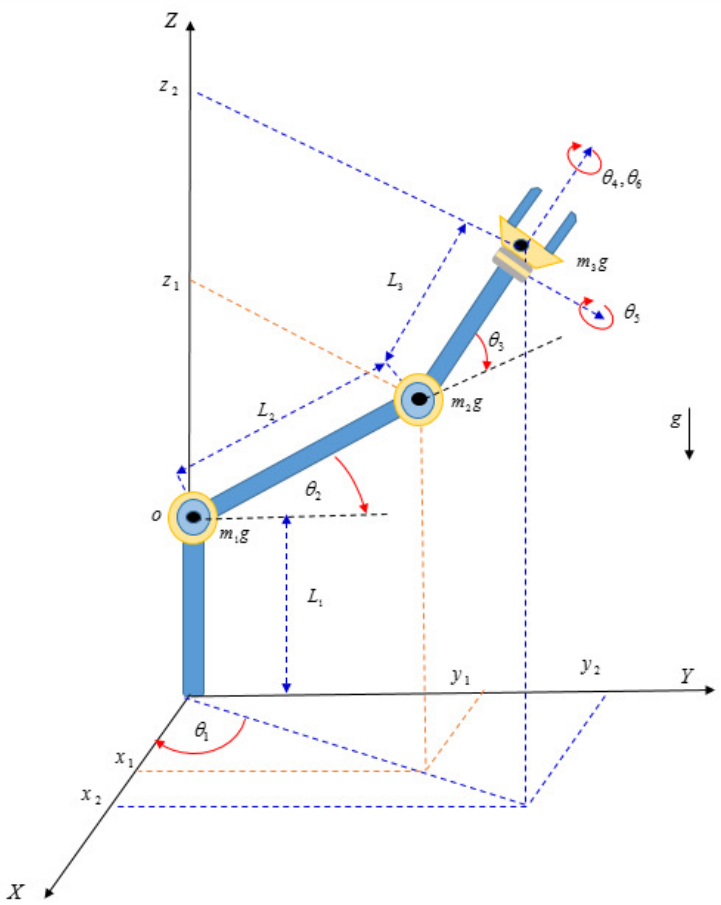


Figure 4.1: 6 DOF Manipulator Arm

The equation (4.1) presents the geometric model which gives us the robot end effector coordinates based on the its articulations values during the its motion.

4.3.2 Kinematic Model

Calculating the derivative of the Forward geometric model (FGM) Eq. (4.1) with respect to time, the kinematic model of this robot is given by the following formula:

$$\begin{cases} \dot{x}_2 = [-l_2 s_2 w_2 - l_3 s_{23}(w_2 + w_3)]c_1 - (l_2 c_2 + l_3 c_{23}) s_1 w_1 \\ \dot{y}_2 = [-l_2 s_2 w_2 - l_3 s_{23}(w_2 + w_3)]s_1 + (l_2 c_2 + l_3 c_{23}) c_1 w_1 \\ \dot{z}_2 = l_2 c_2 w_2 + l_3 (w_2 + w_3) c_{23} \end{cases} \tag{4.2}$$

The Eq. (4.2) presents the kinematic model which gives us end robot effector linear velocities based on its articulations velocities values during its motion.

4.3.3 Dynamic Model

Using the equations (4.1) and(4.2), the total kinetic energy is given by the following formula:

$$\begin{array}{l} K_E = \frac{1}{2}m_2 l_2^2\omega_2^2 + \frac{1}{2}m_3 l_3^2\omega_1^2 c_{23}^2 + \frac{1}{2}m_3 l_2^2\omega_1^2 c_2^2 + m_3 l_2\omega_1^2 c_2 l_3 c_{23} \\ +m_3 l_2 s_2\omega_2^2 l_3 s_{23} + \frac{1}{2}m_2 l_2^2\omega_1^2 c_2^2 + m_3 l_2 s_2\omega_2\omega_1 l_3 s_{23}\omega_3 + \frac{1}{2}m_3 l_2^2\omega_2^2 \\ +\frac{1}{2}m_3 l_3^2\omega_3^2 + \frac{1}{2}m_3 l_3^2\omega_2^2 + m_3 l_3^2\omega_2\omega_3 + m_3 l_2 c_2\omega_2 l_3 c_{23}\omega_3 \\ +\frac{1}{2}I_5\omega_5^2 + \frac{1}{2}I_6\omega_6^2 + \frac{1}{2}m_1 l_1^2\omega_1^2 + m_3 l_2 c_2\omega_2^2 l_3 c_{23} + \frac{1}{2}I_4\omega_4^2 \end{array} \tag{4.3}$$

And the potential energy:

$$P_E = m_1 g l_1 + m_2 g(l_1 + l_2 s_2) + m_3 g(l_1 + l_2 s_2 + l_3 s_{23}) \tag{4.4}$$

Using Euler-Lagrange Formalism:

$$L = K_E - P_E$$

The motion equation of the manipulator robot is the solution of the following equations:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i},\ i = 1, 2, 3, 4, 5, 6 \tag{4.5}$$

Where L and τ_i are respectively the Lagrangian and the torque vector.
Developing the equation (4.5), the dynamic model of a robotic arm with Six degrees of freedom (DOF) is given by the following formula:

$$\begin{cases} \tau_i = M(\theta_i)\ddot{\theta} + C(\theta_i, \dot{\theta}_i) + G(\theta_i), i = 1, 2, 3, 4, 5, 6 \\ y = [\theta_1 \ \ \theta_2 \ \ \theta_3 \ \ \theta_4 \ \ \theta_5 \ \ \theta_6]^T \end{cases} \tag{4.6}$$

where: $y = [\theta_1 \ \ \theta_2 \ \ \theta_3 \ \ \theta_4 \ \ \theta_5 \ \ \theta_6]^T$: Articulation Variables Vector and also is the output vector;
$\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 \end{bmatrix}^{\mathrm{T}}$: is torque vector (control input);

$M(q_i) = \begin{vmatrix} d_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{22} & d_{23} & 0 & 0 & 0 \\ 0 & d_{32} & d_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_6 \end{vmatrix}$: Inertia moment matrix;

$C(q_i, \dot{q}_i) = \left| \begin{array}{cccccc} c_1 & c_2 & c_3 & 0 & 0 & 0 \end{array} \right|^T$: Centripetal and coriolis vector;

$G(q_i) = \left| \begin{array}{cccccc} 0 & g_2 & g_3 & 0 & 0 & 0 \end{array} \right|$:Gravitational vector;

note that:

$$\begin{cases} d_{11} = 2m_3 l_2 l_3 c_2 c_{23} + m_2 l_2^2 c_2^2 + m_3 l_2^2 c_2^2 + m_3 l_3^2 c_{23}^2 + m_1 l_1^2 \\ d_{22} = 2m_3 l_2 l_3 c_2 c_{23} + m_3 l_3^2 + m_3 l_2^2 + m_2 l_2^2 + 2m_3 l_2 l_3 s_2 s_{23} \\ d_{23} = m_3 l_2 l_3 s_2 s_{23} + m_3 l_2 l_3 c_2 c_{23} + m_3 l_3^2 \\ d_{32} = m_3 l_2 l_3 s_2 s_{23} + m_3 l_2 l_3 c_2 c_{23} + m_3 l_3^2 \\ d_{33} = m_3 {l_3}^2 \end{cases}$$

$$\begin{cases} h_1 = -2\omega_1[m_3 l_2 l_3 \omega_2 s_2 c_{23} + m_3 l_2 l_3 \omega_2 c_2 s_{23} + m_3 l_2 l_3 \omega_3 c_2 s_{23} \\ +m_2 l_2^2 c_2 s_2 \omega_2 + m_3 l_2^2 c_2 s_2 \omega_2 + m_3 l_3^2 c_{23} s_{23} \omega_2 + m_3 l_3^2 c_{23} s_{23} \omega_3] \\ h_2 = 2m_3 l_2 l_3 \omega_2 \omega_3 s_2 c_{23} - m_3 l_2 l_3 \omega_2^2 s_2 c_{23} + m_3 l_2 l_3 \omega_1^2 c_2 s_{23} \\ +m_3 l_2 l_3 \omega_1^2 s_2 c_{23} - m_3 l_2 l_3 \omega_3^2 c_2 s_{23} + m_3 l_2 l_3 \omega_2^2 c_2 s_{23} - 2m_3 l_2 l_3 \omega_2 \omega_3 c_2 s_{23} \\ +m_3 l_3^2 \omega_1^2 c_{23} s_{23} + m_2 l_2^2 \omega_1^2 c_2 s_2 + m_3 l_2^2 \omega_1^2 c_2 s_2 - m_3 l_2 l_3 \omega_2^2 c_2 s_{23} + m_3 l_2 l_3 \omega_3^2 s_2 c_{23} \\ h_3 = -m_3 l_3[-l_2 s_2 c_{23} \omega_2 \omega_3 + l_2 c_2 s_{23} \omega_2 \omega_3 - l_2 c_2 s_{23} \omega_1^2 - l_2 c_2 s_{23} \omega_2^2 \\ +l_2 s_2 c_{23} \omega_2 \omega_3 - \omega_1^2 l_3 c_{23} s_{23} + l_2 s_2 c_{23} \omega_2^2 - l_2 c_2 s_{23} \omega_2 \omega_3] \end{cases}$$

$$\begin{cases} g_2 = m_2 g l_2 c_2 + m_3 g l_2 c_2 + m_3 g l_3 c_{23} \\ g_3 = m_3 g l_3 c_{23} \end{cases}$$

4.4 Control Algorithm

In this section we have used the linearized dynamic models developed so far in the present work (6 DOF manipulator arm and differential-Drive mobile Robot) as well as the obtained MPC gains which can be summarized as follow:

- Manipulator Arm:

$\bullet DynamicModel : \begin{cases} \tau_i = M(\theta_i)\ddot{\theta} + C(\theta_i, \dot{\theta}_i) + G(\theta_i), i = 1,2,3,4,5,6 \\ y = [\theta_1 \ \ \theta_2 \ \ \theta_3 \ \ \theta_4 \ \ \theta_5 \ \ \theta_6]^T \end{cases}$

$\bullet CostFunction : J = e_1^2(t+h) + \rho \dot{e}_1^2(t+h)$

$-MPCgains : \begin{cases} k_{pa} = \frac{2}{h^2+4\rho} \\ k_{da} = \frac{2h^2+4\rho}{h^3+4\rho h} \end{cases}$

- Mobile Base (DDMR):

$\bullet DynamicModel : \tau = \bar{M}u + \bar{V}$

$\bullet CostFunction : J = e_1^2(t+h) + \rho \int_t^{t+h} v^2 dt$

$\bullet MPCgains : \begin{cases} k_{pb} = \frac{2h}{h^3+4\rho} \\ k_{db} = \frac{2h^2}{h^3+4\rho} \end{cases}$

To find the best values of the MPC gains, as we have explained in the second and third chapters,based on the imposed constraints conditions, we have used a Matlab script to generate the maximum,minimum and the settling time variation curves as a function of ω and the results are as follow:

- Manipulator Arm: $\begin{cases} |\tau_i| \leq 100Nm \\ T_{R\pm5\%} \leq 1.7S \\ i = 1,2,3 \end{cases} \Rightarrow \begin{cases} k_{pa} = 7.78 \\ k_{da} = 3.94 \end{cases}$

- Mobile Base (DDMR): $\begin{cases} |\tau_i| \leq 100Nm \\ T_{R\pm5\%} \leq 1.7S \\ i = 1,2,3 \end{cases} \Rightarrow \begin{cases} k_{pb} = 72.26 \\ k_{db} = 12 \end{cases}$

In this section we will be using a different way to tune the MPC gains which is a

constrained particle swarm optimization method, and to the suitability of the proposed method, a simulation and results analysis are provided in the following section.

4.5 MPC parameters tuning with a COPSO

4.5.1 PSO Description

The particle swarm optimization idea was proposed for the first time in 1995 by Kennedy and Eberhart[106], In PSO algorithm, we have a swarm of N particles fly through the search space, each particle represents a candidate solution to the optimization problem. Each particle flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also the flying experience of the other particles[107],[108],[109],[110],[111],[112],[113],[114],[115],[116]
The mathematical description of the basic particle swarm optimization is as follows:
Supposing the scale of swam is N, the position and the velocity of particle i, can be expressed as follow:

$$\begin{aligned} x_i &= (x_{i1}, x_{i2}, ..., x_{id}) \quad i = 1, 2, ..., N \\ v_i &= (v_{i1}, v_{i2}, ..., v_{id}) \end{aligned} \tag{4.7}$$

The position and the velocity of each particle I at k iteration is updated via the following formula:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 \left(P_d(k) - x_i(k)\right) + c_2 r_2 \left(P_g(k) - x_i(k)\right) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \tag{4.8}$$

Note that: - k: iteration number :1,..,Maxit
- i: particle number: 1,. . . ,N
- $P_d(k)$: Personal best position of i particle among the k personal positions.
- $x_i(k)$: Current position of i particle at k iteration.
- $P_g(k)$: Global best position of the whole swarm among the k global positions.
- c_1: The acceleration factor related to $P_d(k)$.
- c_2: The acceleration factor related to $P_g(k)$.
- r_1, r_2: Two random number between 0 and 1.
- ω :Inertia factor is brought into the equation to balance between the global search and local search capability[107]. It can be a positive constant or even a positive linear or non-linear function of time.

4.5.2 PSO based Model Predictive controller

In this work, an optimization algorithm based on a constrained particles swarm was developed; in order to find the optimal values of the parameters for each articulation of the manipulator arm, and then by identification between these optimal parameters and those of the MPC(15), we can calculate the parameters h and ρ of the MPC controller. The structure diagram of COPSO based MPC is shown in Fig.4.2
The performance criterion in MPC is as follow:
- For the manipulator Arm:

$$\delta_{ma} = \sum e^2(t) + \sum \dot{e}^2(t) \tag{4.9}$$

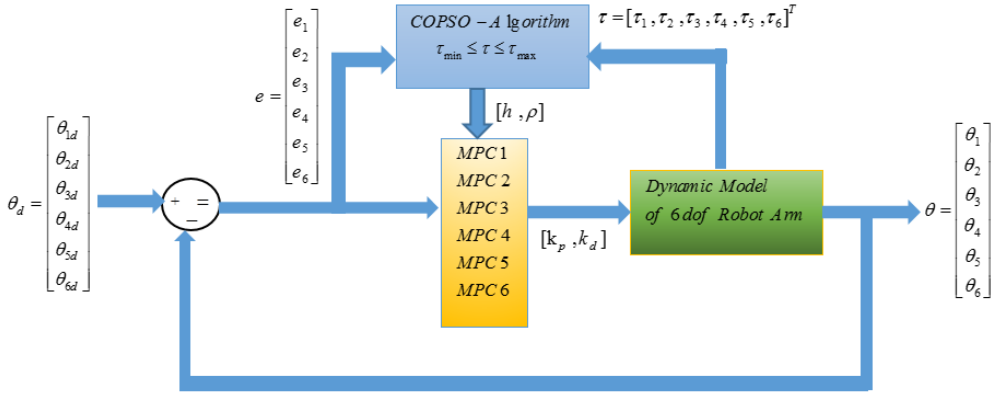


Figure 4.2: COPSO based MPC closed loop structure's for 6 dof Manipulator Arm

- For the Mobile Base:

$$\delta_m b = \sum e^2 (t+h) + \sum v^2 (t+h) \tag{4.10}$$

i	1	2	3	4	5	6
k_{pma}	21	21	25	17	25	25
k_{dma}	9	9	10	7	10	10

Table 4.1: k_{pma} and k_{dma} values found by COPSO Algorithm

i	1	2	3	4	5	6
h	0.1311	0.1311	0.1171	0.1839	0.1171	0.1171
ρ	0.0195	0.0195	0.0165	0.0209	0.0165	0.0165

Table 4.2: h and ρ values computed based on k_{pma} and k_{dma}

The above values of h and ρ are the solutions of the following equations systems, and their numerical values are computed by using Maple software:

$$\begin{cases} k_1(h^2 + 4\rho) - 2 = 0 \\ k_2(h^3 + 4\rho h) - 2h^2 + 4\rho = 0 \end{cases} \tag{4.11}$$

The Fig. 4.3 shows the latest fitness convergence curve of the COPSO algorithm for the manipulator arm, note that we have executed the COPSO several time, for each run we have resized the search interval in order to get a better (minimum) value of the cost function, which explains the the convergence of the cost function after a 10 iterations only. The Fig. 4.5 shows as well the latest fitness convergence curve of the COPSO algorithm for the DDMR, note that we have executed the COPSO several time, for each run we have resized the search interval in order to get a better (minimum) value of the cost function, which explains the the convergence of the cost function after a 15 iterations only.

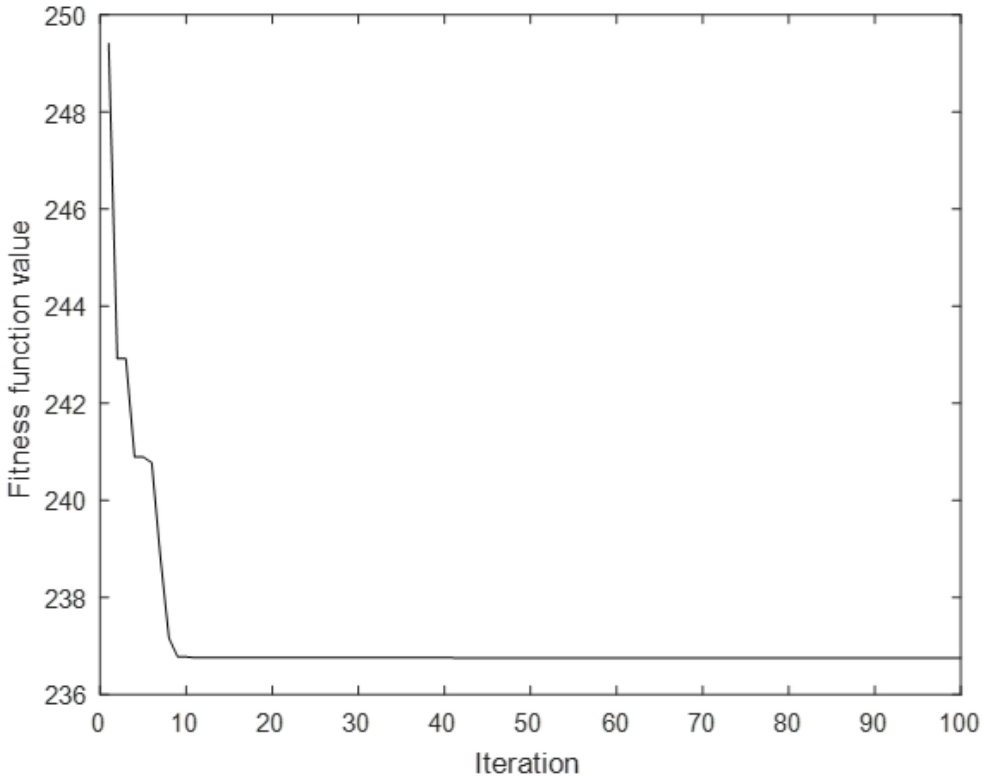


Figure 4.3: Manipulator arm fitness convergence

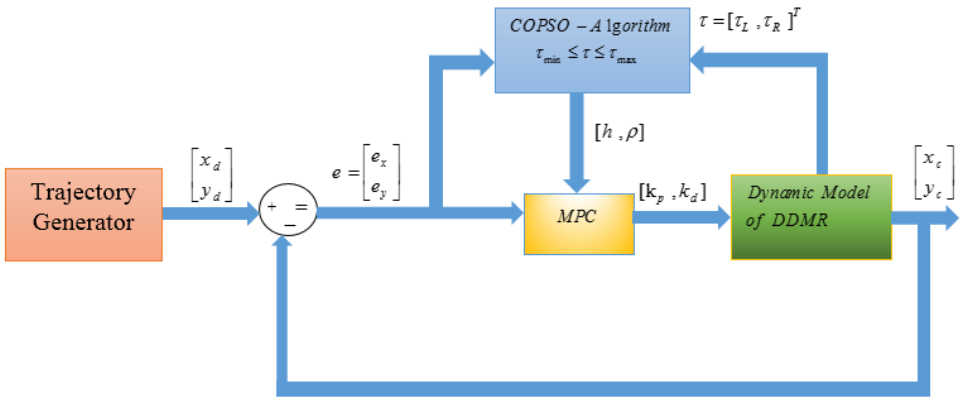


Figure 4.4: COPSO based MPC closed loop structure's for DDMR

The trajectory generator script can only generate one trajectory for each value of prediction time h, which means the both left and right wheels must have the same k_p and k_d, which is not the case of the manipulator arm we could have a different values for each link. therefore:

$$\begin{cases} h = 0.1483 \\ \rho = 10^{-}4 \end{cases} \Rightarrow \begin{cases} k_{p-mb} = 89.31 \\ k_{d\_mb} = 13.36 \end{cases}$$

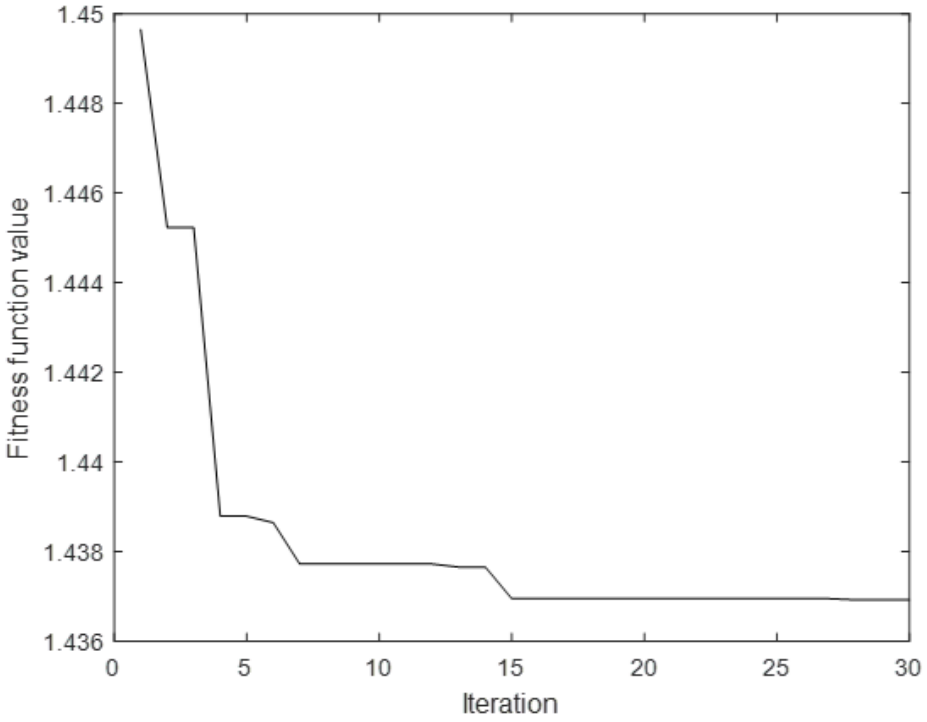


Figure 4.5: DDMR Fitness Convergence

4.6 Trajectory Planning

We want the mobile manipulator robot to move up to well-determined object from its initial location to another desired one by choosing the shortest possible path, it is assumed that there are no obstacles along the path whatsoever for the mobile base as well as the manipulator arm. The mobile manipulator arm during its movement will pass through

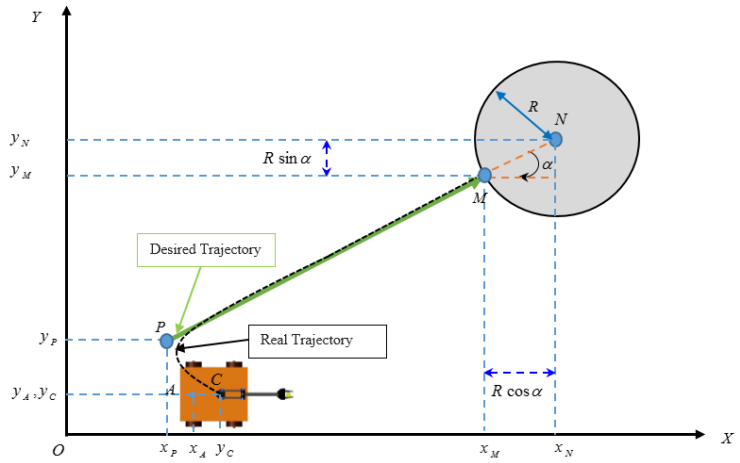


Figure 4.6: Trajectory Planning

two phases:
- First phase: the mobile base must starts moving first from the initial position the point A until the point M which is the intersection of the line (PN) and the circle centered in the

point N and has radius R $=\frac{(L_2+L_3)}{2}$, Fig. 4.6.

The calculation of the pint M coordinates is as follow:

Based on the Fig. 4.6, we can write:

$$\tan\alpha = \frac{y_N - y_M}{x_N - x_M} \tag{4.12}$$

Therefore:

$$\begin{cases} x_M = x_N - R\cos\alpha \\ y_M = y_N - R\sin\alpha \end{cases} \tag{4.13}$$

and the α angle it can be calculated as well as below:

$$\alpha = a\tan 2(\mathrm{x}_N - x_A, \mathrm{y}_N - y_A) \tag{4.14}$$

Therefore the M point is the mobile base desired position, where it must stop at it, and it is as well a permissive condition for the manipulator arm prior to start moving to grab the target object.
Now, let define the desired trajectory which the mobile base has to follow, suppose the starting of the mobile base (point A) and final position of the end effector (Point N) are as below:

$$\begin{cases} A = (1, 1) \\ M = (30, 50, 1.75) \end{cases} \tag{4.15}$$

as per the equations (4.13), (4.14), the M point coordinates and α values are as below:

$$\begin{cases} \mathrm{M} = (29.1349, 49.4907) \\ \alpha \approx 60^\circ \end{cases} \tag{4.16}$$

Since the general form of a straight line is as follow:

$$\begin{cases} x = a_x t + b_x \\ y = a_y t + b_y \\ t(s) \in \left| \begin{array}{cc} 0 & 60 \end{array} \right| \end{cases} \tag{4.17}$$

Therefore:

$$\begin{cases} x = 0.4833t + 1 \\ y = 0.8167t + 1 \\ t(s) \in \left| \begin{array}{cc} 0 & 60 \end{array} \right| \end{cases} \Rightarrow \begin{cases} y = 1.69x - 0.69 \\ x \in \left| \begin{array}{cc} 1 & 30 \end{array} \right| \\ y \in \left| \begin{array}{cc} 1 & 50 \end{array} \right| \end{cases} \tag{4.18}$$

- Second phase: once the mobile base reaches its desired position (M point) it has to stop, at this time the manipulator arm has the green light to start moving to grab the object target.

In order that the FGM and the IGM of the manipulator arm function properly, we have to update its initial position by the latest coordinate of the mobile base centre which is its desired point ,therefore the updated FGM and IGM are as follow:

- Forward Geometric Model of the first and the second links:

$$\begin{cases} x_2 = (l_2c_2 + l_3c_{23})c_1 + x_M \\ y_2 = (l_2c_2 + l_3c_{23})s_1 + y_M \\ z_2 = l_1 + l_2s_2 + l_3s_{23} + z_M \end{cases} \tag{4.19}$$

$$\begin{cases} x_1 = l_2c_2c_1 + x_M \\ y_1 = l_2c_2s_1 + y_M \\ z_1 = l_1 + l_2s_2 + z_M \end{cases} \tag{4.20}$$

- Inverse Geometric Model

$$\theta_1 = a\tan 2(\mathrm{x}_2 - x_M, \mathrm{y}_2 - y_M) \tag{4.21}$$

and:

$$\theta_3 = ATAN2(\pm\sqrt{(1-\tau^2)}, \tau) \tag{4.22}$$

where:

$$\tau = \frac{\left(\left(\frac{y_2 - y_M}{s_1}\right)^2 + (z_2 - L_1 - z_M)^2 - (L^2{}_2 + L^2{}_3)\right)}{2L_2L_3}$$

and:

$$\begin{cases} cos(\theta_2) & = \frac{(z_2 - L_1 - z_M)[L_2 + L_3cos(\theta_3)] + (y_2 - y_M)[L_2sin(\theta_3)}{(L_2 + L_3c_3)^2 + (L_3s_3)^2} \\ sin(\theta_2) & = \frac{(y_2 - y_M)[L_2 + L_3cos(\theta_3)] - (z_2 - L_1 - z_M)[L_2sin(\theta_3)}{(L_2 + L_3c_3)^2 + (L_3s_3)^2} \end{cases} \tag{4.23}$$

therefor:

$$\theta_2 = ATAN2(sin\theta_2, cos\theta_2) \tag{4.24}$$

The motion flow chart Fig. 4.7 can be summarize as follow:first of all, the robot controller should read the mobile base initial position, the trajectory starting point (P) and the end effector desired position (N) as well as the manipulator arm initialθ_0 articulations values:

$$\begin{cases} A = (x_A, y_A) \\ P = (x_P, y_P) \\ N = (x_N, y_N, z_N) \\ \theta = \begin{bmatrix} \theta_{10} & \theta_{20} & \theta_{30} & \theta_{40} & \theta_{50} & \theta_{60} \end{bmatrix} \end{cases}$$

then, based on the above data, the controller compute the mobile base desired position and calculate the initial posture of the manipulator arm based on the last position of the mobile base, which should be its desired position (M). then compute the the desired posture θ_d through the inverse geometric model in order to be as set point vector for MPC controller.
Let suppose that:
$\theta_{init} = [\theta_{10}, \theta_{20}, \theta_{30}, \theta_{40}, \theta_{50}, \theta_{60}] = [0, -\pi/3, \pi/7, 3\pi/4, \pi/8, \pi/5]$
By using the manipulator arm Inverse Geometric model, the desired posture will be:
$\theta_{desir} = [\theta_{1d}, \theta_{2d}, \theta_{3d}, \theta_{4d}, \theta_{5d}, \theta_{6d}] = [0.5321, 0, 1.5669, 3\pi/3, \pi/4, \pi/6]$
note that, the θ_4,θ_5 and θ_6 we just supposed to have the above values.

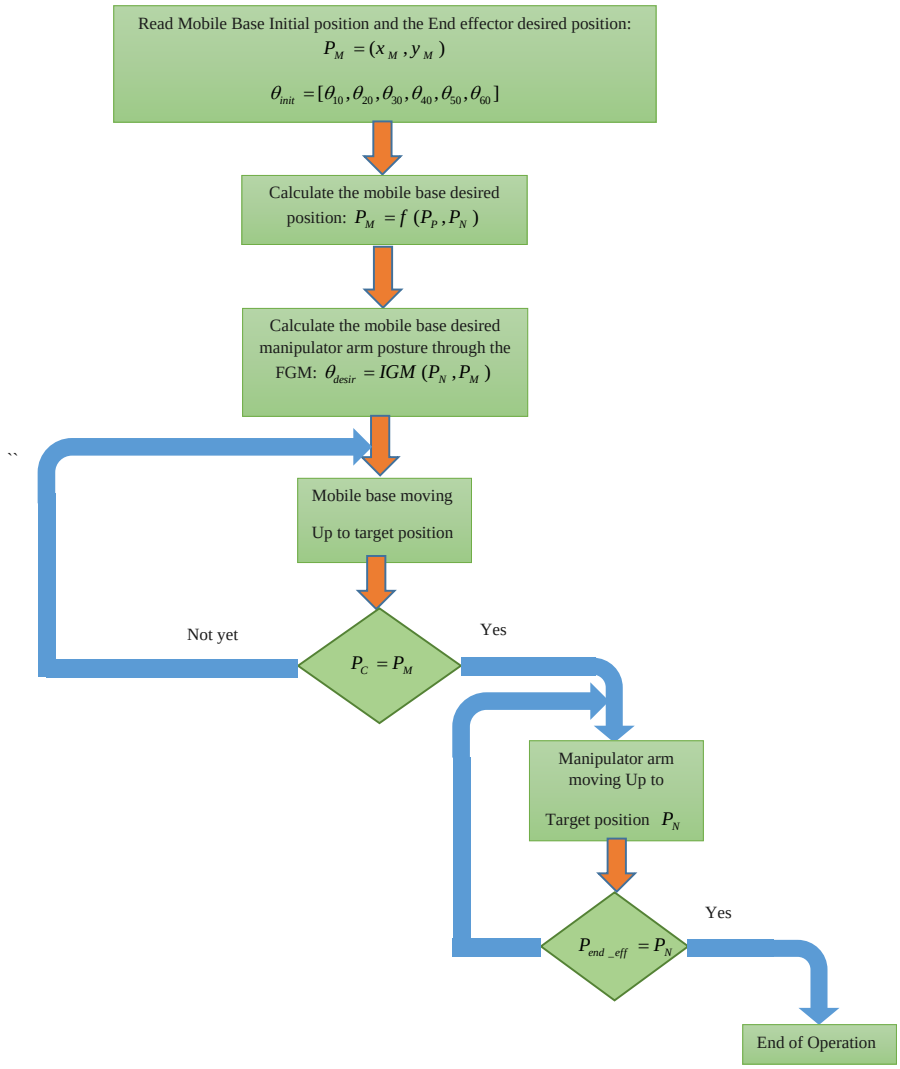


Figure 4.7: MMR Flow Chart Motion

4.7 Simulation and Results

In order to show the efficiency of our approach, let consider the following:
The Manipulator Arm: the mass, moments of inertia and the length of the Manipulator Arms are $\begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} = \begin{bmatrix} 2 & 0.5 & 0.3 \end{bmatrix} Kg$, $\begin{bmatrix} I_4 & I_5 & I_6 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix} N.Kg$ and $\begin{bmatrix} L_1 & L_2 & L_3 \end{bmatrix} = \begin{bmatrix} 0.5 & 1 & 1 \end{bmatrix} m$ respectively.
The initial and desired Articulations configurations are as follow:

$\theta_{init} = [\theta_{10}, \theta_{20}, \theta_{30}, \theta_{40}, \theta_{50}, \theta_{60}] = [0, -\pi/3, \pi/7, 3\pi/4, \pi/8, \pi/5]$
$\theta_{desir} = [\theta_{1d}, \theta_{2d}, \theta_{3d}, \theta_{4d}, \theta_{5d}, \theta_{6d}] = [0.5321, 0, 1.5669, 3\pi/3, \pi/4, \pi/6]$ The PSO tuning parameters are:
n=100: Population size
$\omega_{\min} = 0.4$:Minimum inertia weight
$\omega_{\min} = 0.9$:Maximum inertia weight
$c_1 = 2$: Acceleration factor related to the personnel best position
$c_2 = 2$: Acceleration factor related to the global best position.
$Maxit = 50$: Maximum iteration.

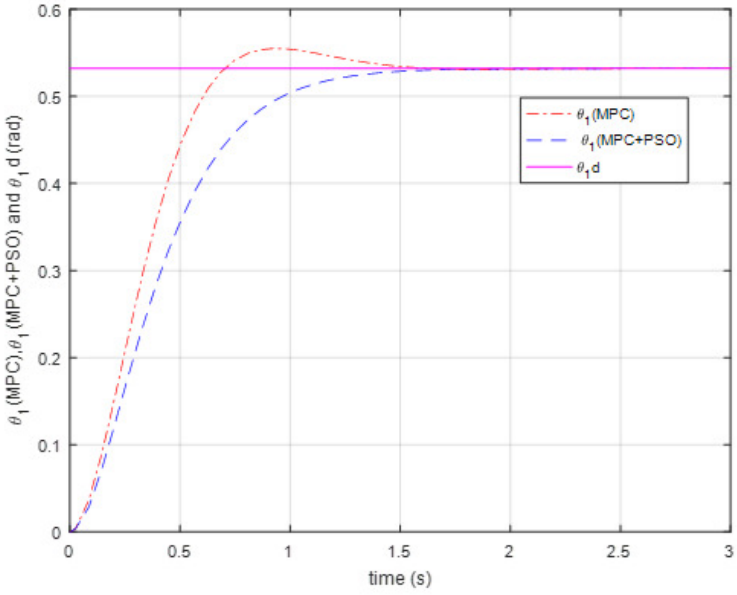


Figure 4.8: Desired and Real Articulation θ_1

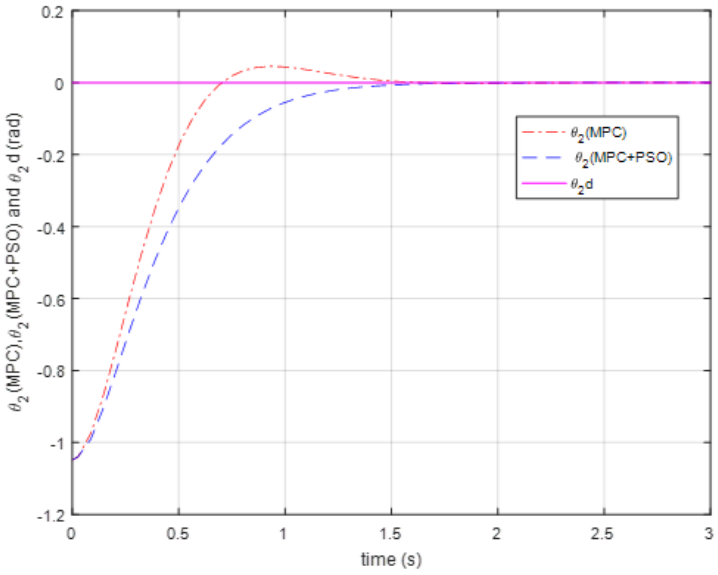


Figure 4.9: Desired and Real Articulation θ_2

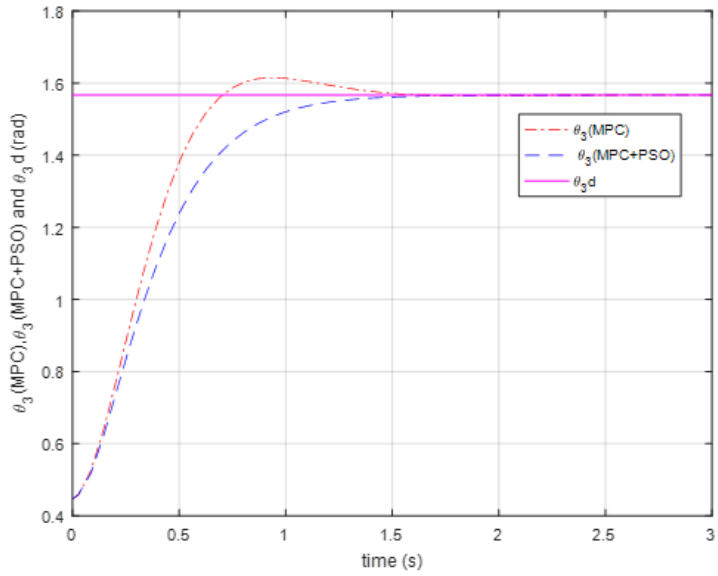


Figure 4.10: Desired and Real Articulation θ_3

Fig.4.8 up to 4.13 shows that MPC tuned with PSO provides better settling time $T_{r\pm5\%}$ than the MPC tuned with a graphics method and with out any oscillations or overshoot for all the outputs θ_i. Fig.14.14 shows, that the MPC tuned with PSO provides better dynamic convergence to zero of the angular errors compared with MPC tuned with a graphics method and all of them towards to zero once the end effector reaches the desired configuration. In Fig.4.15 and Fig. 4.16 we could see that, the angular velocities w_1 up to w_6 and the synthetic controls v_1 up to v_6 converge to zero once the end effector reaches the desired configuration. The different torques shown in Fig.4.17 and 4.18, proves that by using the tuned MPC with COPSO requires less energy compare it with MPC tuned with a graphic method. And the maximum torques is well respected by both methods The Fig.

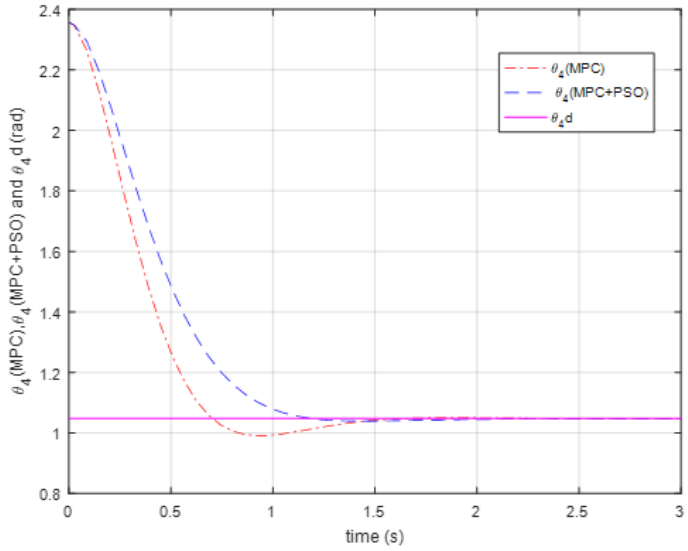


Figure 4.11: Desired and Real Articulation θ_4

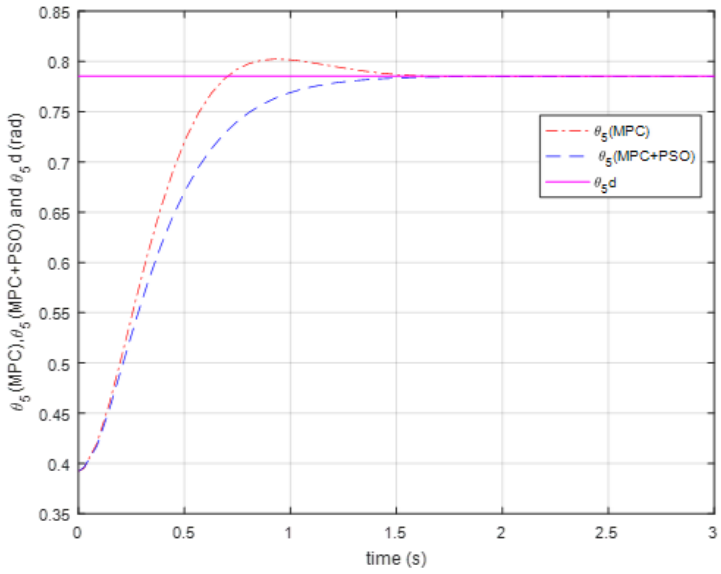


Figure 4.12: Desired and Real Articulation θ_5

4.19, shows the manipulator arm end effector trajectory from the initial conditions to the desired configurations.

The DDMR figures : Fig.4.20, Fig. 4.21, Fig.4.22, Fig.4.23, Fig. 4.24, Fig.4.25 and Fig.4.26 shows that the tuned MPC with COPSO did not add a significant contribution compared with MPC tuned with the graphic method.

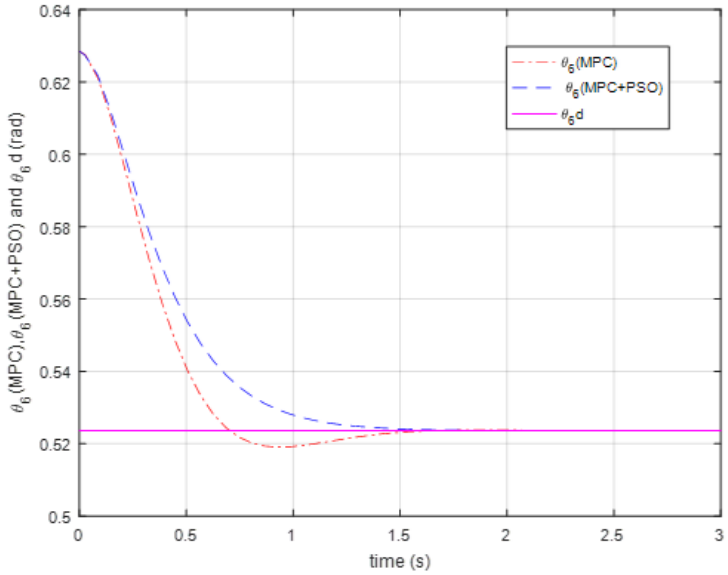


Figure 4.13: Desired and Real Articulation θ_6

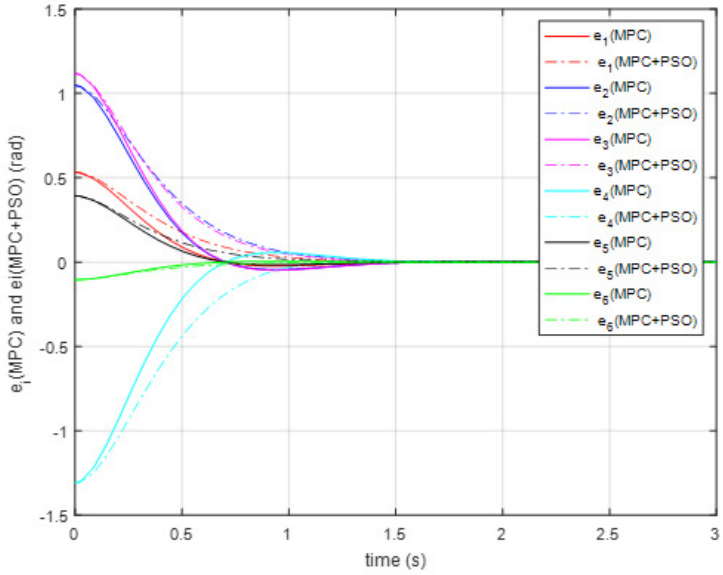


Figure 4.14: Articulations Errors

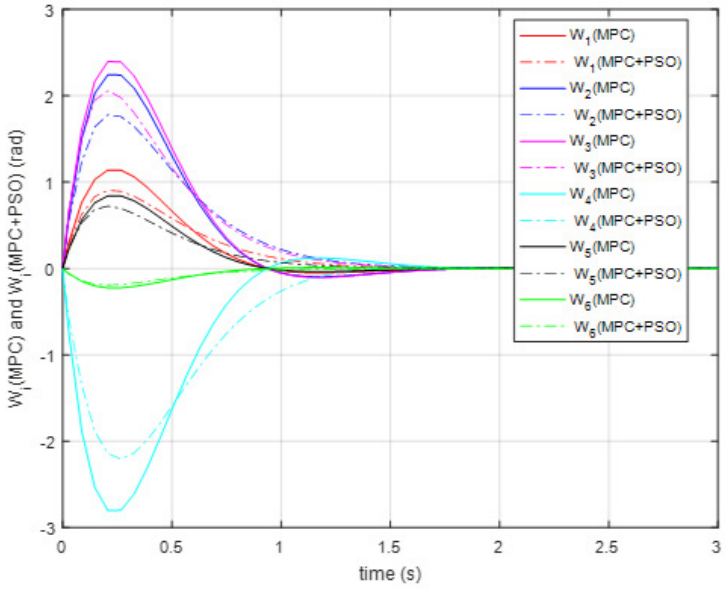


Figure 4.15: Angular Velocities

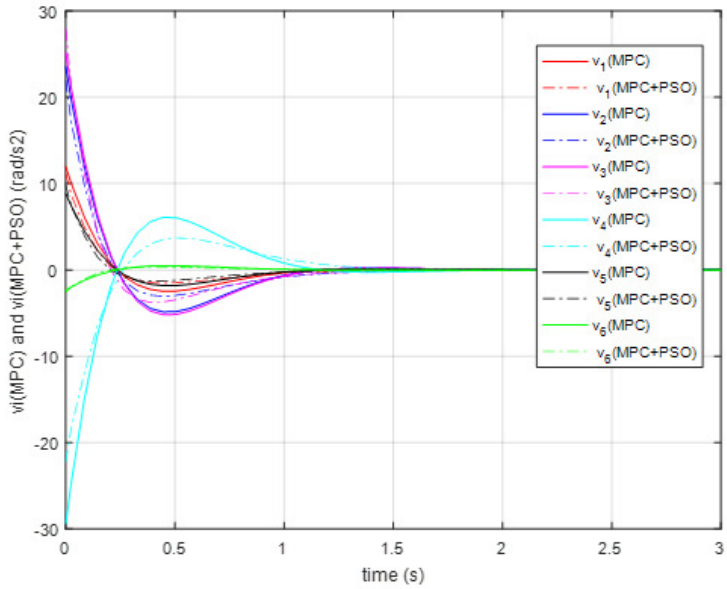


Figure 4.16: Synthetic Controls

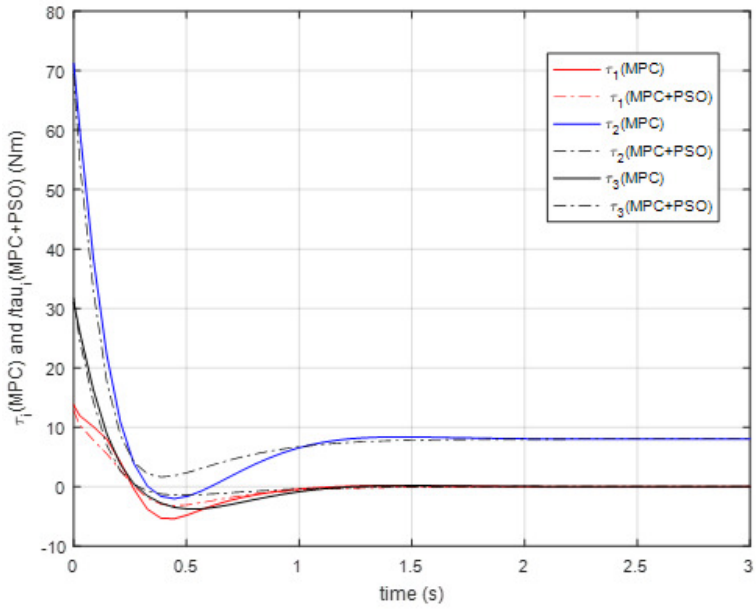


Figure 4.17: Manipulator robot Torques τ_1,τ_2,τ_3

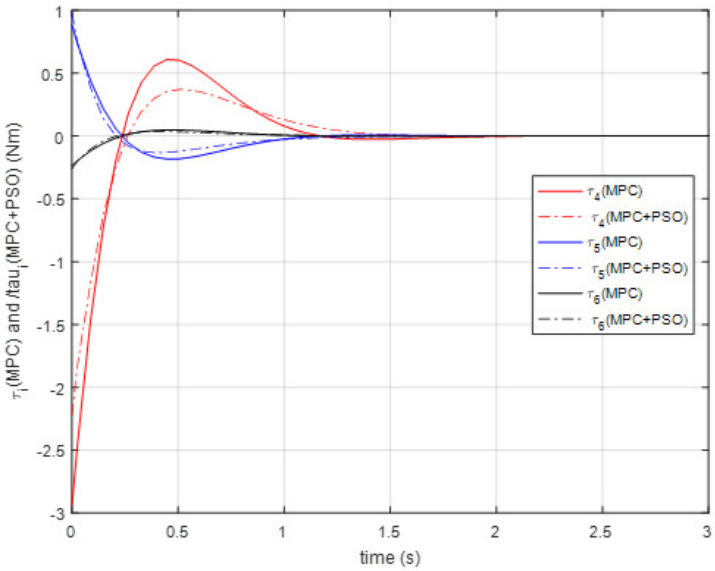


Figure 4.18: Manipulator robot Torques τ_4,τ_5,τ_6

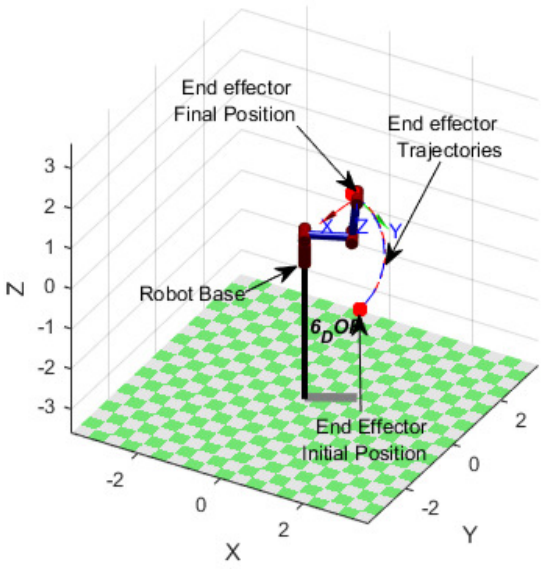


Figure 4.19: Arm Manipulator End Effector in the Final Position

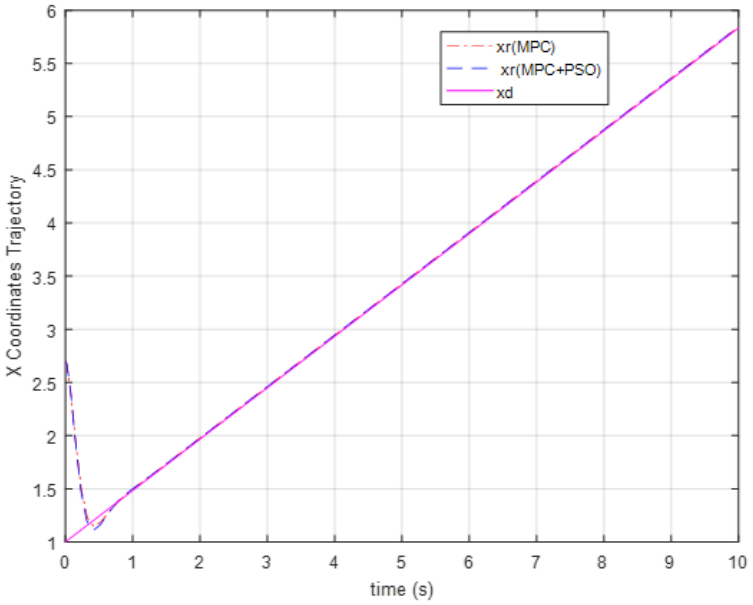


Figure 4.20: Desired and Real Xc

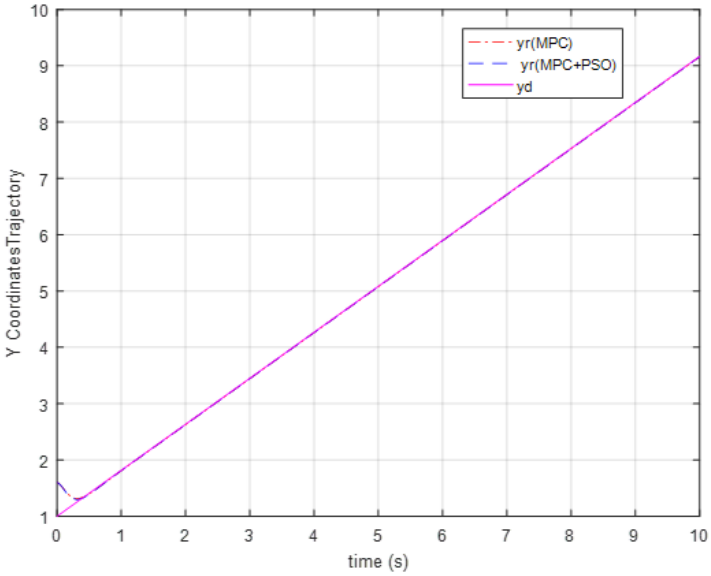


Figure 4.21: Desired and Real Yc

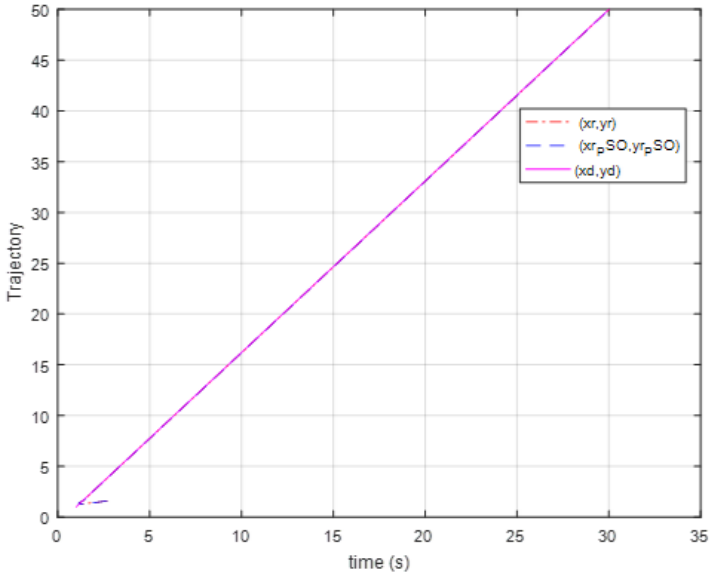


Figure 4.22: Desired and Real Trajectory

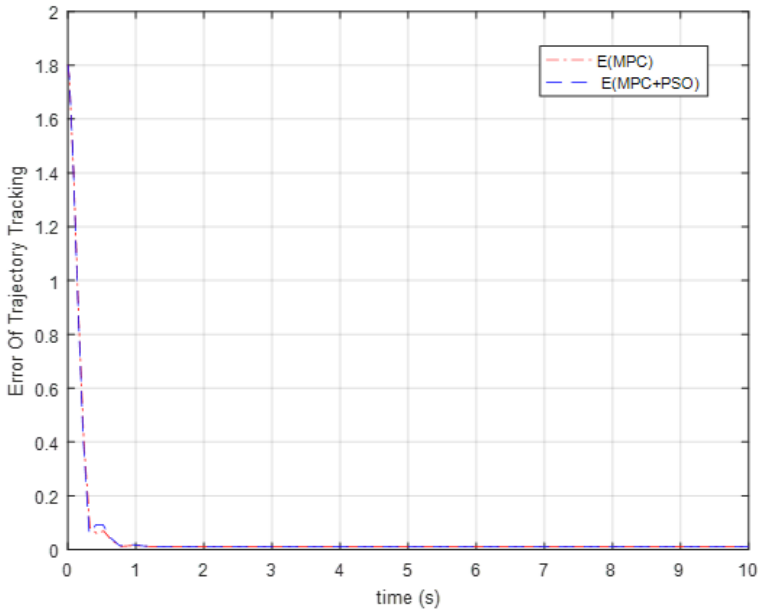


Figure 4.23: Trajectory tracking

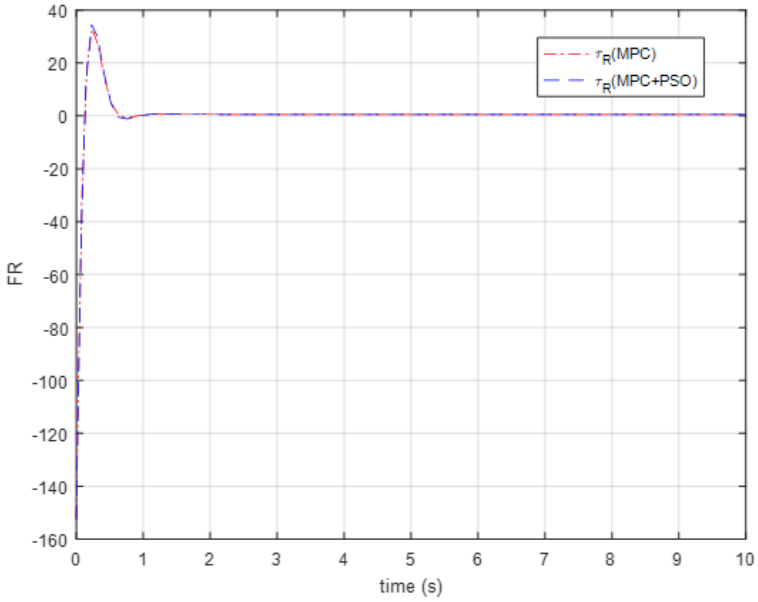


Figure 4.24: R-Wheels acting Torques

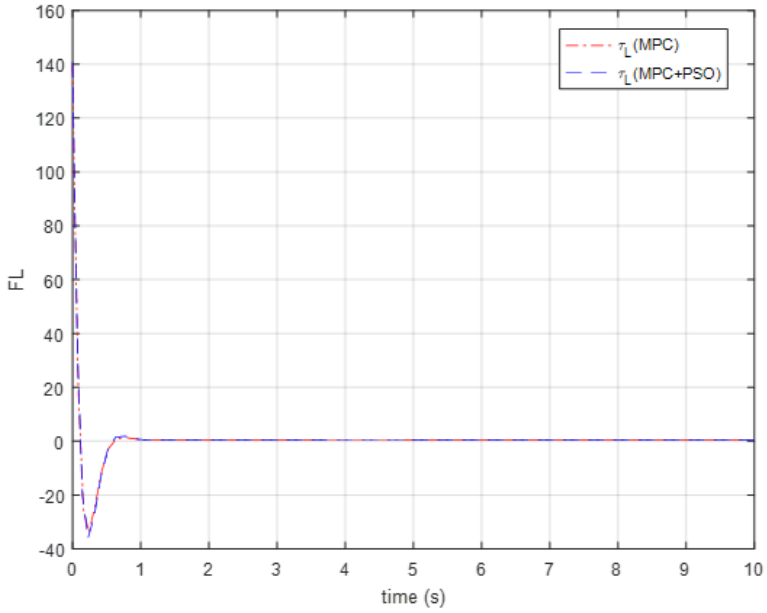


Figure 4.25: L-Wheels acting Torques

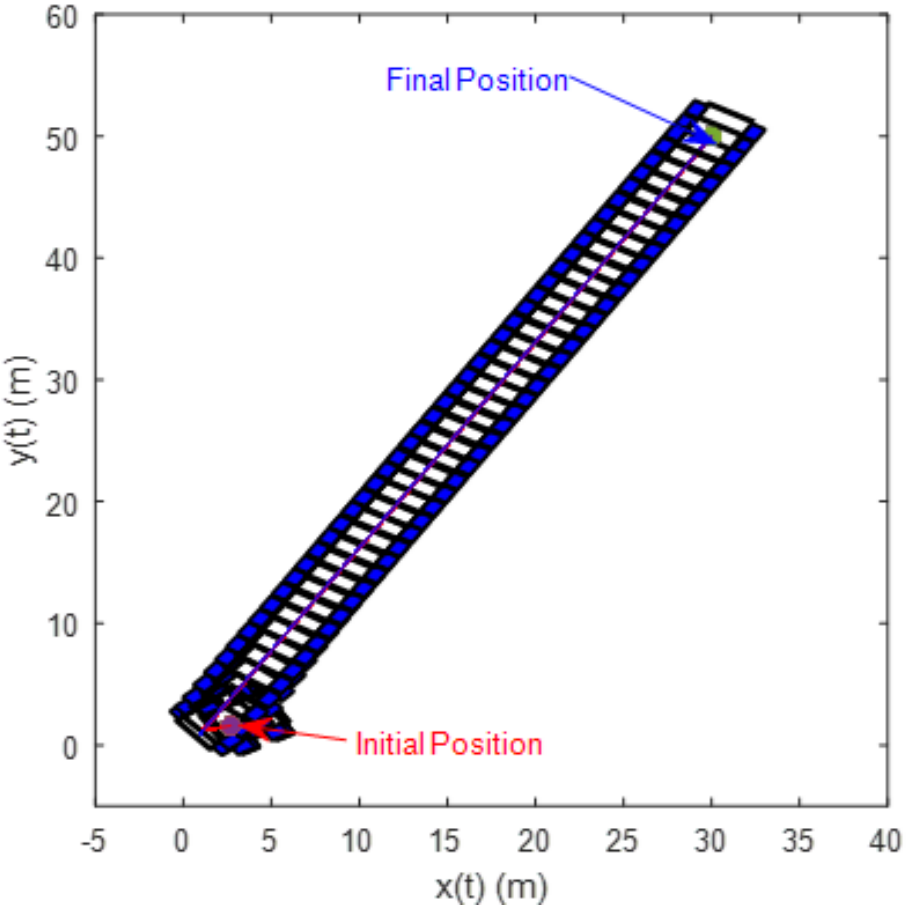


Figure 4.26: DDMR Trajectory simulation

4.8 Conclusion

In this chapter, we proposed a constrained particle swarm optimization (COPSO) technic to better tune the predictive controller gains for a six degrees of freedom manipulator arm as well as for the differential-drive mobile robot. We started by linearizing both nonlinear dynamic models by using a feedback linearization technic. Once the linear model was obtained, a model predictive control is designed in the next step. The prediction horizon time h and the weight factor ρ are tuned based on constrained particle swarm optimization. The simulation results for the manipulator arm show that the COPSO algorithm contrasted with graphic analysis method, has many advantages, such as the fast response time, minimum overshoot and optimum allowed torque for the robot. It shows that the optimized MPC parameters based on COPSO algorithm provides better results than the optimized MPC based on graphical method. While for the DDMR simulations results of both tuning methods are almost the same, and this is probably to the trajectory generator constraints, which impose to have unique prediction time for both X and Y coordinates.

General conclusion and perspectives

Conclusions

There are many constraints and challenges that control engineer has to take into account while looking or developing an adequate controller, such that:

1- Nonlinear models (kinematic and dynamic) of the mobile robots, which present the most difficult problem prior to develop a suitable control strategy.

2- Provides excellent dynamic performances: short response time, minimum overshooting and less oscillation.

3- Respect the mobile robot workspace constraints: minimum torque and shortest trajectory.

4- Maintain the above features for a higher degree of freedom mobile robot.

Therefore our proposed control approach is a combination of a feedback linearization control, in order to transform the nonlinear dynamics of the system to a complete or partial linear one, and the MPC control, to obtain better performance for the system.
To resolve the first problem and get a linearized dynamic models, we have used the feedback linearization control; which is obtained by differentiating the output vector until the control inputs appears.For manipulator arm, we have applied the input-state feedback linearization technique, where we could linearize the map between the transformed inputs and the entire state vector, while for the differential-drive mobile robot, due to the non-holonomic constraints, the above technic was not possible, therefore we have used input-output feedback linearization technique, where we could find a linear function between the input state vector and the selected output vector.

To respond the second problem and ensure an excellent dynamic performances, we have defined one-horizon time quadratic objective function in order to find the optimized control input, where the parameters (h and ρ) of the obtained control law are calculated to have a specific behavior of the second order closed loop system.

Then to overcome the lats problem and respect the workspace constraints we have introduced the following two methods:

1-Graphic tool: which consist of drawn the variation of the maximum, minimum torques and settling time curves as a function of the natural frequency, then we have compare it with the imposed dynamics and constraints in order to take the trade-off value.

2- Constraint particle swarm optimization algorithm: the idea here is to use the COPSO to find the optimum value of the obtained MPC gains, then by identifying these optimal gains with the MPC gains expression, we can calculate the parameters h and ρ.

Finally to verify and make the proposed control capability of maintaining the above features, we have successfully validate it with a comparative study with linear quadratic control, then we have extend it for the below higher degrees of freedom robots:

A- 3-DOF manipulator arm.

B- Differential-drive mobile robot.

C- 6-DOF mobile manipulator robot.

The obtained simulations results showed the efficiency of the proposed control approach and all the workspace constraints are well respected.

Future Work

Future work is aimed towards to:

• To test the proposed MPC control on a real robot, to verify this method's robustness

• Take the interaction between the manipulator arm and the mobile base during the motion of the hole mobile manipulator robot in order to study the effect of the manipulator arm on the stability of entire system.

• To develop a predictive control approach in a discrete domain for controlling a mobile manipulator robot and challenge to obtain a predictive control law for following more complicated predefined trajectory (in the case of obstacle avoidance for example).

• Apply the COPSO based tuning of an MPC control for controlling different mobile robots, such as: robot like a car, tri-cycle mobile robot. . . .etc.

• To propose a predictive control architecture with observer to take into account the measurement noise

References

[1] Yoshio Yamamoto and Xiaoping Yun. Coordinating locomotion and manipulation of a mobile manipulator. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 2643–2648. IEEE, 1992.

[2] MJ Er. Recent developments and futuristic trends in robot manipulator control. In *Proceedings 1993 Asia-Pacific Workshop on Advances in Motion Control*, pages 106–111. IEEE, 1993.

[3] I David and G Robles. Pid control dynamics of a robotic arm manipulator with two degrees of freedom. *Control de Procesos y Robotica*, pages 1–7, 2012.

[4] Alain Liégeois. Modélisation et commande des robots manipulateurs. *Techniques de l'Ingénieur, S*, 7:730, 2000.

[5] Nasr A Elkhateeb and Ragia I Badr. Novel pid tracking controller for 2dof robotic manipulator system based on artificial bee colony algorithm. *Electrical, Control and Communication Engineering*, 13(1):55–62, 2017.

[6] F Sado, Shahrul Na'im Sidek, and Hazlina M Yusuf. Independent joint control of a 3-dof robotic system using pi controller. In *2014 International Conference on Computer and Communication Engineering*, pages 115–118. IEEE, 2014.

[7] Tien C Hsia, Ty A Lasky, and ZY Guo. Robust independent robot joint control: Design and experimentation. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1329–1334. IEEE, 1988.

[8] Daniel E Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 10(2):47–53, 1969.

[9] J Luh, M Walker, and R Paul. Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, 25(3):468–474, 1980.

[10] Fabrizio Caccavale, Stefano Chiaverini, and Bruno Siciliano. Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: Theory and experiments. *IEEE/ASME Transactions on Mechatronics*, 2(3):188–194, 1997.

[11] Chien-Chern Cheah, Masanori Hirano, Sadao Kawamura, and Suguru Arimoto. Approximate jacobian control for robots with uncertain kinematics and dynamics. *IEEE transactions on robotics and automation*, 19(4):692–702, 2003.

[12] P Falb and William Wolovich. Decoupling in the design and synthesis of multivariable control systems. *IEEE transactions on automatic control*, 12(6):651–659, 1967.

[13] E Freund. Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators. *The International Journal of Robotics Research*, 1(1):65–78, 1982.

[14] Qing-Guo Wang. *Decoupling control*, volume 285. Springer Science & Business Media, 2002.

[15] Frank L Lewis, Darren M Dawson, and Chaouki T Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[16] Fu Ks, RC Gonzalez, and CSG Lee. Robotics control, sensing, vision and intelligence, 1987.

[17] Homayoun Seraji. Decentralized adaptive control of manipulators: theory, simulation, and experimentation. *IEEE Transactions on Robotics and Automation*, 5(2):183–201, 1989.

[18] Chutiphon Pukdeboon. A review of fundamentals of lyapunov theory. *The Journal of Applied Science*, 10(2):55–61, 2011.

[19] Nur Uddin. Lyapunov-based control system design of two-wheeled robot. In *2017 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 121–125. IEEE, 2017.

[20] J Anthony Rossiter. *Model-based predictive control: a practical approach*. CRC press, 2003.

[21] Grüne Lars and P Jürgen. Nonlinear model predictive control theory and algorithms, 2011.

[22] Vishank Bhatia, RK Ganesh Ram, V Kalaichelvi, and R Karthikeyan. Application of model predictive controller for 2-dof robot manipulator. In *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, pages 1–5. IEEE, 2015.

[23] JA Rossiter. *A first course in predictive control*. CRC press, 2018.

[24] Charles R Cutler and Brian L Ramaker. Dynamic matrix control?? a computer control algorithm. In *joint automatic control conference*, number 17, page 72, 1980.

[25] Charles Ray Cutler. Dynamic matrix control: An optimal multivariable control algorithm with constraints. 1984.

[26] David W Clarke, Coorous Mohtadi, and PS Tuffs. Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[27] David W Clarke, Coorous Mohtadi, and PS Tuffs. Generalized predictive control—part ii extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[28] EF12068240775 Camacho. Constrained generalized predictive control. *IEEE transactions on automatic control*, 38(2):327–332, 1993.

[29] RMC De Keyser and AR Van Cauwenberghe. Extended prediction self-adaptive control. *IFAC Proceedings Volumes*, 18(5):1255–1260, 1985.

[30] Jacques Richalet and Donal O'Donovan. *Predictive functional control: principles and industrial applications*. Springer Science & Business Media, 2009.

[31] Ronald Soeterboek. *Predictive control: a unified approach*. Prentice-Hall, Inc., 1992.

[32] Alberto Bemporad, Francesco Borrelli, Manfred Morari, et al. Model predictive control based on linear programming˜ the explicit solution. *IEEE transactions on automatic control*, 47(12):1974–1985, 2002.

[33] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser, 2012.

[34] Květoslav Belda and Oliver Rovnỳ. Predictive control of 5 dof robot arm of autonomous mobile robotic system motion control employing mathematical model of the robot arm dynamics. In *2017 21st International Conference on Process Control (PC)*, pages 339–344. IEEE, 2017.

[35] Manel Mendili and Faouzi Bouani. Predictive control of mobile robot using kinematic and dynamic models. *Journal of Control Science and Engineering*, 2017, 2017.

[36] Mohamed A Kamel and Youmin Zhang. Linear model predictive control via feedback linearization for formation control of multiple wheeled mobile robots. In *2015 IEEE International Conference on Information and Automation*, pages 1283–1288. IEEE, 2015.

[37] Eric Ostertag. *Mono-and multivariable control and estimation: linear, quadratic and LMI methods*. Springer Science & Business Media, 2011.

[38] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[39] Barış Ata and Ramazan Coban. Linear quadratic optimal control of an inverted pendulum on a cart using artificial bee colony algorithm: an experimental study. *Cukurova University Journal of the Faculty of Engineering and Architecture*, 32(2):109–123, 2017.

[40] Zhi Liu, Ci Chen, Yun Zhang, and CL Philip Chen. Coordinated fuzzy control of robotic arms with actuator nonlinearities and motion constraints. *Information Sciences*, 296:1–13, 2015.

[41] Tahar Guesbaya. *Identification et commande de robot manipulateur rigide et flexible en utilisant les réseaux de neurones et la logique floue*. PhD thesis, Université Mohamed Khider-Biskra, 2012.

[42] Mohammad Javad Mahmoodabadi and A Ziaei. Inverse dynamics based optimal fuzzy controller for a robot manipulator via particle swarm optimization. *Journal of Robotics*, 2019, 2019.

[43] Christopher Edwards and Sarah Spurgeon. *Sliding mode control: theory and applications*. Crc Press, 1998.

[44] Giorgio Bartolini, Leonid Fridman, Alessandro Pisano, and Elio Usai. *Modern sliding mode control theory: New perspectives and applications*, volume 375. Springer, 2008.

[45] El-Hadi Guechi, Samir Bouzoualegh, Lotfi Messikh, and Sašo Blažic. Model predictive control of a two-link robot arm. In *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pages 409–414. IEEE, 2018.

[46] El-Hadi Guechi, Samir Bouzoualegh, Youcef Zennir, and Sašo Blažič. Mpc control and lq optimal control of a two-link robot arm: A comparative study. *Machines*, 6(3):37, 2018.

[47] Samir Bouzoualegh, El-Hadi Guechi, and Youcef Zennir. Model predictive control of a three degrees of freedom manipulator robot. In *2019 International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pages 84–89. IEEE, 2019.

[48] Samir Bouzoualegh, El-Hadi Guechi, and Ridha Kelaiaia. Model predictive control of a differential-drive mobile robot. *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, 10(1):20–41, 2018.

[49] Min Zheng, Wudai Liao, Changhai Yin, and Aihui Wang. Nonlinear tracking control design of a robot arm using robust right coprime factorization and sliding mode approaches. In *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pages 11–16. IEEE, 2014.

[50] Vanni Zanotto, Alessandro Gasparetto, Albano Lanzutti, Paolo Boscariol, and Renato Vidoni. Experimental validation of minimum time-jerk algorithms for industrial robots. *Journal of Intelligent & Robotic Systems*, 64(2):197–219, 2011.

[51] Qing Guo, Tian Yu, and Dan Jiang. Robust h_∞ positional control of 2-dof robotic arm driven by electro-hydraulic servo system. *ISA transactions*, 59:55–64, 2015.

[52] Erol Özgür and Youcef Mezouar. Kinematic modeling and control of a robot arm using unit dual quaternions. *Robotics and Autonomous Systems*, 77:66–73, 2016.

[53] Elisha D Markus, John T Agee, and Adisa A Jimoh. Flat control of industrial robotic manipulators. *Robotics and Autonomous systems*, 87:226–236, 2017.

[54] Jacob Wilson, Meaghan Charest, and Rickey Dubay. Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator. *Robotics and Computer-Integrated Manufacturing*, 41:23–30, 2016.

[55] Tomasz Rybus, Karol Seweryn, and Jurek Z Sasiadek. Control system for free-floating space manipulator based on nonlinear model predictive control (nmpc). *Journal of Intelligent & Robotic Systems*, 85(3-4):491–509, 2017.

[56] Rahma Boucetta. Generalized predictive control for a flexible single-link manipulator. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pages 499–510. Springer, 2013.

[57] Paolo Boscariol, Alessandro Gasparetto, and Vanni Zanotto. Model predictive control of a flexible links mechanism. *Journal of Intelligent and Robotic Systems*, 58(2):125–147, 2010.

[58] Michiel Plooij, Wouter Wolfslag, and Martijn Wisse. Robust feedforward control of robotic arms with friction model uncertainty. *Robotics and Autonomous systems*, 70:83–91, 2015.

[59] Richard S Hartenberg and Jacques Denavit. A kinematic notation for lower pair mechanisms based on matrices. *Journal of applied mechanics*, 77(2):215–221, 1955.

[60] Pradip N Sheth and JJ Uicker Jr. A generalized symbolic notation for mechanisms. 1971.

[61] Marc Renaud. *Contribution a l'etude de la modélisation et de la commande des systèmes mécaniques articulés*. PhD thesis, Université Paul Sabatier, 1975.

[62] Wisama Khalil. *Modélisation et commande par calculateur du manipulateur MA 23: extension à la conception par ordinateur des manipulateurs*. PhD thesis, Université des Sciences et Techniques du Languedoc, 1976.

[63] Paul Borrel. *Modèles de comportement de manipulateurs: application à l'analyse de leurs performances et à leur commande automatique*. PhD thesis, Université des Sciences et Techniques du Languedoc, 1979.

[64] John J Craig. *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[65] Cyril Quennouelle. Modelisation geometrico-statique des mecanismes paralleles compliants. 2009.

[66] Wisama Khalil and J Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179. IEEE, 1986.

[67] Wisama Khalil and Etienne Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[68] Richard P Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[69] Donald L Peiper. The kinematics of manipulators under computer control. Technical report, Stanford univ ca dept of computer science, 1968.

[70] Madhusudan Raghavan and Bernard Roth. Inverse kinematics of the general 6r manipulator and related linkages. 1993.

[71] Rached Dhaouadi and A Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics & Automation*, 2(2):1–7, 2013.

[72] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, volume 118. Springer, 2017.

[73] Zhijun Li and Shuzhi Sam Ge. *Fundamentals in modeling and control of mobile manipulators*, volume 49. CRC Press, 2013.

[74] ASHOK Sapra, ISHAN Gupta, VIPIN Kumar, ANUJ Khare, TARUN Wig, and MONIKA Chawla. Robotics: Appin knowledge solutions, 2007.

[75] Shuai Li, Long Jin, and Mohammed Aquil Mirza. *Kinematic Control of Redundant Robot Arms Using Neural Networks*. John Wiley & Sons, 2019.

[76] Carl D Crane III and Joseph Duffy. *Kinematic analysis of robot manipulators*. Cambridge University Press, 2008.

[77] Saeed B Niku. *Introduction to robotics: analysis, control, applications*. John Wiley & Sons, 2020.

[78] Christine Chevallereau and Wisama Khalil. Efficient method for the calculation of the pseudo inverse kinematic problem. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1842–1848. IEEE, 1987.

[79] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.

[80] Mohsen Shahinpoor and Mo Shahinpoor. *A robot engineering textbook*. Harper & Row New York, 1987.

[81] KS Fu, RC Gonzalez, and CS Lee. G. robotics: Control, sensing, vision and intelligence. *New York: Mc Graw-Hill*, pages 78–82, 1987.

[82] Roy Featherstone. The calculation of robot dynamics using articulated-body inertias. *The international journal of robotics research*, 2(1):13–30, 1983.

[83] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[84] Farzin Piltan, MohammadHossain Yarmahmoudi, Mina Mirzaie, Sara Emamzadeh, and Zahra Hivand. Design novel fuzzy robust feedback linearization control with application to robot manipulator. *International Journal of Intelligent Systems and Applications*, 5(5):1, 2013.

[85] Lalo Magni, Riccardo Scattolini, and Karl Johan Åström. Global stabilization of the inverted pendulum using model predictive control. *IFAC Proceedings Volumes*, 35(1):141–146, 2002.

[86] Adam Mills, Adrian Wills, and Brett Ninness. Nonlinear model predictive control of an inverted pendulum. In *2009 American control conference*, pages 2335–2340. IEEE, 2009.

[87] Péter Bauer and József Bokor. Development and performance evaluation of an infinite horizon lq optimal tracker. *European Journal of Control*, 39:8–20, 2018.

[88] Lorenzo Ntogramatzidis and Augusto Ferrante. On the solution of the riccati differential equation arising from the lq optimal control problem. *Systems & Control Letters*, 59(2):114–121, 2010.

[89] Radu-Emil Precup, Plamen Angelov, Bruno Sielly Jales Costa, and Moamar Sayed-Mouchaweh. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry*, 74:75–94, 2015.

[90] Radu-Emil Precup, Marius-Csaba Sabau, and Emil M Petriu. Nature-inspired optimal tuning of input membership functions of takagi-sugeno-kang fuzzy models for anti-lock braking systems. *Applied Soft Computing*, 27:575–589, 2015.

[91] MIT Open Course Ware. Dynamics and control, 2008.

[92] El-Hadi Guechi, Alexandre Abellard, and Patrick Abellard. Ts-fuzzy predictor observer design for trajectory tracking of wheeled mobile robot. In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pages 319–324. IEEE, 2011.

[93] Mendili Manel and Bouani Faouzi. Predictive control based on dynamic modeling of omnidirectional mobile robot. In *2017 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE, 2017.

[94] Spyros Maniatopoulos, Dimitra Panagou, and Kostas J Kyriakopoulos. Model predictive control for the navigation of a nonholonomic vehicle with field-of-view constraints. In *2013 American Control Conference*, pages 3967–3972. IEEE, 2013.

[95] Z Sinaeefar and M Farrokhi. Adaptive fuzzy model-based predictive control of nonholonomic wheeled mobile robots including actuator dynamics. *Int. Journal of Scientific & Eng. Research*, 3(9), 2012.

[96] Alicja Mazur. Hybrid adaptive control laws solving a path following problem for non-holonomic mobile manipulators. *International Journal of Control*, 77(15):1297–1306, 2004.

[97] Srdjan T Mitrovic and Zeljko M Djurovic. Fuzzy-based controller for differential drive mobile robot obstacle avoidance. *IFAC Proceedings Volumes*, 43(16):67–72, 2010.

[98] El-Hadi Guechi. *Suivi de trajectoires d'un robot mobile non holonome: approche par modèle flou de Takagi-Sugeno et prise en compte des retards*. PhD thesis, 2010.

[99] Robert L Williams, Brian E Carter, Paolo Gallina, and Giulio Rosati. Dynamic model with slip for wheeled omnidirectional robots. *IEEE transactions on Robotics and Automation*, 18(3):285–293, 2002.

[100] Hajime Asama, Masatoshi Sato, Hayato Kaetsu, Koichi Ozaki, Akihiro Matsumoto, and Isao Endo. Development of an omni-directional mobile robot with 3 dof decoupling drive mechanism. *Journal of the Robotics Society of Japan*, 14(2):249–254, 1996.

[101] Phillip John McKerrow and Phillip McKerrow. *Introduction to robotics*. Addison-Wesley Sydney, 1991.

[102] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[103] Woojin Chung and Karl Iagnemma. Wheeled robots. In *Springer Handbook of Robotics*, pages 575–594. Springer, 2016.

[104] Mihai Olimpiu Tătar, Florin Haiduc, and Dan Mândru. Design of a synchro-drive omnidirectional mini-robot. In *Solid State Phenomena*, volume 220, pages 161–167. Trans Tech Publ, 2015.

[105] Peter J Gawthrop and Liuping Wang. Intermittent predictive control of an inverted pendulum. *Control Engineering Practice*, 14(11):1347–1356, 2006.

[106] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[107] SYS Hussien, HI Jaafar, NA Selamat, FS Daud, and AFZ Abidin. Pid control tuning via particle swarm optimization for coupled tank system. *International Journal of Soft Computing and Engineering (IJSCE)*, 4(2):202–206, 2014.

[108] Yun-Hyung Lee, Ki-Tak Ryu, Jae-Jung Hur, and Myung-Ok So. Pso based tuning of pid controller for coupled tank system. , 38(10):1297–1302, 2014.

[109] Zafer Bingül and Oğuzhan Karahan. A fuzzy logic controller tuned with pso for 2 dof robot trajectory control. *Expert Systems with Applications*, 38(1):1017–1031, 2011.

[110] Nasr A Elkhateeb and RI Badr. Dynamic inertia weight artificial bee colony versus ga and pso for optimal tuning of pid controller. *International Journal of Modelling, Identification and Control 5*, 22(4):307–317, 2014.

[111] Xiangyong Li, Peng Tian, and Min Kong. A novel particle swarm optimization for constrained optimization problems. In *Australasian Joint Conference on Artificial Intelligence*, pages 1305–1310. Springer, 2005.

[112] Konstantinos E Parsopoulos, Michael N Vrahatis, et al. Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, 76(1):214–220, 2002.

[113] Xiaohui Hu and Russell Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the sixth world multiconference on systemics, cybernetics and informatics*, volume 5, pages 203–206. Citeseer, 2002.

[114] Guo Zhengxiong and Ge Xinsheng. A particle swarm optimization for the motion planning of wheeled mobile robot. In *2010 8th World Congress on Intelligent Control and Automation*, pages 2410–2414. IEEE, 2010.

[115] Sedigheh Ahmadzadeh and Mehdi Ghanavati. Navigation of mobile robot using the pso particle swarm optimization. *Journal of Academic and Applied Studies (JAAS)*, 2(1):32–38, 2012.

[116] Ghusn A Ibraheem and Ibraheem K Ibraheem. Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order pid controller. *Engineering and Technology Journal*, 34(13 Part (A) Engineering):2406–2419, 2016.