Master Thesis

Model-Based Stability Analysis for Mobile Manipulators

OTH
OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

RRRU
REGENSBURG ROBOTICS
RESEARCH UNIT

at the Faculty of Mechanical Engineering
of the Ostbayerische Technische Hochschule Regensburg
in the course of studies Industrial Engineering
under supervision of Prof. Dr.-Ing. Thomas Schlegl

of

Mr. Marco Bertagnoli
Date of birth: 15.06.1992
Matriculation No.: 2897292

Via G. Muraro, 2
37020 Cerro Veronese, Verona (Italy)
Email: marcobertagnoli@gmail.com
Tel.: +39 3407739251

Co-supervison: Mr M.Sc. Oliver Kotz
Beginning: 10.10.2016
End: 31.06.2017

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

REGENSBURG ROBOTICS
RESEARCH UNIT

Labor Robotik
Prof. Dr.-Ing. Thomas Schlegl

Masterarbeit für Herr Marco Bertagnoli
Matrikelnummer: 2897292
Industrial Engineering Master

Model-Based Stability Analysis for Mobile Manipulators

Problem description

Mobile manipulators, which consist of an autonomous mobile platform and one or more manipulators, have attracted much interest, because of their extended workspace, with respect to conventional, fixed base manipulators. When they carry heavy payloads, e.g. in intralogistic tasks, move over irregular terrains or get sudden external disturbances, they may experience postural instabilities. The underlying idea for controlling the stability of such systems, follows the model of humans and other animals, which are intrinsically dynamically stable. However in the literature, there are some criteria which can be effectively used for the tip-over stability measure of both, legged robots and mobile manipulators.

In this work several parts of a robot stability analysis have to be treated on the example of a mobile manipulator developed in the RRRU laboratory, composed by Adept Lynx$^{\text{TM}}$mobile platform and gomTec Roberta P80.800 manipulator. Hence, the focus of this treatment lies on finding a stability monitor methodology, in order to identify and forbid critical movements of the center of mass of the system, mainly related to the motion of the manipulator and excessive payload. Information regarding the manipulator such as direct and inverse kinematics and dynamics equations are given by previous work. A tip-over avoidance methodology will be employed on the control of the system, taking the combined dynamics of platform, manipulator and payload into account. The final methodology, will be evaluated on defined use cases and then, a possible stability enhancement strategy, according with the available data, will conclude the work.

Tasks

- Literature research regarding manipulator dynamics and tip-over avoidance for mobile platforms
- Derivation of the combined dynamics for platform, manipulator and external loads
- Development of a tip-over avoidance methodology
- Verification and validation of all proposed methods through simulations and experiments on the real system

Related literature

- On the dynamic tip-over stability of wheeled mobile manipulators [1]
- Tip-Over Stability of Omni-Directional Mobile Robot [2]
- A New Measure of Tipover Stability Margin for Mobile Manipulators [3]

Supervision: Mr M.Sc. Oliver Kotz Period of time: 10.10.2016–31.06.2017

Abstract

This thesis deals with a tip-over analysis for mobile manipulators, considering the prototype developed by the Regensburg Robotics Research Unit (RRRU), composed of the differentially driven Adept Lynx autonomous platform, and the lightweight robot gomTec Roberta P80.800. This work will set the basis for realizing intelligent applications, by taking care of the robot safe state.

Hence, a stability monitoring feature is going to be suggested, that yields a stability measure, evaluated before the motion starts on the wished trajectory to perform, in a way that the user can be alerted before the tip-over occurrence. In order to accomplish this, a model based strategy is necessary to be applied in advance, on simulations of the mobile platform behavior.

Because of the restricted available information about the Roberta dynamic parameters, the Moment-Height Stability measure is the applicable algorithm taken into account, that let us design a stability measure based on the manipulator base reaction forces (base wrench), considered as interaction forces, between the upper Roberta robot and the Adept Lynx at its base. Therefore, figuring out a solution for the base wrench dynamics for the mobile manipulator in study, is the main challenge of this work. For this reason, parts of previous work performed in the RRRU, in particular the identified dynamic model of the Roberta P80.800, are adapted to obtain a reliable formulation for the base wrench, avoiding a complete identification procedure.

Contents

List of Figures xi

List of Tables xiii

1 Introduction 1
1.1 Problem Description . . . 2
1.2 State of the Art . . . 3
1.3 Goal of the Work . . . 4
1.4 Structure of this Thesis . . . 5

2 Kinematics 7
2.1 Kinematics of a Serial Link Manipulator . . . 7
  2.1.1 Transformation of Frames and Vectors . . . 7
  2.1.2 Direct Kinematic Model . . . 8
  2.1.3 Recursive Computation of Link Velocities . . . 11
  2.1.4 Inverse Kinematic Model . . . 12
2.2 Kinematics of a Differential Mobile Platform . . . 13
  2.2.1 Adept Lynx Autonomous Mobile Platform . . . 13
  2.2.2 Cartesian Geometric Modeling . . . 15
  2.2.3 Non-Holonomic Constraint . . . 18
  2.2.4 Polar Geometric Modeling . . . 19

3 Dynamics 21
3.1 Transformation of Force and Torque . . . 21
3.2 Dynamic Parameters of a Rigid Body . . . 22
3.3 Dynamics for the Roberta P80.800 . . . 23
  3.3.1 Newton-Euler Formulation . . . 23
  3.3.2 Moving Base Manipulator . . . 26
  3.3.3 Inverse Dynamics Linear in the Base Parameters . . . 26
  3.3.4 Base Wrench . . . 27
  3.3.5 Base Wrench Dynamics Verification . . . 27
  3.3.6 Linear Expressions for the Base Wrench . . . 27
3.4 Theory of Parameter Identification . . . 29
  3.4.1 Least Squares Method . . . 30
  3.4.2 LS Estimator for Linear Models . . . 31
3.5 Base Wrench Adjustment . . . 32
  3.5.1 Systematic Error Compensation . . . 32
  3.5.2 Estimation of the Non Identified Parameters . . . 34
3.6 Adept Lynx Dynamics . . . 35
3.7 Lagrangian Approach . . . 35
  3.7.1 Payload Integration . . . 37
3.8 Newton-Euler Based Approach . . . 37

4 Base Wrench Validation 43
4.1 Preliminary Transformations . . . 43
4.2 Optimal trajectory . . . 44
4.3 Base Parameter Set Results . . . 45
4.4 Results with the Systematic Error Compensation . . . 46
4.5 Results with the Parameter Identification . . . 48
4.6 Cross Validation on Different Trajectories . . . 51

5 Tip-Over Analysis 55
5.1 Moment-Height Stability Measure Introduction . . . 55
5.1.1 Geometric Model . . . 56
5.2 MHS Algorithm for a Stationary Base . . . 57
5.3 MHS Algorithm for a Moving Base . . . 59
5.4 MHS Verification . . . 60
5.4.1 Roberta Clockwise Rotation . . . 60
5.5 Application on Demo-RedBox Trajectory . . . 62
5.6 Offline Tip-over Evaluation . . . 63

6 Conclusions and Future Perspectives 65

7 Bibliography 67

Appendices I

A Cross Correlation Plots III

B Lower Subsystem Mass Integration in (5.8) VII

C Base Principle of MHS Measure IX

List of Figures

1.1 Mobile robot and manipulator example. 1
1.2 Mobile manipulator developed by RRRU. 2

2.1 Transformation of a point. Figure from [4]. 8
2.2 Frames according to modified Denavit-Hartenberg convention, $\boldsymbol{q} = \mathbf{0}$ [4]. . 10
2.3 Augmented velocities of link $i-1$ and i [4]. 11
2.4 Adept Lynx Autonomous Intelligent Vehicle. 13
2.5 Platform dimensions in mm. 14
2.6 Center of Gravity of the platform without payload structure. Units in mm. 15
2.7 Global and local reference frames. 16
2.8 Platform with parameters for polar modeling. 20

3.1 Link i moving in space [4]. 22
3.2 Forces and moments on link i. 24
3.3 Derivation of $\boldsymbol{\varepsilon}$. 29
3.4 Systematic error adjustment scheme. 33
3.5 Non identified parameters contribution is now taken into account. 34
3.6 Lynx platform parameters and frames for Newtonian dynamic modeling. . 38

4.1 Base wrench validation scheme. 43
4.2 Base wrench simulations and measurements, w.r.t. their reference frame. . 44
4.3 Base wrench force components, simulated with known parameters in $\boldsymbol{\kappa}$, are compared with the measurements. The plot shows 5 periods. 46
4.4 Base wrench moment components, simulated with known parameters in $\boldsymbol{\kappa}$, are compared with the measurements. The plot shows 5 periods. 47
4.5 Base wrench moment components, simulated with the $\boldsymbol{\varepsilon}_1$ error compensation in (3.65), are compared with the measurements. The plot shows 5 periods. 47
4.6 Base wrench force components, simulated with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods. 49
4.7 Base wrench moment components, simulated with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods. 50
4.8 Comparison between errors $\boldsymbol{\varepsilon}_{in}$ and $\boldsymbol{\varepsilon}_{fin}$, depicted in half of a period. . . . 50
4.9 Comparison between errors $\boldsymbol{\varepsilon}_{in}$ and $\boldsymbol{\varepsilon}_{fin}$, depicted in half of a period. . . . 51
4.10 Actual position and acceleration of the Roberta joints, under the optimal trajectory: 5 periods are shown. 53
4.11 Actual position and acceleration of the Roberta joints, under the PTP and LIN. The entire trajectory is shown. 53

4.12 Wrench force components, simulated on Demo Red Box trajectory with error compensation in (3.72), are compared with the measures. 5 periods are shown. . . . 54
4.13 Wrench moment components, simulated on Demo Red Box trajectory with error compensation in (3.72), are compared with the measures. 5 periods are shown. . . . 54

5.1 Upper and lower subsystems with the interaction forces. . . . 55
5.2 Lower subsystem reference frames. . . . 56
5.3 [Roberta from configuration $\boldsymbol{q}=\mathbf{0}$, rotate itself in clockwise direction. . . . 60
5.4 Evaluated stability measure along the time, applied on the Roberta base wrench simulations, with simulation time 0.1 s. Tip-over edges are shown when the system is unstable. . . . 61
5.5 Evaluated stability measure, applied on the Roberta base wrench simulations, with different simulation time. Tip-over edges are omitted. . . . 61
5.6 Evaluated stability measure applied on the Roberta base wrench simulations, performed on Demo Red Box trajectory. . . . 62
5.7 Proposed offline tip-over evaluation. . . . 63

A.1 Base wrench force components, simulated on the *Dynamics excitation* trajectory with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods. . . . III
A.2 Base wrench moment components, simulated on the *Dynamics excitation* trajectory with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods. . . . IV
A.3 Base wrench force components, simulated on the *PTP and LIN* movement with the overall error compensation in (3.72), are compared with the measurements. The plot shows the entire trajectory. . . . V
A.4 Base wrench moment components, simulated on the *PTP and LIN* movement with the overall error compensation in (3.72), are compared with the measurements. The plot shows the entire trajectory. . . . VI

C.1 Acting forces, lengths, and base reference frame of the example. . . . IX

List of Tables

2.1 Modified DH Parameter P80.800 9

3.1 Additional base inertial parameters in $\boldsymbol{\kappa}^f$. 28
3.2 Symbolic error for every wrench component, computed through (3.41). . . 29
3.3 Base inertial parameter of Roberta P80.800, from [4]. 40
3.4 Wrench inertial parameter of Roberta P80.800 41

4.1 Symbolic error for every wrench component, computed w.r.t. (3.61)-(3.62). Here the g_is refer to a generic function. 45
4.2 Additional base inertial parameters in $\boldsymbol{\kappa}^{\mathrm{ext}}$. 48
4.3 RMSE values, for each wrench component, with and without the compensation. 49
4.4 RMSE values for each component of the base wrench, of the three trajectories. 52

5.1 Inertia values $I_{\mathrm{p}_i,x}$ with respect to x_{p_i}-axes. 57
5.2 Coord. of the vertices of the support polygon, w.r.t. S_{b} frame. Units in mm. 58

Symbols

$\bullet$	Actual value of the quantity $\bullet$
$\dot{\bullet}$	First order derivative of the quantity $\bullet$, w.r.t. the time
$\ddot{\bullet}$	Second order derivative of the quantity $\bullet$, w.r.t. the time
$\tilde{\bullet}$	Processed measurements of the quantity $\bullet$
$\hat{\bullet}$	Quantity with similar meaning but different definition as $\bullet$
$\widehat{\bullet}$	Skew-symmetric matrix, defined by the components of vector $\bullet$
$\bullet^{*}$	Variant of the quantity $\bullet$
$\bullet^{\star}$	Matrix $\bullet$ converted in the joint space

General Matrices and Vectors

x	Scalar
$\boldsymbol{x}$	Column vector
$\boldsymbol{x}^{\top}$	Row vector
$\boldsymbol{X}$	Matrix
$\boldsymbol{X}^{-1}$	Inverse of matrix $\boldsymbol{X}$
$\boldsymbol{X}^{\top}$	Transposed of matrix $\boldsymbol{X}$
${}^{i}\boldsymbol{X}_{j}$	Force-torque transformation matrix from frame i to j
$\boldsymbol{I}$	Identity matrix

Kinematic Modeling

S_i	Coordinate system i
${}^{j}\boldsymbol{p}$	Coordinate vector written w.r.t. S_j
${}^{j}\boldsymbol{p}^{h}$	Vector ${}^{j}\boldsymbol{p}$ written in homogeneous coordinates
${}^{i}\boldsymbol{p}_{j}$	Vector representing the coordinates of the origin of frame S_j expressed in frame S_i
${}^{i}\boldsymbol{A}_{j}$	Rotation matrix of frame S_j with respect to frame S_i
${}^{i}\boldsymbol{T}_{j}$	Homogeneous transformation matrix
$\boldsymbol{\xi}$	Mapping matrix for the platform forward kinematics
$\boldsymbol{\Lambda}$	Matrix of the non-holonomic constraint
$\boldsymbol{w}$	Vector of platform generalized coordinates
$\boldsymbol{q}$	Roberta joint space coordinates, vector of joint angles
$\boldsymbol{q}_{\text{plat}}$	Platform joint space vector
q_i	Roberta i-th joint angle
${}^{i}\mathbb{T}_{i-1}$	Screw transformation matrix
$\mathbb{a}_i$	Vector of actuator drive direction
${}^{i}\mathbb{v}_{i}$	Vector of spatial velocities
r	Platform wheel radius

b	Distance between the two platform active wheels
$x_{\mathrm{F}}, y_{\mathrm{F}}$	Coordinates of platform point F, in the World frame
φ	Platform orientation w.r.t. World frame
$\vartheta_{\mathrm{l}}, \vartheta_{\mathrm{r}}$	Angles of the left and right platform wheel
$v_{\mathrm{pl}}, v_{\mathrm{pr}}$	Linear velocities of the left and right platform active wheel
$v_{\mathrm{P}},\ \omega_{\mathrm{P}}$	Linear and angular velocity of the platform point P
$\hat{p}_{\mathrm{G}}$	Complex number that identifies platform point G in polar coordinates
$v_{\mathrm{u}}, v_{\mathrm{w}}$	Radial and tangential platform velocity in the polar modeling
$a_{\mathrm{u}}, a_{\mathrm{w}}$	Radial and tangential platform acceleration in the polar modeling

Dynamic Modeling

$\boldsymbol{\mathfrak{f}}$	Force vector
$\mathfrak{f}$	Force vector component
$\boldsymbol{\mathfrak{f}}_i$	Force exerted on link i by link $i-1$
$\boldsymbol{\mathfrak{f}}_{ei}$	Force exerted by link i on the environment
$\boldsymbol{\mu}$	Moment vector
μ	Moment vector component
$\boldsymbol{\mu}_i$	Moment about S_i exerted on link i by link $i-1$
$\boldsymbol{\mu}_{ei}$	Moment about S_i exerted by link i on the environment
$\boldsymbol{f}$	Generalized force-torque vector
$\boldsymbol{g}$	Vector of gravitational forces
$\boldsymbol{F}_i$	External forces on link i
$\boldsymbol{M}_i$	Moment of external forces $\boldsymbol{F}_i$ on link i, about S_i
$\boldsymbol{v}_i$	Linear velocity of S_i, shorter notation for ${}^{i}\boldsymbol{v}_i$
$\boldsymbol{\omega}_i$	Angular velocity of link i, shorter notation for ${}^{i}\boldsymbol{\omega}_i$
$\boldsymbol{J}_i$	Inertia tensor
${}^{i}\boldsymbol{J}_{\mathrm{G},i}$	Inertia tensor about the link center of mass
$J_{\mathrm{d},i}$	Moment of inertia of the i-th rotor
$\boldsymbol{ms}_i$	First order moment of inertia of i-th link
$\mathrm{F}_{\mathrm{c}i}$	Parameter of Coulomb friction acting at joint i
$\mathrm{F}_{\mathrm{v}i}$	Parameter of viscous friction acting at joint i
$\boldsymbol{M}$	Inertia matrix
$\boldsymbol{C}$	Coriolis/Centrifugal matrix
$\boldsymbol{\psi}$	External force vector
$\boldsymbol{V}$	Vector of velocity dependent forces
$\boldsymbol{E}$	Input transformation matrix
$\boldsymbol{\kappa}_{\mathrm{std},i}$	Dynamic parameters of link i
$\boldsymbol{\kappa}_{\mathrm{std}}$	Vector of all inertial parameters of the manipulator
$\boldsymbol{\kappa}$	Vector of base inertial parameters of the manipulator
$\boldsymbol{\kappa}^{\mathrm{w}}$	Wrench parameters not identified in previous work
$\boldsymbol{\kappa}^{f}$	Wrench parameter set s.t. $\boldsymbol{\kappa} \cup \boldsymbol{\kappa}^{\mathrm{w}}$
$\boldsymbol{\kappa}^{f*}$	Wrench parameter set, with parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$ equal to zero
$\boldsymbol{\kappa}^{\mathrm{J}}$	Vector composed by the inertia of the first two rotors
$\boldsymbol{\kappa}^{\mathrm{ext}}$	Vector containing all the parameters to identify
$\boldsymbol{W}_{\mathrm{std}}$	Matrix of inverse dynamics as a function of the standard inertial parameters
$\boldsymbol{W}$	Matrix of inverse dynamics as a function of the base inertial parameters

$\boldsymbol{W}^*$	Ideal matrix of the base wrench as a function of the base inertial parameters
$\boldsymbol{W}^f$	Matrix of the base wrench, as a function of the wrench parameters
$\boldsymbol{W}_\mathrm{J}$	Matrix representing the contribution of $\boldsymbol{\kappa}^\mathrm{J}$ in the approximation $\hat{\boldsymbol{f}}^{\mathrm{w}*}$
$\boldsymbol{W}_\mathrm{w}$	Matrix representing the contribution of $\boldsymbol{\kappa}^\mathrm{w}$ in the approximation $\hat{\boldsymbol{f}}^{\mathrm{w}*}$
$\boldsymbol{f}^\mathrm{w}$	Base wrench of the manipulator
$\boldsymbol{f}_\mathrm{Ex,Man}$	Force-torque that the manipulator exerts on the platform
$\hat{\boldsymbol{f}}^\mathrm{w}$	Computable approximation of the base wrench $\boldsymbol{f}^\mathrm{w}$, linear w.r.t. the $\boldsymbol{\kappa}^f$
$\boldsymbol{f}^\mathrm{F}$	Wrench exerted on platform point F
$\boldsymbol{f}_\mathrm{sen}$	Base wrench measurement from force-torque sensor
$\boldsymbol{f}_\mathrm{sim}$	Simulated Base wrench
$\hat{\boldsymbol{f}}^{\mathrm{w}*}$	Approximated base wrench formulation linear w.r.t. $\boldsymbol{\kappa}^{f*}$ parameter set
$\bar{\boldsymbol{f}}^\mathrm{w}$	adjusted formulation of $\hat{\boldsymbol{f}}^{\mathrm{w}*}$ with error $\boldsymbol{\varepsilon}$ compensation
$\boldsymbol{\varepsilon}$	Symbolic expression of the error in the base wrench formulation
$\tilde{\varepsilon}$	Computed error
L	Function of Lagrange
J_p	Platform moment of inertia
λ	Lagrange multiplier
F_ul, F_ur	Longitudinal force exerted on the vehicle, by the left and right wheel
F_wl, F_wr	Lateral force exerted on the vehicle, by the left and right wheel
$\boldsymbol{\tau}$	Vector of actuating joint torque
m_i	Mass of link i
m_p	Platform mass
m_c	Control cabinet mass

Parameter Identification

$\boldsymbol{y}$	System outputs vector
$\boldsymbol{g}$	Function of the system inputs and the parameter vector
$\boldsymbol{x}$	System excitation vector
$\boldsymbol{\kappa}$	Parameters of interest, in the identification procedure
$\boldsymbol{\varepsilon}$	Error vector
$\boldsymbol{n}$	Noise
$\boldsymbol{\mathcal{W}}$	Weighting matrix
$\boldsymbol{R}$	Variance matrix
$\boldsymbol{v}$	Random vector with unknown probability distribution
$\hat{\boldsymbol{\kappa}}_\mathrm{LS}$	Weighted least square estimator
$\hat{\psi}_\mathrm{LS}$	Least square function to minimize
M	Residual degree of freedom
N	Number of mesurements
r	Fitting parameters

Tip-over Evaluation

$\boldsymbol{e}_i$	i-th edge of the platform support polygon

$\hat{\boldsymbol{a}}_i$	i-th unit vector of the platform support polygon
$I_{\mathrm{p}i,x}$	i-th inertia value of the platform support polygon
$\boldsymbol{M}_{v_i}$	Moment of the external wrench, about the i-th vertex of the support polygon
$\boldsymbol{M}_i$	Projection of $\boldsymbol{M}_{v_i}$ about the i-th edge
α_i	Dynamic stability margin about the i-th edge
α	Dynamic MHS measure
$\hat{\alpha}$	Normalized dynamic MHS measure

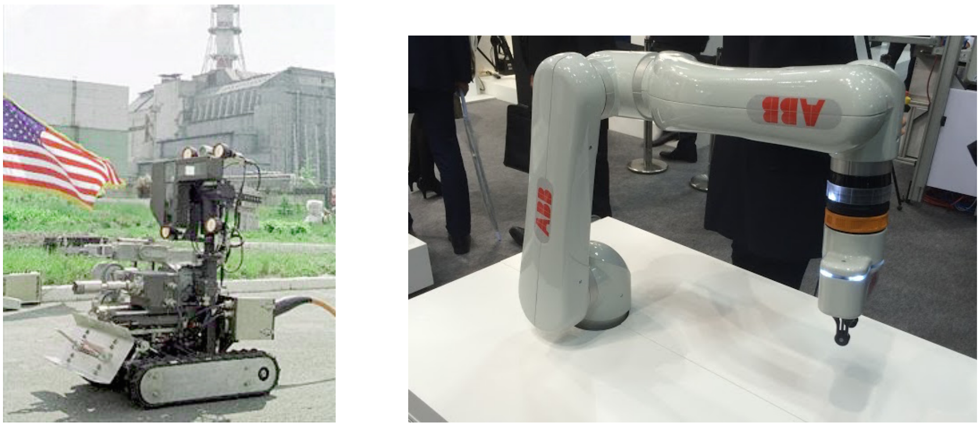
Abbreviations

RRRU	Regensburg robotics research unit
DoF	Degree of freedom
DH	Denavit-Hartenberg
WLS	Weighted least square
LS	Least square
SSE	Sum of squares error
MSE	Mean squared error
$RMSE$	Root mean squared error
dyn	Dynamics excitation trajectory
red	Demo red box trajectory
PL	PTP and LIN trajectory
des	Desired quantity
act	Measured quantity
plat	Platform
in	Initial
fin	Final
nom	Nominal value

1 Introduction

Mobile machines equipped with manipulator arms and controlled by on-board human operators, are commonplace systems for example in the construction, mining, and forestry industries. When these systems exert large forces, move heavy payloads, or operate over very uneven or sloped terrain, tip-over instabilities may occur which endanger the operator, risk damaging the machine, and reduce productivity. To reduce this risk, a supervisory control was introduced in mobile manipulator systems, so the safety and productivity of these mobile manipulators could be improved by automatic detection and prevention of tip-over instabilities.

In the same way, a similar issue comes out for a different class of robots. Mobile manipulators, which generically consist of an autonomous mobile platform and one or more manipulators, have attracted much interest along the years, because of their extended workspace, with respect to conventional, fixed base manipulators. Such systems are used in different kind of fields such as fire fighting, dismantling bombs, toxic waste cleanup, transportation of nuclear materials, and similar applications proposed by the nuclear, military and aerospace industries. An example of application of these systems is the Pioneer robot, depicted in Figure 1.1(a), donated to Ukraine from the US government, after the nuclear disaster in Chernobyl. The Pioneer is a remotely controlled robot which captured images of the interior of the sarcophagus, that engineers have used to create a 3D model of the reactor, in order to inspect the structural damage within a dangerous environment. When such systems carry heavy payloads, e.g. in intra-logistic tasks, move over irregular terrains or get sudden external disturbances, as we said, they would also benefit by a monitoring of the tip-over occurrence, as long as they may experience postural instabilities. Therefore, estimation and evaluation of the stability, with appropriate easy-computed measure throughout these systems motion, is needed, though it might represent a challenging task.



(a) Pioneer mobile robot.

(b) New ABB Roberta prototype.

Figure 1.1: Mobile robot and manipulator example.

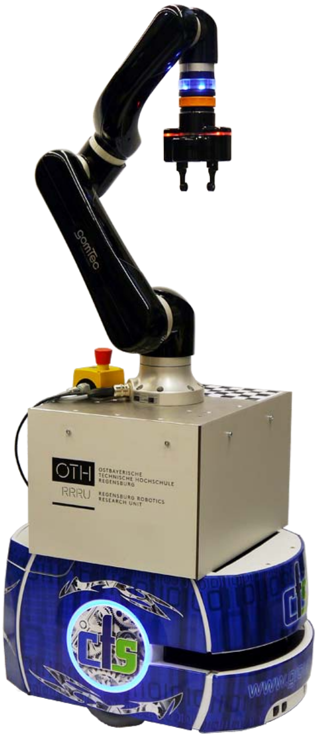


Figure 1.2: Mobile manipulator developed by RRRU.

1.1 Problem Description

The work that follows is under the supervision of the Regensburg Robotics Research Unit (RRRU), which is part of the Ostbayerische Technische Hochschule Regensburg and works in different fields of the robotics such as robot development and control, intelligent robot applications, mobile robotics, as well as Human-Machine-Interaction, teleoperation systems and image processing. RRRU works on the development of a mobile manipulator which is composed by the Lightweight robot system gomTec Roberta P80.800 included with its mounting base, and the autonomous mobile platform Adept Lynx. The overall system is depicted in Figure 1.2.

The Adept Lynx platform is a general-purpose mobile robot platform, self-guided and self-charging, designed to carry loads while working around people. The platform combines hardware and software, to provide an intelligent mobile platform to transport e specific payload structure. The Lightweight robot system gomTec Roberta P80.800, is a 6 degrees of freedom manipulator which is able to lift up to 8 kg. Two early releases of this manipulator, the Roberta P80.800 and P120.1200, became property of the RRRU in 2015, before the gomTec GmbH was bought by the ABB Group and the sale has been stopped. From ABB in the next future, the robot system Roberta will be likely redesigned with a slightly changed equipment, like the prototype shown in Figure 1.1(b).

This work is developed on the example of the first gomTec Roberta P80.800, combined with the Adept Lynx platform previously described, and it is in close relation with two previous works made by M. Stapf [4] and O. Kotz [5]. Their work concerned the derivation of a solution for the inverse kinematics, of the direct robot kinematics, and to find a dynamic model for the Roberta, with the purpose to design a self-developed control cabinet for it.

As we mentioned, mobile platforms can be adopted in a large variety of applications, and in very different environments, which means that several variables can affect on their assigned task. The most important regards the kind of movements performed by the platform, which could involve heavy linear accelerations, or curves that generate angular acceleration, dangerous for the sideways stability. Here, the stability state of the platform, can also be corrupted by external agents, specially in the case of outdoor environment that can involve atmospheric agents, as well as cases with irregular/sloping terrains. For this reason, we restrict the treatment in the most likely application, that is supposed to be in an indoor environment, with a flat and regular terrain, where we aim to start considering both platform and manipulator moving along a trajectory. Afterward during the work, we can eventually decide the case to put the majority of the effort.

1.2 State of the Art

Because of the huge variety of mobile robots, multi-legged robots and human-driven vehicles, that have been developed through the years, for multiple kind of applications, the tip-over problem became more and more articulated to study. Tip-over occurrence analysis is a topic widely investigated in order to avoid tip-over in automatic systems, as well as for providing human operators with an indication of proximity to tip-over. A consistent part of the literature considers environments with uphills or uneven surfaces, where the mobile manipulators are employed for rescue tasks in unstructured terrains [6], however, considering applications in flat and regular terrains, several stability analysis stragies have been designed.

A possible approach is illustrated by M. J. A. Safar in 2012 [7] where the tip-over analysis is performed on holonomic omnidirectional mobile robots, with active dual-wheel caster assemblies, by estimating the tipping direction. The authors proposed a separated study for the upper and the lower subsystem reaction forces, putting the focus on their interaction. On this purpose, the problem of determining the wheel reaction forces through the Newton and Euler equations of the dynamics, had a solution thanks to the combination of two platforms with three wheels each.

Zero Moment Point (ZMP) stability criterion, was firstly applied by Huang and Sugano in 1996 [8] to avoid overturning of a mobile manipulator, and was also used by Nikhade for omni-directional mobile robots [2]. The criterion is based on comparing the most stable point of the platform with the ZMP, which is defined as the point on the floor under the moving platform, where the resultant moment due to external forces is zero.

An alternative criterion called Force-Angle margin measure, also used by M. J. A. Safar [7], was presented by Papadopoulos and Rey in 2000 [9] and it is based on the minimum angle between resultant force exerted to the mobile base and the normal of tipover axes, and more recently, a measure called MHS criterion, based on stabilizing and destabilizing moments exerted to the mobile base, has been proposed by Moosavian and Alipour in [1]. All these mentioned criteria have been compared on a mobile manipulator by K. Alipour [10] through experiments, where the performance in terms of prediction of the exact time of instability occurrence, has been evaluated.

However, a further step ahead, as the authors in [11] and [12] did for mobile manipulators, is exploiting the fact that the stability of such systems, can be improved by the system itself, by designing a motion that can take the robot to a more stable configuration. In this way it is possible to perform a tip-over stability enhancement, maintaining the center of gravity inside the footprint. On this purpose, Wang [13] enhanced the performance of vehicles under critical driving situations, by optimally controlling the front and rear steering angles and the yaw moment.

In the case that we also want to design a control for the combined system, a coupled dynamic model that considers both the two subsystems dynamics is necessary. For this, the Lagrangian and the Newton-Euler formulation are the common practice, and their equivalence is proven in [14] with an application on a differential mobile platform, and in [15] with an application on a robotic arm manipulator.

Whether the dynamic parameters of the platform and manipulator are provided, we remark that the Lagrangian formulation can provide an easy and compact solution for a full coupled dynamics. As presented in [16], Papadopoulos and Poulakakis adopted this formulation to develop a dynamic model suitable for path planning tasks, explaining how to include the non-holonomic constraint in the equation of the dynamics.

At the same time, when the dynamics of the overall system is partially unknown, the Newton-Euler method can result more versatile. In fact, a manipulator mounted on a moving platform, generates reaction forces and moments at its base, which represent the effect that it has on the platform dynamics, that therefore affects the stability of the overall system. This approach split up the mobile manipulator, letting study the stability of a platform with external forces applied, as done by A. Moosavian in [17] and Grotjahn [18].

1.3 Goal of the Work

In order to clarify a goal for this work, we make a starting differentiation, by identifying two possible application scenarios. The first one, consists of a case where we do not have access to the motion of the system, so we cannot take counteractions such as change the trajectory. In this hypothetic scenario, a stability measure can come in help by stopping the system, or just alerting the user whether the system configuration is getting critical. A second application case, can be when we have access to the motion of the system. In this case, among multiple trajectories, the one that offers the better stability can be chosen, or in addition, counteractions like applying a specific trajectory that enhance the overall stability, can be performed.

For the goal of this work, we would like to start from the first simpler application case presented, where we aim to develop a methodology to forbid critical movements, which can be employed as external feature in the system. Therefore, as a first approach to the problem, we want to design a strategy that, from a given desired trajectory to perform, can provide an evaluation of the stability state of the system, along the time. In this way it is possible to predict when the system is moving toward an unstable configuration, letting the user either stop or rearrange the entire motion. Under this intuitive idea, a possible offline evaluation stability solution can be achieved.

1.4 Structure of this Thesis

Since the system in study in composed of two different kind of robots, this thesis face the kinematics and the dynamics for both of them separately. In Chapter 2 the kinematics of the Roberta P80.800 is introduced, recalling from previous work [4] all the elements that are needed for the current work, and giving mathematical tools necessary for the comprehension. Next the kinematics of the Adept Lynx platform is treated, offering a first Cartesian geometric modeling, putting the focus on the non-holonomic constraint, that plays an important role in the platform dynamic modeling. Then, an additional geometric description in polar coordinates is given, which is useful for the Newtonian dynamic modeling of the platform.

Next, the dynamics in Chapter 3 is presented first for the Roberta, where the main goal is to compute the manipulator base wrench, through the Newton-Euler approach. An initial formulation of the base wrench is found, linear w.r.t. the identified base parameter set, therefore, the error between this first formulation, and the correct Newton-Euler expression is presented and explained. Consequently, knowledge from the parameter estimation theory, is exploited to adjust the base wrench initial formulation, in order to get a more reliable expression. In this chapter, is also presented a possible solution for the control of the platform, that takes into account the dynamic effects of a possible mounted manipulator.

In Chapter 4 experiments are performed through a force-torque sensor, to provide an estimate of the unknown grouped base parameters and consequently to validate the adjusted base wrench formulation, by comparison with the collected force and torque measurements. Afterward additional data is collected by running different trajectories for the manipulator, with the purpose of cross validating the results obtained previously.

After having found a reliable base wrench formulation, Chapter 5 can finally introduce the proposed stability measure algorithm, whose results are shown for simple trajectories examples, and a final offline tip-over evaluation, is proposed as a first solution of the problem of this work.

Chapter 6 an overview of the results obtained is given, followed by suggestions for future work.

2 Kinematics

In the Robotics field, whether we want to control a robotic system, we wish to have the best possible knowledge about it, in order to let it perform a specific task. Thus, is crucial to find an accurate kinematic modeling, which is the study of the motion of mechanical systems without considering the forces that affect the motion. On this purpose the forward kinematics provides a relationship between the joint positions, that allow the robotic system to move, and the linear and angular velocity of the robotic system in Cartesian space. Many applications also require the knowledge of the inverse kinematic model, which provides the opposite relationship, that is velocities of the robotic system in function of the joint angles. In this chapter, the kinematics is treated for both platform and manipulator separately, which set the basis for the rest of the work.

2.1 Kinematics of a Serial Link Manipulator

In this section, the kinematics for the Roberta P80.800 is presented, which gives the relationship between the joint velocities and the corresponding end-effector linear and angular velocity. This mapping is described by a geometric Jacobian, a matrix which depends on the manipulator configuration, which constitutes one of the most important tools for manipulator characterization, useful for instance to describe the mapping between forces applied to the end-effector and resulting torques at the joints.

2.1.1 Transformation of Frames and Vectors

The description of the kinematics as well as the dynamics for a serial link manipulator, requires to express vectors that refer to a certain reference frame, with respect to another one. The reason under this comes from the fact that some quantities referring to different frames, often need to be expressed with respect to an unified coordinate system, specially when they are linked to each other through recursive formulas. On this purpose, here we recall some useful concepts from the theory of the transformation matrices reported in [19]. Let us consider an arbitrary point p in a Cartesian coordinate system S_j. This point can be located by the vector pointing from the origin of S_j to the point itself, that is ${}^j\boldsymbol{p} = [{}^jp^x, {}^jp^y, {}^jp^z]^\top$. This vector can be expressed by using the so called homogeneous coordinates

$$ {}^j\boldsymbol{p}^h = \begin{bmatrix} {}^jp^x, & {}^jp^y, & {}^jp^z, & 1 \end{bmatrix}^\top, \tag{2.1} $$

where ${}^jp^x$, ${}^jp^y$ and ${}^jp^z$ are the coordinates of the point along the corresponding axes of S_j, and the fourth entry, usually set to 1, denotes the scaling factor for these coordinates. We define ${}^i\boldsymbol{A}_j \in \mathbb{R}^{3\times3}$ as the rotation matrix of frame S_j with respect to frame S_i, which basically contains the description of the frame S_j in terms of components of its unit vectors

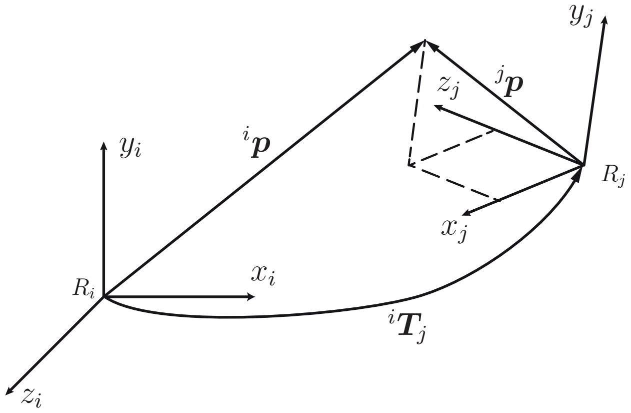


Figure 2.1: Transformation of a point. Figure from [4].

formulated w.r.t. frame S_i. Therefore vector ${}^j\boldsymbol{p}$ can be described in coordinates of another frame S_i through the following linear relationship

$$ {}^i\boldsymbol{p}^h = {}^i\boldsymbol{T}_j {}^j\boldsymbol{p}^h, \tag{2.2} $$

where we remark that the points ${}^i\boldsymbol{p}^h$ and ${}^j\boldsymbol{p}^h$ are written in homogeneous coordinates, see Fig. 2.1. Therefore we define

$$ {}^i\boldsymbol{T}_j = \begin{bmatrix} {}^i\boldsymbol{A}_j & {}^i\boldsymbol{p}_j \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.3} $$

as the *homogeneous transformation matrix*, which describes the relative position and orientation of frame S_j w.r.t. frame S_i, and ${}^i\boldsymbol{p}_j$ the vector that expresses the location of the origin of frame S_j w.r.t. frame S_i.

We also report that any rotation matrix $\boldsymbol{A}$ is orthogonal, and it yields the property $\boldsymbol{A}^{-1} = \boldsymbol{A}^\top$, so the inverse of ${}^i\boldsymbol{T}_j$ is given by

$$ {}^i\boldsymbol{T}_j^{-1} = {}^j\boldsymbol{T}_i = \begin{bmatrix} {}^i\boldsymbol{A}_j^\top & -{}^i\boldsymbol{A}_j^\top {}^i\boldsymbol{p}_j \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \tag{2.4} $$

Last useful property that we mention regard the homogeneous transformation matrices, is that they can be combined by multiplication. In fact let us assume to have multiple frames S_n, $n = 1, \ldots, k$ where each consecutive transformation between these frames is defined w.r.t. the previous frame. Thus, the overall transformation matrix is given by

$$ {}^0\boldsymbol{T}_k = {}^0\boldsymbol{T}_1 {}^1\boldsymbol{T}_2 \ldots {}^{k-1}\boldsymbol{T}_k. \tag{2.5} $$

2.1.2 Direct Kinematic Model

At this point, a geometric model of the manipulator can be derived whether we describe frames and vectors in terms of homogeneous coordinate transformation. The geometric model would describe the current position and orientation of the end-effector of a manipulator, as function of its joint angles as

$$ {}^0\boldsymbol{T}_\mathrm{e} = {}^0\boldsymbol{T}_\mathrm{e}(q_1, q_2, q_3, q_4, q_5, q_6) = {}^0\boldsymbol{T}_\mathrm{e}(\boldsymbol{q}), \tag{2.6} $$

where $\boldsymbol{q} \in \mathbb{R}^{6\times 1}$ contains the joint space coordinates, i.e. the angle of each joint. We observe that just by placing a reference frame in every joint of the manipulator, Eq. (2.6) can be easily determined by applying the multiplication in (2.5), as it has been done by M. Stapf [4].

Modified Denavit-Hartenberg Convention

The modeling of robots in a systematic and automatic way requires an adequate method for the description of their structure. Shortly denoted with DH, the standard Denavit-Hartenberg convention gives a systematic method to define the relative position and orientation of two consecutive links. It is a well known approach to describe such systems in the robotics field, which provides an intuitive way to determine the two frames attached to the two links and to compute the coordinate transformations between them, through homogeneous transformation matrix. Though Denavit-Hartenberg is the most popular method to accurately describe serial structures composed of a sequence of $n+1$ links and n joints, sometimes it can still present ambiguities. Hence, a modified DH version has been preferred to the conventional one, for the fact that it gives a unified description for all the mechanical articulated systems with a minimum number of parameters, and it comes with some useful properties of linearity which will be useful in the rest of the work. This modified convention has been defined in the literature and has been used in this work for the computation of the dynamic model of the Roberta P.80.800 with the Newton-Euler formulation and for the computation of the base reaction forces, at the manipulator base. The main difference to the original DH convention, is the placing of the coordinate systems, while frame S_i in DH convention is placed on the on the common joint axis of link i and link $i+1$, it is now placed along the axis of joint i which connects link $i-1$ with link i. This provides a simpler description of manipulator dynamics compared to the classic DH convention.

Assuming that the links are perfectly rigid and ideal, with no backlash and no elasticity, and the joints are either revolute or prismatic, the links are numbered such that link 0 constitutes the base of the manipulator and link n the terminal link. Then, the modified DH convention specifies how the frame S_i on each link i, has to be assigned. For further information, refer to the work in [4], where the author also derived the Table 2.1, that reveals the modified DH parameters for the Roberta P80.800 manipulator, which were used to create all the transformation matrices in the current work. The distances denoted by d_i and r_i can be found in Fig. 2.2.

Table 2.1: Modified DH Parameter P80.800

i	α_i(rad)	d_i(mm)	$\Delta\mathrm{q_i}$(rad)	r_i(mm)
1	0	0	0	-185
2	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0
3	0	380	0	0
4	$\frac{\pi}{2}$	0	π	420
5	$\frac{\pi}{2}$	0	π	0
6	$\frac{\pi}{2}$	0	0	0
7	0	0	0	$177+141$

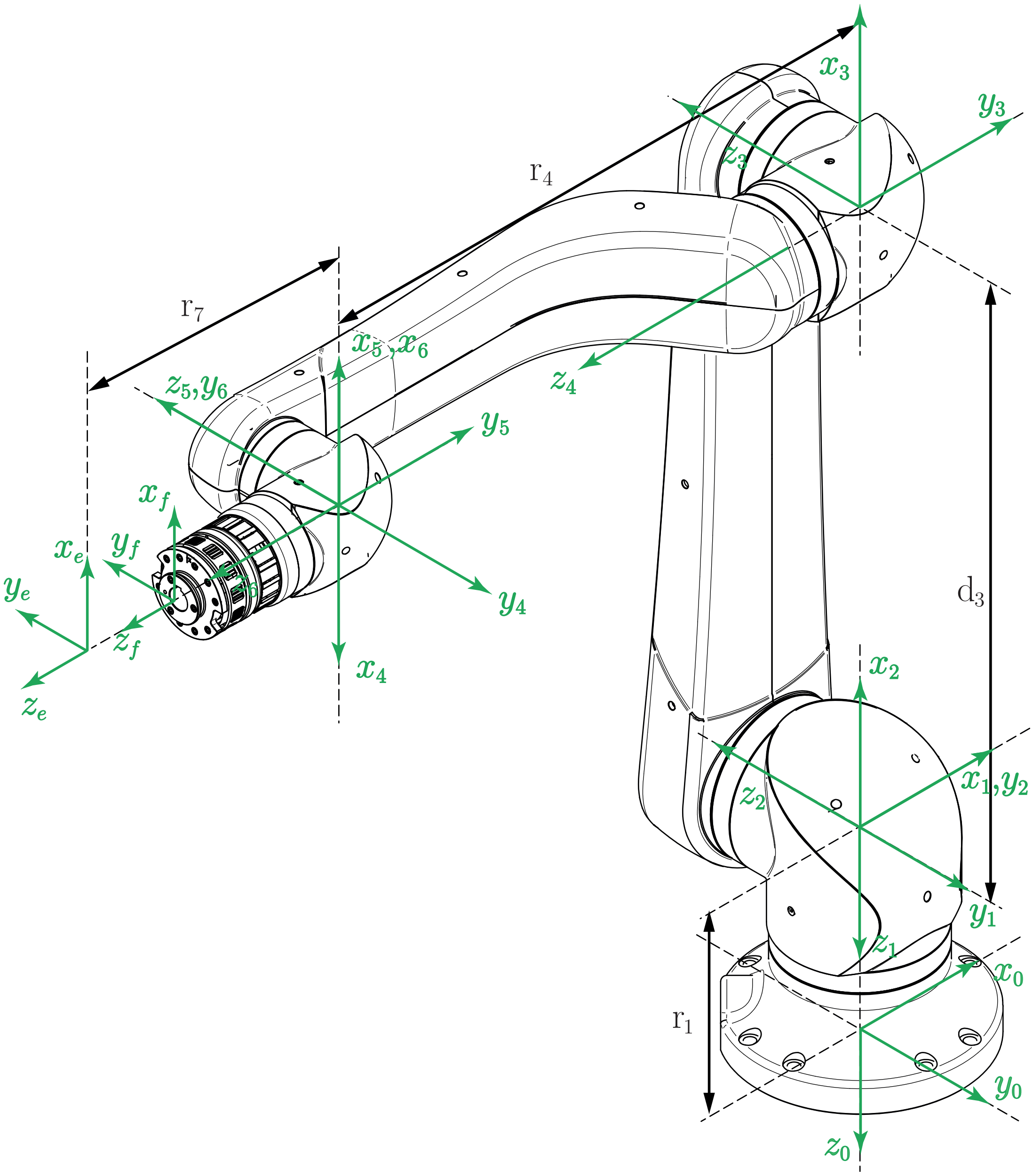


Figure 2.2: Frames according to modified Denavit-Hartenberg convention, $\boldsymbol{q} = \boldsymbol{0}$ [4].

We assume all the links numbered such that the base of the robot is considered frame 0 and the terminal link is numbered 6 in case of the Roberta P80.800 manipulator. The end-effector is given a separate coordinate system denoted S_e, see Figure 2.2.

Skew-Symmetric matrix

Here we provide an algebraic tool which will be useful for the rest of the work. Every cross product between two arbitrary vectors $\boldsymbol{m}$ and $\boldsymbol{n} \in \mathbb{R}^3$ defined as

$$\boldsymbol{m} \times \boldsymbol{n} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \times \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{2.7}$$

admits a linear representation through the so called *skew-symmetric* matrix, defined by the components of the vector $\boldsymbol{m}$ such that

$$\widehat{\boldsymbol{m}} = \begin{bmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{bmatrix} \ . \tag{2.8}$$

This algebraic operation is often used whenever there is the necessity to explicate in a linear form, an expression which involves cross products. The interested reader can verify that the following equation holds

$$\boldsymbol{m} \times \boldsymbol{n} = \widehat{\boldsymbol{m}} \cdot \boldsymbol{n} \ .$$

2.1.3 Recursive Computation of Link Velocities

Although the kinematic modeling of the Roberta has already been addressed by Matthias Stapf in [4], we recall the following computation of the link velocities of the Roberta, needed in the dynamics chapter.

These link velocities can be calculated with knowledge of the position and orientation of every frame S_i w.r.t the base frame. By exploiting the serial architecture of the Roberta P80.800, a recursive symbolic formulation from the base to the end-effector frame can be easily derived, following the formulas in [19].

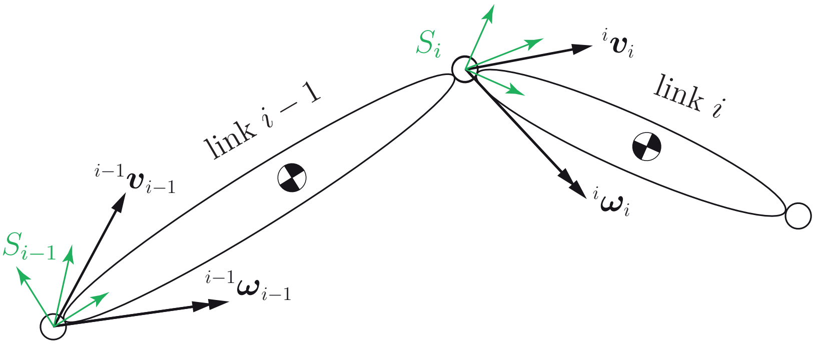

S_i
${}^{i}\boldsymbol{v}_i$
link i
link $i-1$
${}^{i-1}\boldsymbol{v}_{i-1}$
${}^{i}\boldsymbol{\omega}_i$
S_{i-1}
${}^{i-1}\boldsymbol{\omega}_{i-1}$

Figure 2.3: Augmented velocities of link $i-1$ and i [4].

Consider a frame S_{i-1} with known translational velocity ${}^{i-1}\boldsymbol{v}_{i-1}$ and rotational velocity ${}^{i-1}\boldsymbol{\omega}_{i-1}$, both expressed with respect to the frame S_{i-1}, see Fig. 2.3. Defining ${}^{i-1}\boldsymbol{p}_i$ as the vector connecting the origins of S_{i-1} and S_i w.r.t. S_{i-1}, the translational and rotational velocity of frame S_i w.r.t. frame S_{i-1} can be computed with the following

$${}^{i-1}\boldsymbol{v}_i = {}^{i-1}\boldsymbol{v}_{i-1} + {}^{i-1}\boldsymbol{\omega}_{i-1} \times {}^{i-1}\boldsymbol{p}_i \tag{2.9}$$

that can be rewritten exploiting the cross-product property $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$. Thus we obtain

$${}^{i-1}\boldsymbol{v}_i = {}^{i-1}\boldsymbol{v}_{i-1} - {}^{i-1}\boldsymbol{p}_i \times {}^{i-1}\boldsymbol{\omega}_{i-1} \tag{2.10}$$

$${}^{i-1}\boldsymbol{\omega}_i = {}^{i-1}\boldsymbol{\omega}_{i-1} . \tag{2.11}$$

With the help of the rotation matrix ${}^{i}\boldsymbol{A}_{i-1}$ we can express the previous quantities with respect to the frame S_i through

$$
\begin{aligned}
{}^{i}\boldsymbol{v}_i &= {}^{i}\boldsymbol{A}_{i-1}{}^{i-1}\boldsymbol{v}_i && (2.12)\\
{}^{i}\boldsymbol{\omega}_i &= {}^{i}\boldsymbol{A}_{i-1}{}^{i-1}\boldsymbol{\omega}_i. && (2.13)
\end{aligned}
$$

Consequently both (2.12) and (2.13) can be written in the following compact form

$$
{}^{i}\mathbb{v}_i = {}^{i}\mathbb{T}_{i-1}{}^{i-1}\mathbb{v}_{i-1} \tag{2.14}
$$

where

$$
{}^{i}\mathbb{v}_i = \begin{bmatrix} {}^{i}v_i^x & {}^{i}v_i^y & {}^{i}v_i^z & {}^{i}\omega_i^x & {}^{i}\omega_i^y & {}^{i}\omega_i^z \end{bmatrix}^\top \tag{2.15}
$$

is the vector of spatial velocities, and ${}^{i}\mathbb{T}_{i-1}$ is the so-called *screw transformation matrix* defined as

$$
{}^{i}\mathbb{T}_{i-1} = \begin{bmatrix} {}^{i}\boldsymbol{A}_{i-1} & -{}^{i}\boldsymbol{A}_{i-1}{}^{i-1}\widehat{\boldsymbol{p}}_i \\ \boldsymbol{0}_{3\times 3} & {}^{i}\boldsymbol{A}_{i-1} \end{bmatrix} . \tag{2.16}
$$

We remark that in Eq. (2.16), ${}^{i-1}\widehat{\boldsymbol{p}}_i$ is the skew-symmetric matrix, that allows the cross product in (2.10) to be rewritten as $-{}^{i-1}\widehat{\boldsymbol{p}}_i \cdot {}^{i-1}\boldsymbol{\omega}_{i-1}$, according with (2.8). Here we observe that if joint i is revolute, the overall angular velocity ${}^{i}\boldsymbol{\omega}_i$ is imposed by the rotation $\dot{q}_i$ of the drive unit. Hence the spatial velocity of frame S_i is obtained with

$$
{}^{i}\mathbb{v}_i = {}^{i}\mathbb{T}_{i-1}{}^{i-1}\mathbb{v}_{i-1} + {}^{i}\mathbb{a}_i\dot{q}_i \tag{2.17}
$$

where

$$
{}^{i}\mathbb{a}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top \tag{2.18}
$$

denotes the rotation of the driving unit along axis $\boldsymbol{z}_i$ according to the modified DH convention. From (2.17), the spatial link velocities can be computed recursively as a function of $\boldsymbol{q}$ with initial condition ${}^{0}\mathbb{v}_0 = 0$ in case of a fixed manipulator base.

2.1.4 Inverse Kinematic Model

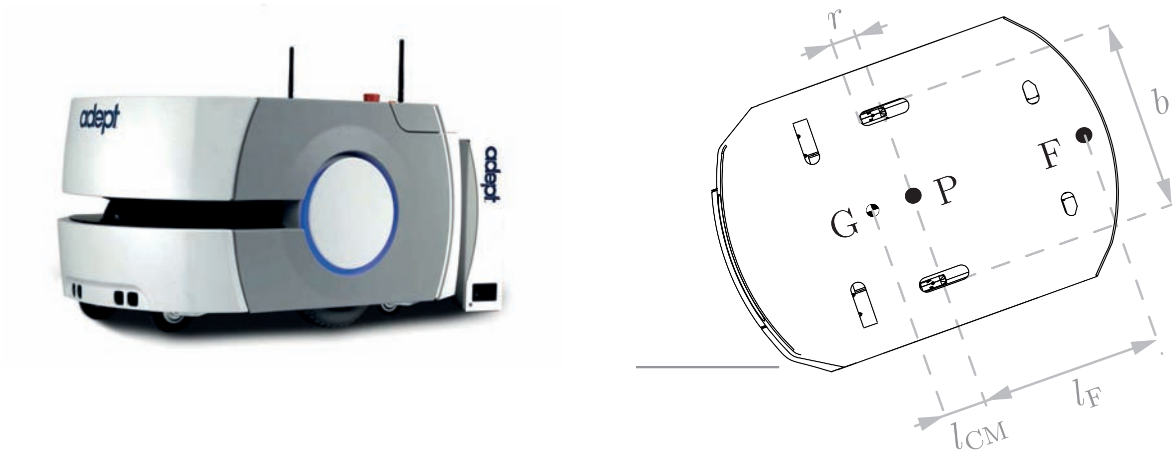
Despite the inverse kinematic problem is not a topic of this work, for completeness we mention that a solution for the inverse kinematic problem, which find joint angles that lead to a desired Cartesian pose

$$
\boldsymbol{q} = \mathbf{f}({}^{0}\boldsymbol{T}_6^{\text{des}}),
$$

has been found by Oliver Kotz in his work [5], where the solution is presented and explained in detail.

2.2 Kinematics of a Differential Mobile Platform

The kinematic modeling of a mobile platform can refer to model the relationship between the linear and angular velocity of the platform and its control variables, that are the wheels velocities. Mobile manipulators can be built on different kind of mobile platform designs, that differ by the drive mechanism employed. The most commonly available mobile platforms use either a car-like drive, which is a tricycle design where the driving front wheel is also used for steering, or a differential drive which employs two independently driven wheels with a common axis, and casters that add stability to the mobile platform. Further information about other driving configurations can be found in [19]. In this work, the employed mobile platform uses a two-wheel differential-drive, composed by two independent drive wheels and four passive wheels, known as *casters*, two in the front and two in the rear. Fig. 2.4 shows the described wheel configuration and three platform points of interest, named as P the point which lies in the common axis between the two wheels, equidistant from both of them, G the point which identifies the platform center of mass and F, which will identify the payload reference frame.



(a) Lynx Platform picture - Copyright Adept.　(b) Platform lengths and points of interest.

Figure 2.4: Adept Lynx Autonomous Intelligent Vehicle.

2.2.1 Adept Lynx Autonomous Mobile Platform

Before going into deep with the kinematics, the platform taken into account in this work is going to be introduced. The Adept Lynx is a general-purpose autonomous intelligent mobile platform, see Fig. 2.4(a), designed to carry payload structures up to 60 kg while working around people, in any wheelchair-accessible environment. Autonomous and intelligent are referred to the fact that it is self-guided and self-charging, with an automated docking station, and it has the ability to know its position within a workspace, and to navigate safely and autonomously to any accessible destination, continuously and without human intervention. Such platform is a good candidate for this work, and it has been chosen thanks to the fact that just by providing a final position and orientation, it will be able to control itself by taking care of the entire motion on the terrain. Its primary guidance uses a safety scanning laser to navigate, comparing the laser readings to a digital map stored on the platform. The laser is backed up by two front and two rear-facing sonar pairs, a front sensing bumper, and a gyroscope mounted on the internal Lynx core. The platform is a two-wheel differential-drive vehicle, with two passive casters in the front and in the rear, and two independent drive wheels. Its solid wheels are at the mid-line of the platform, so that the platform can turn in place.

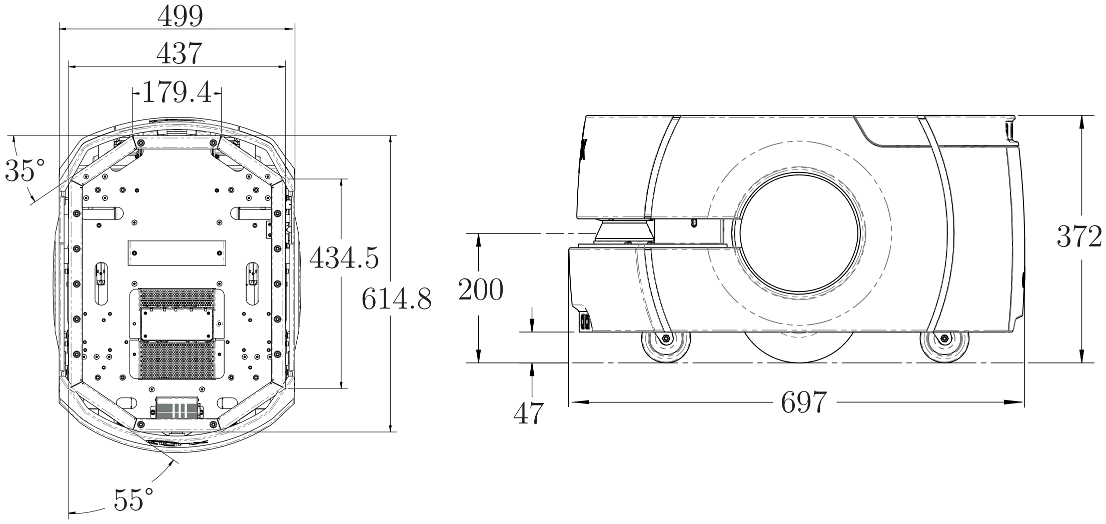


Figure 2.5: Platform dimensions in mm.

Payload integration

As it has been mentioned already, the Lynx platform can be customized with a payload structure attached to the top of it, represented by the gomTec Roberta P80.800 in this work. Mobility and navigation needed, as well as power and I/O connections between the platform and the payload structure are provided, in a way that both platform and payload can work together.

The main factors to consider in including a payload structure are the size, weight, center of gravity of the payload structure, and power requirements. An increased payload structure weight will decrease the vehicle run-time, and the stability of the combined system can get critical, specially in cases where the Lynx platform is equipped with a robotic arm, which can lift items off-center from the Lynx platform.

Platform Stability Considerations

On this purpose, the manufacturer provided further information, which is crucial at the point when we integrated a payload, and we need to take care about the steadiness of the platform. An heavy payload structure, with most of its weight concentrated just above the Lynx platform, will be much more stable than the same weight payload structure in which the weight is either off-center or high above the top of the platform. In order to avoid dangerous configurations, the payload structure must be kept no wider and no longer than the Lynx platform dimensions in Figure 2.5. It is also recommendable to keep it higher than the top of the Lynx platform, in order to avoid that any of the platform's sensors, e.g. scanning lasers, get blocked. Stability conditions are also highly influenced by the center of gravity of the payload, see Fig. 2.6, which should be kept centered on the Lynx platform, and as close as possible to the platform top to obtain the best stability; specially when crossing thresholds or irregularities in the floor.

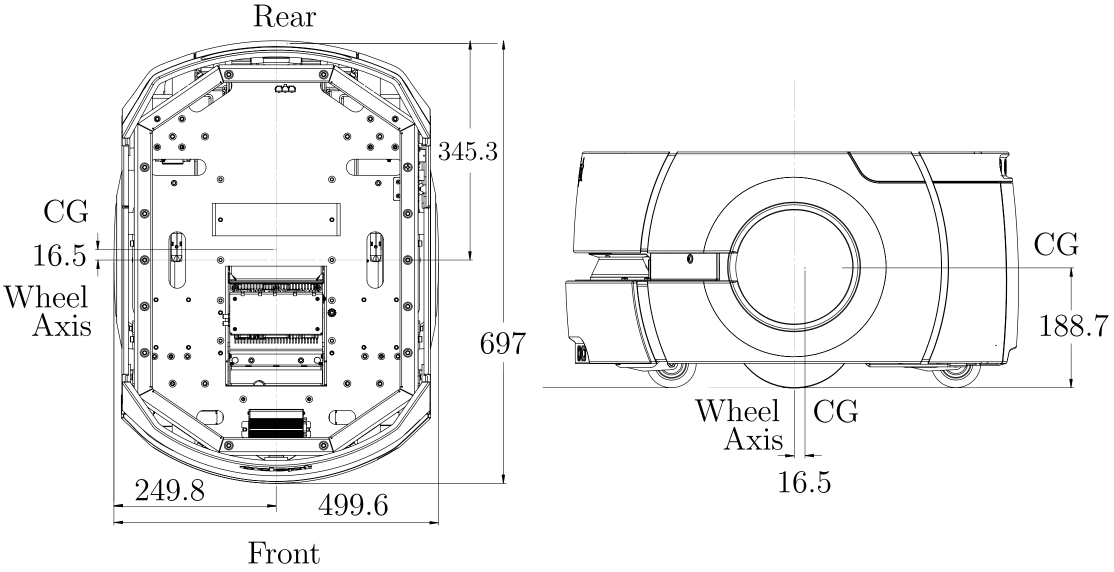


Figure 2.6: Center of Gravity of the platform without payload structure. Units in mm.

2.2.2 Cartesian Geometric Modeling

For controlling the pose of the platform in the free space, we need to describe its position and orientation in the two-dimensional environment. Referring to Fig. 2.7 we first define two different coordinate frames:

- World Reference Frame S_{w}, is commonly known in the literature as the *inertial* reference frame. This coordinate system is a global frame which is fixed in the environment in which the platform moves in.
- Platform Reference Frame S_{p}, the local base frame attached to the platform point P, and thus moving with it.

At this stage is important to define the mapping between these two frames. The position of any point on the moving robot defined in the Platform Frame, can also be described with respect to the World Frame through the following rotation matrix

$$ {}^{\mathrm{w}}\boldsymbol{A}_{\mathrm{p}} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.19} $$

More details regarding the rotation matrices theory can be found in [19]. Now, letting ${}^{\mathrm{w}}\boldsymbol{p}$ and ${}^{\mathrm{p}}\boldsymbol{p}$ be the coordinates of a given point in the World and in the Platform Frame respectively, the following relations hold

$$
{}^{\mathrm{w}}\boldsymbol{p} = {}^{\mathrm{w}}\boldsymbol{A}_{\mathrm{p}}\,{}^{\mathrm{p}}\boldsymbol{p} \tag{2.20}
$$

$$
{}^{\mathrm{p}}\boldsymbol{p} = \left({}^{\mathrm{w}}\boldsymbol{A}_{\mathrm{p}}\right)^{\top}\,{}^{\mathrm{w}}\boldsymbol{p} \ . \tag{2.21}
$$

Having placed two coordinate systems, we can now describe the absolute position and orientation of the platform, in the World Frame. Fig. 2.7 shows that the platform is moving in the free space by rotating its two independent wheels, so we identify $\dot{\vartheta}_{\mathrm{l}}$ and $\dot{\vartheta}_{\mathrm{r}}$ respectively angular velocities of the left and right wheel, as the variables that can be exploited to control the platform. Then, in order to properly describe the platform kinematics and the non-holonomic constraint, we define the following vector containing five generalized coordinates

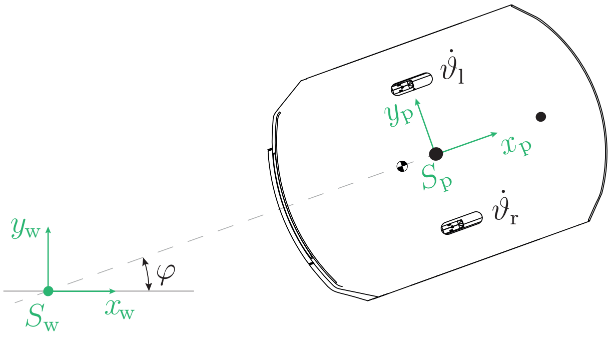


Figure 2.7: Global and local reference frames.

$$
\boldsymbol{w} = [x_{\mathrm{F}}, \quad y_{\mathrm{F}}, \quad \varphi, \quad \vartheta_{\mathrm{l}}, \quad \vartheta_{\mathrm{r}}] \ , \tag{2.22}
$$

where we added three coordinates that describe respectively the position of the platform in the World Frame, identified with point F and the orientation of the platform.

In this work, we choose point F to identify the platform position, even though point P is generally the easiest choice, for the fact that this will help us to keep track of the payload position, which is attached on the platform. We also highlight the fact that the caster wheels are passive, so they can be ignored during the kinematic modeling.

Forward Kinematics

The aim of the forward kinematics in the case of mobile platform, is to derive a relationship between its control variables, $\dot{\vartheta}_{\mathrm{l}}$ and $\dot{\vartheta}_{\mathrm{r}}$, and the platform linear and angular velocities in the World Frame. Now, under the assumption of low vehicle speeds, we get a *pure rolling motion* where each wheel maintains one contact point with the ground, without any sideways slipping. Denoting with v_{pr} and v_{pl} the linear velocities of the contact points of the left and right active wheel, in the Platform frame, the pure rolling motion implies

$$
v_{\mathrm{pr}} = r\dot{\vartheta}_{\mathrm{r}} \tag{2.23}
$$

$$
v_{\mathrm{pl}} = r\dot{\vartheta}_{\mathrm{l}} \ , \tag{2.24}
$$

which means that the angular velocity of the wheel has to be equivalent to the linear velocity of the corresponding contact point. In this way, see [19], from $\dot{\vartheta}_{\mathrm{l}}$ and $\dot{\vartheta}_{\mathrm{r}}$, we can obtain the linear and angular velocity for the platform point P, where we denote with r the wheels radius and with b the platform width, as stated in Fig. 2.4(b).

$$v_{\mathrm{P}} = \frac{r}{2}(\dot{\vartheta}_{\mathrm{r}} + \dot{\vartheta}_{\mathrm{l}}) \tag{2.25}$$

$$\omega_{\mathrm{P}} = \frac{r}{b}(\dot{\vartheta}_{\mathrm{r}} - \dot{\vartheta}_{\mathrm{l}}) \tag{2.26}$$

Now, through φ, angle between the two frames, we can describe these quantities with respect to the World frame

$$\dot{x}_{\mathrm{P}} = v_{\mathrm{P}} \cos\varphi = \frac{r}{2}\dot{\vartheta}_{\mathrm{r}} \cos\varphi + \frac{r}{2}\dot{\vartheta}_{\mathrm{l}} \cos\varphi \tag{2.27}$$

$$\dot{y}_{\mathrm{P}} = v_{\mathrm{P}} \sin\varphi = \frac{r}{2}\dot{\vartheta}_{\mathrm{r}} \sin\varphi + \frac{r}{2}\dot{\vartheta}_{\mathrm{l}} \sin\varphi \tag{2.28}$$

$$\dot{\varphi} = \omega_{\mathrm{P}} = \frac{r}{b}(\dot{\vartheta}_{\mathrm{r}} - \dot{\vartheta}_{\mathrm{l}}) \ . \tag{2.29}$$

As we mentioned, point F is the one of interest, so defining l_{F} as the distance between point P and F, we obtain

$$x_{\mathrm{F}} = x_{\mathrm{P}} + l_{\mathrm{F}} \cos\varphi \tag{2.30}$$

$$y_{\mathrm{F}} = y_{\mathrm{P}} + l_{\mathrm{F}} \sin\varphi \ . \tag{2.31}$$

Through time differentiation, we obtain the velocity of point F, with respect to the World Frame, as function of $\dot{x}_{\mathrm{P}}$ and $\dot{y}_{\mathrm{P}}$. By substituting $\dot{x}_{\mathrm{P}}$ and $\dot{y}_{\mathrm{P}}$ we obtain

$$\dot{x}_{\mathrm{F}} = \dot{x}_{\mathrm{P}} - l_{\mathrm{F}}\dot{\varphi}\sin\varphi = \frac{r}{b}(\frac{b}{2}\cos\varphi + l_{\mathrm{F}}\sin\varphi)\dot{\vartheta}_{\mathrm{l}} + \frac{r}{b}(\frac{b}{2}\cos\varphi - l_{\mathrm{F}}\sin\varphi)\dot{\vartheta}_{\mathrm{r}} \tag{2.32}$$

$$\dot{y}_{\mathrm{F}} = \dot{y}_{\mathrm{P}} + l_{\mathrm{F}}\dot{\varphi}\cos\varphi = \frac{r}{b}(\frac{b}{2}\sin\varphi - l_{\mathrm{F}}\cos\varphi)\dot{\vartheta}_{\mathrm{l}} + \frac{r}{b}(\frac{b}{2}\sin\varphi + l_{\mathrm{F}}\cos\varphi)\dot{\vartheta}_{\mathrm{r}} \tag{2.33}$$

which leads to the following platform forward kinematic model

$$\begin{bmatrix} \dot{x}_{\mathrm{F}} \\ \dot{y}_{\mathrm{F}} \\ \dot{\varphi} \\ \dot{\vartheta}_{\mathrm{l}} \\ \dot{\vartheta}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \frac{r}{b}(\frac{b}{2}\cos\varphi + l_{\mathrm{F}}\sin\varphi) & \frac{r}{b}(\frac{b}{2}\cos\varphi - l_{\mathrm{F}}\sin\varphi) \\ \frac{r}{b}(\frac{b}{2}\sin\varphi - l_{\mathrm{F}}\cos\varphi) & \frac{r}{b}(\frac{b}{2}\sin\varphi + l_{\mathrm{F}}\cos\varphi) \\ -\frac{r}{b} & \frac{r}{b} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\vartheta}_{\mathrm{l}} \\ \dot{\vartheta}_{\mathrm{r}} \end{bmatrix} \ . \tag{2.34}$$

where we define $\boldsymbol{\xi}(\boldsymbol{w})$ as the mapping matrix for the forward kinematics in Eq. (2.34).

On this purpose, we remark that in most of the applications, specially within the path planning topic, the autonomous platform is asked either to follow a desired path or to reach a specific destination in the free space, identified through a specific reference system; e.g. the Adept Lynx platform employed in this work, which asks for a Cartesian pose in a two dimensional map. Hence, fulfill these tasks can result awkward and problematic whether we control the velocities of the wheels, since they don't keep any information about the absolute pose of the platform on the ground. Therefore this situation demand for a crucial geometric relationship between the coupled motion of the two driving wheels, and the Cartesian pose of the platform, identified with position and orientation in the space.

Another alternative form can be obtained by representing the platform velocities in terms of the linear and angular velocities in the Platform Frame. From Eq. (2.27)-(2.29) and (2.32)-(2.33) we obtain

$$\begin{bmatrix} {}^{\mathrm{P}}\dot{x}_{\mathrm{F}} \\ {}^{\mathrm{P}}\dot{y}_{\mathrm{F}} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -l_{\mathrm{F}}\sin\varphi \\ \sin\varphi & l_{\mathrm{F}}\cos\varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{\mathrm{P}} \\ \omega_{\mathrm{P}} \end{bmatrix} . \tag{2.35}$$

2.2.3 Non-Holonomic Constraint

An important role in this work is covered by the non-holonomic constraint which is involved in the motion of the Adept Lynx platform, and generally arises from rolling contacts between two or more rigid bodies. Examples of mechanical systems that involve rolling contacts include wheeled vehicles such as conventional automobiles, unconventional actively coordinated robotic systems such as planetary rovers, manipulators grasping an object and legged locomotion systems. Wheeled mobile robots are typical examples of mechanical systems with non-holonomic constraints, where the rolling contact is maintained passively through external forces such as gravitational forces. As we will see in the Dynamics chapter, such systems have often highly nonlinear dynamics, and are more complex to study with respect to the holonomic ones, for the fact that they present constraints which cannot be violated.

Now, considering a differential-drive mobile robot, its motion is characterized by two non-holonomic constraint equations, which are obtained by the assumptions of pure rolling motion and no sideways slipping (see [20] and [21]). The *pure rolling motion*, represents the fact that each wheel maintains a one contact point with the ground and there is no slipping of the wheel in its longitudinal axis. This assumption has already been introduced, and it implies the constraints in Eq. (2.23) and (2.24).

At this point we denote with $[{}^{\mathrm{w}}\dot{x}_{\mathrm{pr}}, {}^{\mathrm{w}}\dot{y}_{\mathrm{pr}}, {}^{\mathrm{w}}\dot{z}_{\mathrm{pr}}]^{\top}$ the coordinate vector, in the World frame, of v_{pr} in Eq. (2.23), that we express w.r.t. the Platform frame by applying the rotation matrix ${}^{\mathrm{w}}\boldsymbol{A}_{\mathrm{p}}^{-1}$. Thus, doing the same for v_{pl}, and considering just the longitudinal x_{p}-component of both the transformed vectors, the rolling constraint equations in (2.23) and (2.24) can be formulated as follows

$$ {}^{\mathrm{w}}\dot{x}_{\mathrm{pr}}\cos\varphi + {}^{\mathrm{w}}\dot{y}_{\mathrm{pr}}\sin\varphi = r\dot{\vartheta}_{\mathrm{r}} \tag{2.36}$$

$$ {}^{\mathrm{w}}\dot{x}_{\mathrm{pl}}\cos\varphi + {}^{\mathrm{w}}\dot{y}_{\mathrm{pl}}\sin\varphi = r\dot{\vartheta}_{\mathrm{l}} \ . \tag{2.37}$$

Observe that for the right wheel contact point, it holds

$$ {}^{\mathrm{w}}\dot{x}_{\mathrm{pr}} = \dot{x}_{\mathrm{P}} + b/2\cdot\dot{\varphi}\cos\varphi \tag{2.38}$$

$$ {}^{\mathrm{w}}\dot{y}_{\mathrm{pr}} = \dot{y}_{\mathrm{P}} + b/2\cdot\dot{\varphi}\sin\varphi \ , \tag{2.39}$$

where $\dot{x}_{\mathrm{P}}$ and $\dot{y}_{\mathrm{P}}$ are defined in (2.27) and (2.28). Similarly for the left wheel contact point, we have

$$ {}^{\mathrm{w}}\dot{x}_{\mathrm{pl}} = \dot{x}_{\mathrm{P}} - b/2\cdot\dot{\varphi}\cos\varphi \tag{2.40}$$

$$ {}^{\mathrm{w}}\dot{y}_{\mathrm{pl}} = \dot{y}_{\mathrm{P}} - b/2\cdot\dot{\varphi}\sin\varphi \ . \tag{2.41}$$

Now using (2.32)-(2.33) and (2.38)-(2.41), we can express the velocities of the contact points in Eq. (2.36) and (2.37) as function of components of $\boldsymbol{w}$, where we algebraically obtain

$$\dot{x}_{\mathrm{F}} \cos\varphi + \dot{y}_{\mathrm{F}} \sin\varphi + b/2 \cdot \dot{\varphi} = r\dot{\vartheta}_{\mathrm{r}} \tag{2.42}$$

$$\dot{x}_{\mathrm{F}} \cos\varphi + \dot{y}_{\mathrm{F}} \sin\varphi - b/2 \cdot \dot{\varphi} = r\dot{\vartheta}_{\mathrm{l}} \ . \tag{2.43}$$

At the same time, *no sideways slipping* constraint simply means that the robot point P cannot move sideward, which implies, in the Platform frame

$$^{\mathrm{P}}\dot{y}_{\mathrm{P}} = 0 \ . \tag{2.44}$$

Recalling that $[\dot{x}_{\mathrm{P}}, \dot{y}_{\mathrm{P}}, \dot{z}_{\mathrm{P}}]^{\top}$ is the coordinate vector of P in the World frame, we express this vector w.r.t. the Platform frame, by applying the rotation matrix $^{\mathrm{w}}\boldsymbol{A}_{\mathrm{p}}^{-1}$. Hence, taking the second component, we have expressed Eq. (2.44) with

$$-\dot{x}_{\mathrm{P}} \sin\varphi + \dot{y}_{\mathrm{P}} \cos\varphi = 0 \ . \tag{2.45}$$

At the same time, substituting $\dot{x}_{\mathrm{P}}$ and $\dot{y}_{\mathrm{P}}$ through (2.32) and (2.33), we obtain a constraint for point F, that is

$$\dot{x}_{\mathrm{F}} \sin\varphi - \dot{y}_{\mathrm{F}} \cos\varphi + \dot{\varphi} l_{\mathrm{F}} = 0 \ . \tag{2.46}$$

Roughly speaking, the last constraint says that the lateral velocity of the platform point F is only due to the rotation of the platform. Eq. (2.46) is a non-holonomic constraint, and it can be proven that it cannot be integrated analytically.

Having $\dot{\boldsymbol{w}} = \begin{bmatrix} \dot{x}_{\mathrm{F}}, & \dot{y}_{\mathrm{F}}, & \dot{\varphi}, & \dot{\vartheta}_{\mathrm{l}}, & \dot{\vartheta}_{r} \end{bmatrix}$, finally the three constraints can be written in the form $\boldsymbol{\Lambda}(\boldsymbol{w})\dot{\boldsymbol{w}} = \boldsymbol{0}$ where

$$\boldsymbol{\Lambda}(\boldsymbol{w}) = \begin{bmatrix} \sin\varphi & -\cos\varphi & l_{\mathrm{F}} & 0 & 0 \\ \cos\varphi & \sin\varphi & b/2 & 0 & -r \\ \cos\varphi & \sin\varphi & -b/2 & -r & 0 \end{bmatrix} . \tag{2.47}$$

It is straightforward to verify that this matrix satisfies $\boldsymbol{\Lambda}(\boldsymbol{w})\boldsymbol{\xi}(\boldsymbol{w}) = \boldsymbol{0}$, that is equivalent to

$$\boldsymbol{\xi}^{\top}(\boldsymbol{w})\boldsymbol{\Lambda}^{\top}(\boldsymbol{w}) = \boldsymbol{0} \ , \tag{2.48}$$

where $\boldsymbol{\xi}(\boldsymbol{w})$ has been defined in Eq. (2.34). This property plays a crucial role in the dynamic modeling, and it will be viewed more in detail in the next chapters.

2.2.4 Polar Geometric Modeling

At the same time, here we want to mention an alternative way to model the kinematics, which set the basis for the Newton-Euler dynamics approach, for the platform, that will be explained in the dynamics chapter. Therefore, we initially define the quantities shown in Fig. 2.8, where $(v_{\mathrm{u}}, v_{\mathrm{w}})$ are the velocities of the platform center of gravity G, respectively the longitudinal velocity, and the lateral velocity. Similarly we define $(a_{\mathrm{u}}, a_{\mathrm{w}})$ as the

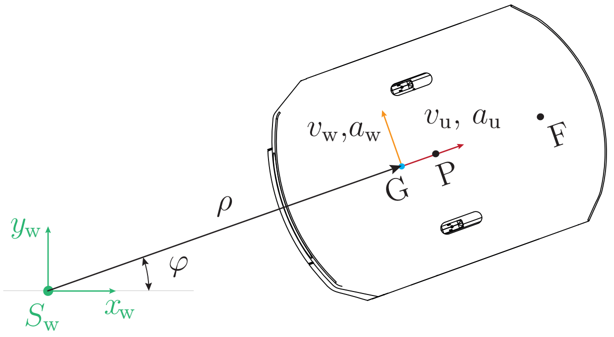


Figure 2.8: Platform with parameters for polar modeling.

longitudinal and lateral accelerations of G. We remark that v_{p} in Eq. (2.25) has the same meaning as v_{u} here, where we change the notation for highlight its difference with v_{w}.

Following the approach used by the authors in [14], we describe the platform center of mass G in polar coordinates, w.r.t the World reference frame. This is represented by the complex vector

$$\hat{p}_{\mathrm{G}} = \rho e^{j\varphi} \quad . \tag{2.49}$$

Then differentiating the center of gravity position vector with respect to the time, we obtain velocity and acceleration of G in the World frame, that are

$$\dot{\hat{p}}_{\mathrm{G}} = [\dot{\rho}]\, e^{j\varphi} + [\rho\dot{\varphi}]\, e^{j(\varphi+\frac{\pi}{2})} \tag{2.50}$$
$$\ddot{\hat{p}}_{\mathrm{G}} = \left[\ddot{\rho} - \rho\dot{\varphi}^2\right] e^{j\varphi} + [2\dot{\rho}\dot{\varphi} + \rho\ddot{\varphi}]\, e^{j(\varphi+\frac{\pi}{2})} \tag{2.51}$$

where we have also obtained an expression for the radial and tangential velocities $(v_{\mathrm{u}}, v_{\mathrm{w}})$ and accelerations $(a_{\mathrm{u}}, a_{\mathrm{w}})$, that result

$$v_{\mathrm{u}} = \dot{\rho} \tag{2.52}$$
$$v_{\mathrm{w}} = \rho\dot{\varphi} \tag{2.53}$$
$$a_{\mathrm{u}} = \ddot{\rho} - \rho\dot{\varphi}^2 \tag{2.54}$$
$$a_{\mathrm{w}} = 2\dot{\rho}\dot{\varphi} + \rho\ddot{\varphi} \quad . \tag{2.55}$$

Therefore, writing these expressions in platform radial and tangential velocity terms, we obtain an expression for the longitudinal and lateral platform accelerations

$$a_{\mathrm{u}} = \dot{v}_{\mathrm{u}} - v_{\mathrm{w}}\dot{\varphi} \tag{2.56}$$
$$a_{\mathrm{w}} = \dot{v}_{\mathrm{w}} - v_{\mathrm{u}}\dot{\varphi} \quad . \tag{2.57}$$

In addition, assuming a pure rolling motion in the longitudinal direction, and no sliding in the lateral direction, we can derive the non-holonomic constraint, by imposing the velocity of point P w.r.t. the local platform frame, equal to zero. Similarly in what has been done for obtaining (2.46), we obtain

$$v_{\mathrm{w}} = -l_{\mathrm{CM}}\dot{\varphi} \quad , \tag{2.58}$$

in a way that the lateral velocity of G, is only due to the rotation of the platform, where the length l_{CM} is the distance between the platform point P and its center of mass, referring to Fig. 2.4(b). Finally, through the polar coordinates, we have obtained an expression for the longitudinal and lateral platform accelerations as well as for the non-holonomic constraint, that will be helpful in the dynamic modeling of the platform.

3 Dynamics

As well as the kinematics, the study of the dynamics is important and has to be accurate in order to be able, for instance, to design a control strategy for the robotic system in study, which is one of the main purposes in the robotics. So generally, the dynamic model yields an accurate representation of the behavior of the system, when it interacts with its external environment. This is useful, for example, to provide motion simulations that allow control algorithms to be tested without using a physically available system, as well as they provide useful information for designing joints, transmissions and actuators.

The literature usually provides two main approaches to deal with the dynamics, the Lagrangian and the Newton-Euler formulation, that will be discussed in this chapter. The first method is conceptually simple and systematic, based on the Lagrange formulation, which concerns the kinetic and potential energies, and the second method, based on the Newton and Euler dynamic equations, yields the model in a recursive and generally in a computationally more efficient form.
In this chapter, the dynamics for the Roberta P80.800 is derived in order to compute the reaction forces of the manipulator at its base, and techniques of parameter identification are introduced to improve its accuracy.

3.1 Transformation of Force and Torque

Since forces and torques in the work that follows, need to be adjusted and transformed w.r.t. a specific reference frame, e.g. the readings from a sensor, it is worth recalling their behavior when the transformation matrices have to deal with them.

We recall that a free vector, e.g. the velocity vector, refers to a vector that may be positioned anywhere in space without loss or change of meaning, provided that magnitude and direction are preserved. Observe that if we have a moment vector ${}^{j}\boldsymbol{\mu}$ that is known in terms of S_j, then we calculate the same moment in terms of frame S_i with ${}^{i}\boldsymbol{\mu} = {}^{i}\boldsymbol{A}_j{}^{j}\boldsymbol{\mu}$.

Unlike the free vectors, the force vector is sensitive to the origins of the reference frames and every time it gets translated, it also generates a moment. In order to derive a compact form to transform force and moment, we define the *generalized force-torque vector* which contains linear components of force $\mathfrak{f} = [\mathfrak{f}_x, \mathfrak{f}_y, \mathfrak{f}_z]^\top$ and moment $\boldsymbol{\mu} = [\mu_x, \mu_y, \mu_z]^\top$ as

$$\boldsymbol{f} = \begin{bmatrix} \mathfrak{f} \\ \boldsymbol{\mu} \end{bmatrix} \in \mathbb{R}^{6\times 1}. \tag{3.1}$$

Thus, a force-torque vector can be transformed from frame S_j to frame S_i through the following linear relationship

$${}^{i}\boldsymbol{f} = {}^{i}\boldsymbol{X}_j{}^{j}\boldsymbol{f}. \tag{3.2}$$

So we denote ${}^{i}\boldsymbol{X}_j$ as the *force-torque transformation* defined as

$$ {}^{i}\boldsymbol{X}_j = \begin{bmatrix} {}^{i}\boldsymbol{A}_j & \boldsymbol{0}_{3\times 3} \\ {}^{i}\boldsymbol{S}_j \cdot {}^{i}\boldsymbol{A}_j & {}^{i}\boldsymbol{A}_j \end{bmatrix} , \tag{3.3} $$

where ${}^{i}\boldsymbol{A}_j$ represents the rotation matrix of frame S_j w.r.t. frame S_i and ${}^{i}\boldsymbol{S}_j$ is the skew-symmetric matrix

$$ {}^{i}\boldsymbol{S}_j = \begin{bmatrix} 0 & -{}^{i}p_j^z & {}^{i}p_j^y \\ {}^{i}p_j^z & 0 & -{}^{i}p_j^x \\ -{}^{i}p_j^y & {}^{i}p_j^x & 0 \end{bmatrix} $$

from (2.8), defined by the components of the translation vector ${}^{i}\boldsymbol{p}_j = \left[{}^{i}p_j^x, {}^{i}p_j^y, {}^{i}p_j^z\right]^\top$ describing the origin of frame S_j with respect to frame S_i.

3.2 Dynamic Parameters of a Rigid Body

Here we define some quantities needed to compute the dynamic model, that are identified as standard dynamic link parameters.

- In Fig. 3.1 are shown the translational and rotational velocity, respectively ${}^{i}\boldsymbol{v}_i$ and ${}^{i}\boldsymbol{\omega}_i$ already computed in (2.12) and (2.13), with respect to the frame S_i.

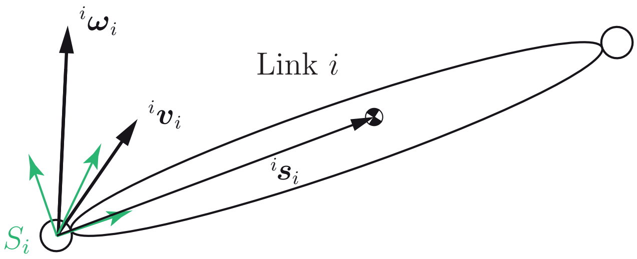


Figure 3.1: Link i moving in space [4].

- ${}^{i}\boldsymbol{J}_i \in \mathbb{R}^{3\times 3}$ is the symmetric inertial tensor w.r.t. frame S_i, known as second moment of inertia, defined as

$$ {}^{i}\boldsymbol{J}_i = \begin{bmatrix} J_{xx,i} & J_{xy,i} & J_{xz,i} \\ J_{xy,i} & J_{yy,i} & J_{yz,i} \\ J_{xz,i} & J_{yz,i} & J_{zz,i} \end{bmatrix} . \tag{3.4} $$

  Since we want to describe all the quantities w.r.t. the origin of the link frame S_i, we remark that it is computed according to the Steiner's Theorem such that

$$ {}^{i}\boldsymbol{J}_i = {}^{i}\boldsymbol{J}_{\mathrm{G},i} - m_i {}^{i}\widehat{\boldsymbol{s}}_i {}^{i}\widehat{\boldsymbol{s}}_i \ , \tag{3.5} $$

  where ${}^{i}\boldsymbol{J}_{\mathrm{G},i} \in \mathbb{R}^{3\times 3}$ denotes the inertia tensor about the center of mass, m_i is the link mass, ${}^{i}\boldsymbol{s}_i$ denotes the vector from the link frame origin to the link center of mass, see Fig. 3.1 and ${}^{i}\widehat{\boldsymbol{s}}_i$ is the corresponding skew-symmetric matrix equivalent to the cross-product, introduced in (2.8).

- The product of link mass and vector pointing to the center of gravity also known as first moment of inertia, is defined as

  $$m_i {}^i\boldsymbol{s}_i = \begin{bmatrix} m_i {}^i s_i^x & m_i {}^i s_i^y & m_i {}^i s_i^z \end{bmatrix}^\top \quad , \tag{3.6}$$

  and it is shortly denoted with $\boldsymbol{ms}_i = [ms_{x,i} \quad ms_{y,i} \quad ms_{z,i}]^\top$.

- Finally the link mass is denoted with m_i, and the rotor inertia with $J_{\mathrm{d},i}$.

Thus, for each link i, we can group all these quantities in a vector $\boldsymbol{\kappa}_{\mathrm{std},i} \in \mathbb{R}^{11\times 1}$ that contains all the dynamic link parameters, also known as inertial parameters

$$\boldsymbol{\kappa}_{\mathrm{std},i} = [J_{xx,i} \quad J_{xy,i} \quad J_{xz,i} \quad J_{yy,i} \quad J_{yx,i} \quad J_{zz,i} \quad ms_{x,i} \quad ms_{y,i} \quad ms_{z,i} \quad m_i \quad J_{\mathrm{d},i}]^\top \; . \tag{3.7}$$

3.3 Dynamics for the Roberta P80.800

In this work, the goal of the dynamics is to describe the forces and moments that act at the base of the manipulator, which set the basis for evaluate the stability of the platform and the mounted Roberta. In this section, the Newton-Euler formulation is exploited to derive the equations of motion for the Roberta P.80.800, that carries an expression for the reaction forces and moments at the manipulator base.

3.3.1 Newton-Euler Formulation

The Newton-Euler formulation yields the dynamic model in a recursive form, which is computationally more efficient since it exploits the typically open structure of the manipulator kinematic chain. Several implementations of this algorithm have been performed, e.g. in [22] where all the formulation was based on the computation of forces and moments acting on the center of mass of each link. This leads to a dynamic model that consists in a non-linear function in the inertial parameters, which is a tough drawback for example in the dynamic parameter identification. Therefore, the Newton-Euler algorithm rearranged by Khalil in [23] is taken into account hereafter, where forces and moments are computed with respect to the origin of frame S_i attached to joint i, instead of the center of mass, so that the final model results linear in the identifiable parameters.

From basic theory of mechanics, referring to the 6-DoF Roberta P80.800 in study, described with the modified DH convention (Fig. 2.2), we can compute the external forces and corresponding generated moments on link i about the origin of frame S_i, generally known as wrench, through the Newton-Euler equations

$$\boldsymbol{F}_i = m_i \dot{\boldsymbol{v}}_i + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{ms}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{ms}_i) \tag{3.8}$$

$$\boldsymbol{M}_i = \boldsymbol{J}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{J}_i \boldsymbol{\omega}_i) + \boldsymbol{ms}_i \times \dot{\boldsymbol{v}}_i \quad . \tag{3.9}$$

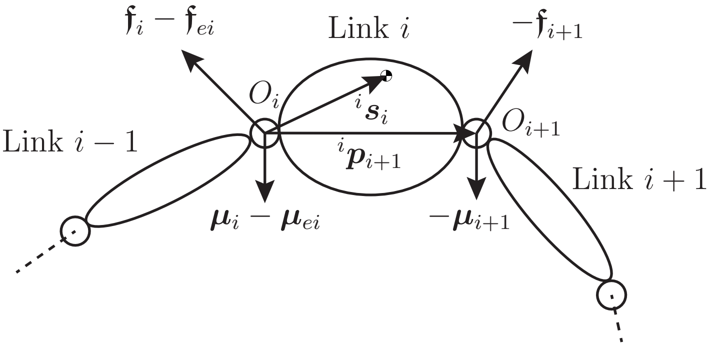

$\mathfrak{f}_i - \mathfrak{f}_{ei}$
Link i
$-\mathfrak{f}_{i+1}$
O_i
${}^i\boldsymbol{s}_i$
O_{i+1}
Link $i-1$
${}^i\boldsymbol{p}_{i+1}$
Link $i+1$
$\boldsymbol{\mu}_i - \boldsymbol{\mu}_{ei}$
$-\boldsymbol{\mu}_{i+1}$

Figure 3.2: Forces and moments on link i.

Forward Recursion

For computing (3.8) and (3.9), the linear and angular velocities here shortly denoted with $\boldsymbol{v}_i$ and $\boldsymbol{\omega}_i$, are computed with

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \boldsymbol{\omega}_{i-1} \times {}^{i-1}\boldsymbol{p}_i + \bar{\sigma}_i \dot{q}_i \mathbb{a}_i \tag{3.10}$$

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \bar{\sigma}_i \dot{q}_i \mathbb{a}_i \ , \tag{3.11}$$

where $\bar{\sigma}_i = 1 - \sigma_i$ and $\sigma_i = 0$ if the joint is revolute, like every joint of the Roberta P80.800, and $\sigma_i = 1$ if the joint is prismatic. Differentiating equations (3.10) and (3.11) with respect to time, it gives

$$\dot{\boldsymbol{v}}_i = \dot{\boldsymbol{v}}_{i-1} + \dot{\boldsymbol{\omega}}_{i-1} \times {}^{i-1}\boldsymbol{p}_i + \boldsymbol{\omega}_{i-1} \times (\dot{\boldsymbol{\omega}}_{i-1} \times {}^{i-1}\boldsymbol{p}_i) + \sigma_i(\ddot{q}_i \mathbb{a}_i + 2\boldsymbol{\omega}_{i-1} \times \dot{q}_i \mathbb{a}_i) \tag{3.12}$$

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \bar{\sigma}_i(\ddot{q}_i \mathbb{a}_i + \boldsymbol{\omega}_{i-1} \times \dot{q}_i \mathbb{a}_i) \ , \tag{3.13}$$

where the initial conditions for a robot with a fixed base are $\boldsymbol{\omega}_0 = \boldsymbol{0}$, $\dot{\boldsymbol{\omega}}_0 = \boldsymbol{0}$ and $\dot{\boldsymbol{v}}_0 = \boldsymbol{0}$.

Backward Recursion

Here for each link i s.t. $i = \mathrm{n}, \dots, 1$, we write the Newton-Euler equations at the origin of frame S_i as follows

$$\boldsymbol{F}_i = \mathfrak{f}_i - \mathfrak{f}_{i+1} + \boldsymbol{M}_i \boldsymbol{g} - \mathfrak{f}_{ei} \tag{3.14}$$

$$\boldsymbol{M}_i = \boldsymbol{\mu}_i - \boldsymbol{\mu}_{i+1} - {}^i\boldsymbol{p}_{i+1} \times \mathfrak{f}_{i+1} + {}^i\boldsymbol{s}_i \times \boldsymbol{M}_i \boldsymbol{g} - \boldsymbol{\mu}_{ei} \ , \tag{3.15}$$

where $\mathfrak{f}_{ei}$ and $\boldsymbol{\mu}_{ei}$, represent the force and moment exerted by link i on the environment, whose value is assumed to be known. We observe that the contribution of the gravity in Eq. (3.14) and (3.15) can be taken into account by setting up the initial linear acceleration as $\dot{\boldsymbol{v}}_{\boldsymbol{0}} = -\boldsymbol{g}$. Therefore we can obtain the expression for the force exerted on link i by link $i-1$ and the moment about S_i exerted on link i by link $i-1$ through

$$\mathfrak{f}_i = \boldsymbol{F}_i + \mathfrak{f}_{i+1} + \mathfrak{f}_{ei} \tag{3.16}$$

$$\boldsymbol{\mu}_i = \boldsymbol{M}_i + \boldsymbol{\mu}_{i+1} + {}^i\boldsymbol{p}_{i+1} \times \mathfrak{f}_i + \boldsymbol{\mu}_{ei} \ , \tag{3.17}$$

if we initialize the backward recursion by $\mathfrak{f}_{n+1} = 0$ and $\boldsymbol{\mu}_{n+1} = 0$. At this point, the joint torque $\boldsymbol{\tau}_i$ can be obtained by projecting either $\mathfrak{f}_i$ or $\boldsymbol{\mu}_i$ on the joint axis, depending on the prismatic or revolute nature of the joint. We can also consider the friction forces and the rotor inertia by a common model such that

$$\boldsymbol{\tau}_i = (\sigma_i \mathfrak{f}_i + \sigma_i \boldsymbol{\mu}_i)^\top \mathbb{a}_i + J_{\mathrm{d},i}\ddot{q}_i + \mathrm{F}_{\mathrm{c}i}\mathrm{sign}(\dot{q}_i) + \mathrm{F}_{\mathrm{v}i}\mathrm{sign}(\dot{q}_i) \ . \tag{3.18}$$

Practical form of the Newton-Euler algorithm

Observe that $\boldsymbol{J}_i$ and $\boldsymbol{ms}_i$ are constants when referred to their own link coordinates. From this, by referring the velocities, accelerations, forces and moments to the local link coordinate system, the forward recursion can be computed in this more efficient form, for $i = 1, \dots, n$:

$$
\begin{align}
{}^i\boldsymbol{\omega}_{i-1} &= {}^i\boldsymbol{A}_{i-1}{}^{i-1}\boldsymbol{\omega}_{i-1} \tag{3.19}\\
{}^i\boldsymbol{\omega}_i &= {}^i\boldsymbol{\omega}_{i-1} + \bar{\sigma}_i \dot{\boldsymbol{q}}_i{}^i\mathbb{a}_i \tag{3.20}\\
{}^i\dot{\boldsymbol{\omega}}_i &= \frac{d}{dt}\left({}^i\boldsymbol{\omega}_i\right) = {}^i\boldsymbol{A}_{i-1}{}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} + \bar{\sigma}_i(\ddot{\boldsymbol{q}}_i\mathbb{a}_i + {}^i\boldsymbol{\omega}_{i-1} \times \dot{\boldsymbol{q}}_i\mathbb{a}_i) \tag{3.21}\\
{}^i\dot{\boldsymbol{v}}_i &= {}^i\boldsymbol{A}_0{}^0\dot{\boldsymbol{v}}_i \tag{3.22}\\
&= {}^i\boldsymbol{A}_{i-1}({}^{i-1}\dot{\boldsymbol{v}}_{i-1} + {}^{i-1}\boldsymbol{U}_{i-1}{}^{i-1}\boldsymbol{p}_i) + \sigma_i(\ddot{\boldsymbol{q}}_i\mathbb{a}_i + 2{}^i\boldsymbol{\omega}_{i-1} \times \dot{\boldsymbol{q}}_i\mathbb{a}_i) \tag{3.23}\\
{}^i\boldsymbol{F}_i &= \boldsymbol{M}_i{}^i\dot{\boldsymbol{v}}_i + {}^i\boldsymbol{U}_i\boldsymbol{ms}_i \tag{3.24}\\
{}^i\boldsymbol{M}_i &= {}^i\boldsymbol{J}_i{}^i\dot{\boldsymbol{\omega}}_i + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{J}_i{}^i\boldsymbol{\omega}_i) + \boldsymbol{ms}_i \times {}^i\dot{\boldsymbol{v}}_i \tag{3.25}\\
{}^i\boldsymbol{U}_i &= {}^i\hat{\dot{\boldsymbol{\omega}}}_i + {}^i\hat{\boldsymbol{\omega}}_i{}^i\hat{\boldsymbol{\omega}}_i \ , \tag{3.26}
\end{align}
$$

where, for a stationary base, the initial conditions are such that

$$
\begin{align}
\boldsymbol{\omega}_0 &= \boldsymbol{0} \tag{3.27}\\
\dot{\boldsymbol{\omega}}_0 &= \boldsymbol{0} \tag{3.28}\\
\dot{\boldsymbol{v}}_0 &= -\boldsymbol{g} \ . \tag{3.29}
\end{align}
$$

Therefore, the backward recursive equations, for $i = n, \dots, 1$ gives

$$
\begin{align}
{}^i\mathfrak{f}_i &= {}^i\boldsymbol{F}_i + {}^i\mathfrak{f}_{i+1} + {}^i\mathfrak{f}_{\mathrm{e}i} \tag{3.30}\\
{}^{i-1}\mathfrak{f}_i &= {}^{i-1}\boldsymbol{A}_i{}^i\mathfrak{f}_i \tag{3.31}\\
{}^i\boldsymbol{\mu}_i &= {}^i\boldsymbol{M}_i + {}^i\boldsymbol{A}_{i+1}{}^{i+1}\boldsymbol{\mu}_{i+1} + {}^i\boldsymbol{p}_{i+1} \times {}^i\mathfrak{f}_{i+1} + {}^i\boldsymbol{\mu}_{\mathrm{e}i} \ , \tag{3.32}
\end{align}
$$

which finally bring useful information for finding the base wrench. From these last equations it is also possible to derive the joint torque, that is

$$\boldsymbol{\tau}_i = (\sigma_i{}^i\mathfrak{f}_i + \bar{\sigma}_i{}^i\boldsymbol{\mu}_i)^\top \mathbb{a}_i + J_{\mathrm{d},i}\ddot{q}_i + \mathrm{F}_{\mathrm{c}i}\mathrm{sign}(\dot{q}_i) + \mathrm{F}_{\mathrm{v}i}\mathrm{sign}(\dot{q}_i) \ . \tag{3.33}$$

This practical form of the algorithm can be easily implemented for a general serial robot, and it has a computational complexity of O(n), which means that the number of operations is linear in the number of degrees of freedom. Here we also want to remark that the number of operations required, is even more reduced whether we use a formulation based on the *base inertial parameters*, that we are going to introduce in the next sections.

3.3.2 Moving Base Manipulator

The practical form of the Newton-Euler algorithm that we just reported, provides a formulation for the base wrench, for the case of a static base manipulator. On this purpose, we are also interested in consider the case of a moving base manipulator, that can happen for example when it is mounted on a mobile base, that is our study case.
In this eventuality, the motion of the platform plays a relevant role on the base reactions forces, therefore we need to include all the dynamic effects that the platform exerts on the manipulator, within the dynamics equations. These effects can be included in te manipulator, by implementing the Newton-Euler recursions with the proper initial conditions stated in (3.27)-(3.29), that have to be set taking the platform linear and angular velocities.

3.3.3 Inverse Dynamics Linear in the Base Parameters

Though in Eq. (3.33) we have obtained an expression for the inverse dynamics of a 6-DoF serial link manipulator, this formulation cannot be applied to the Roberta P80.800 for the fact that parameters like link masses and inertias needed for the computation, are unknown.

In the work of M. Stapf [4] an expression for the equations of the inverse dynamics as a linear function of the inertial parameters, has been found for the Roberta P80.800 through the Lagrangian formalism, such that

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{W}_{\text{std}}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}_{\text{std}} \ , \tag{3.34}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ denotes the mass matrix of inertia, $\boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ the matrix of Coriolis and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{6\times 1}$ the vector of torques induced by gravity, so $\boldsymbol{\tau} \in \mathbb{R}^{6\times 1}$ contains the remaining joint torques to change the state of the system (see [4]). As we can see, Eq. (3.34) is linear w.r.t. $\boldsymbol{\kappa}_{\text{std}} \in \mathbb{R}^{66\times 1}$ that denotes the vector of all dynamic parameters

$$\boldsymbol{\kappa}_{\text{std}} = \begin{bmatrix} \boldsymbol{\kappa}_{\text{std},1}^{\top} & \boldsymbol{\kappa}_{\text{std},2}^{\top} & \dots & \boldsymbol{\kappa}_{\text{std},6}^{\top} \end{bmatrix}^{\top} , \tag{3.35}$$

while $\boldsymbol{W}_{\text{std}}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 66}$ denotes the matrix of inverse dynamics. At this point, see [4], it can be algebraically proven that not every dynamic parameter bring its contribution on the equation of the inverse kinematics. This reasonably means that there is no way to identify these parameters, and just a smaller subset can be identified uniquely from measured data. Under this fact, Eq. (3.34) can be rewritten as

$$\boldsymbol{\tau} = \boldsymbol{W}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa} \ , \tag{3.36}$$

where $\boldsymbol{\kappa} \in \mathbb{R}^{40}$ represents the base inertial parameters, derived by Stapf in Tab. 3.3, that is the minimum set of parameters needed to compute the dynamics, deduced by eliminating from $\boldsymbol{\kappa}_{\text{std}}$ the ones that have no effect on the dynamic model and by grouping some others, in order to get the computations more efficient.

3.3.4 Base Wrench

From the Newton-Euler formulation we can now derive an expression for the reaction forces at the base of the manipulator. If we identify the platform as the link 0, by applying Eq. (3.30) and (3.32) for $i = n, \dots, 1$ we can compute ${}^{1}\mathfrak{f}_1$, force exerted on link 1 by the platform, and ${}^{1}\boldsymbol{\mu}_1$, which is the moment about frame S_1 exerted on link 1 by the platform.

According with the transformations of force and torque, we can describe these quantities w.r.t. the frame S_0, therefore Eq. (3.2) becomes ${}^{0}\boldsymbol{f} = {}^{0}\boldsymbol{X}_1 {}^{1}\boldsymbol{f}$, which gives

$$\boldsymbol{f}^{\mathrm{w}} = \begin{bmatrix} {}^{0}\mathfrak{f}_1 \\ {}^{0}\boldsymbol{\mu}_1 \end{bmatrix} = \begin{bmatrix} {}^{0}\boldsymbol{A}_1 & \boldsymbol{0}_{3\times 3} \\ {}^{0}\boldsymbol{S}_1 \cdot {}^{0}\boldsymbol{A}_1 & {}^{0}\boldsymbol{A}_1 \end{bmatrix} \begin{bmatrix} {}^{1}\mathfrak{f}_1 \\ {}^{1}\boldsymbol{\mu}_1 \end{bmatrix} \quad . \tag{3.37}$$

Finally we have obtained an expression for the reaction force and moment at the platform base. Observe that from Eq. (3.37), the force-torque that the manipulator exerts on the platform is given by

$$\boldsymbol{f}_{\mathrm{Ex,Man}} = -\boldsymbol{f}^{\mathrm{w}}. \tag{3.38}$$

3.3.5 Base Wrench Dynamics Verification

The expression of the manipulator base wrench, previously formulated through the Newton-Euler approach, can be verified in this section. In fact, we recall that this expression is computed through forces and moment in Eq. (3.30)-(3.32), that are also involved in the dynamic model.

Observe that the force in Eq. (3.31) is involved in the expression of the moment in Eq. (3.32), which is further involved in the manipulator dynamic model in Eq. (3.33), if computed for the case of the 6 revolute joint Roberta. This means that if we are able to verify the computed dynamic model, we implicitly verify the forces and moments that also compose the base wrench formulation in (3.37).

On this purpose, we recall that in the work of M. Stapf [4], a dynamic model for the Roberta P80.800 has been derived through the Lagrangian formulation. Therefore, implementing in MATLAB the practical form of the Newton-Euler algorithm presented previously, it has been verified the equivalence between the Newtonian dynamic model and the Lagrangian one provided by M. Stapf in its work (Eq. (3.34)), where the friction coefficients have been neglected. As a result of this, for the reason previously mentioned, we verify the computed expression for the base wrench.

3.3.6 Linear Expressions for the Base Wrench

Although Eq. (3.37) provides the needed expression for the base wrench, acting on frame S_0, we remind the fact that this recursive computation cannot be performed due to the fact that some dynamic parameters have not been identified. For the same reason, M. Stapf in [4] addressed his work in identifying the base parameters, that could let the dynamic equations be computed, thanks to the relationship reported in (3.36), which is linear w.r.t.

$\boldsymbol{\kappa}$. Consequently, the easiest way to compute the base wrench, would be to exploit the base parameters already identified by Stapf, taking inspiration from the work made in [18] by Grotjahn and Heimann, finding a relationship like

$$\boldsymbol{f}^{\mathrm{w}} = \boldsymbol{W}^{*}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa} \ . \tag{3.39}$$

On this purpose, is not possible to numerically evaluate the Roberta base wrench through a relationship like Eq. (3.39), e.g. for the fact that it depends on parameters that are out of the base parameter set $\boldsymbol{\kappa}$, which have been neglected by the authors in [18]. Anyways in this work, through $\boldsymbol{\kappa}^f$ we are able to compute an expression $\hat{\boldsymbol{f}}^{\mathrm{w}} \approx \boldsymbol{f}^{\mathrm{w}}$ s.t.

$$\hat{\boldsymbol{f}}^{\mathrm{w}} = \boldsymbol{W}^{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa}^f \ , \tag{3.40}$$

which represents the formulation of the base wrench that we are able to compute numerically, and provides an approximation of the desired $\boldsymbol{f}^{\mathrm{w}}$. Here $\boldsymbol{W}^{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 41}$ and $\boldsymbol{\kappa}^f \in \mathbb{R}^{41}$ different from $\boldsymbol{\kappa}$, is denoted as wrench parameter set. At this point it is worthy to better clarify the differences between these two parameter sets.

1. $\boldsymbol{\kappa}^f$ contains five new parameters w.r.t. $\boldsymbol{\kappa}$, that are grouped in a subset $\boldsymbol{\kappa}^{\mathrm{w}}$ shown in Table 3.1. This means that $\boldsymbol{f}^{\mathrm{w}}$ is function of some parameters that have not been identified yet, which is a disadvantage that makes our base wrench computation more complex to derive.

Table 3.1: Additional base inertial parameters in $\boldsymbol{\kappa}^f$.

Wrench Parameter $\boldsymbol{\kappa}^{\mathrm{w}}$	**Standard parameter $\boldsymbol{\kappa}_{\mathrm{std}}$**
$J_{xz\mathrm{R},1}$	$J_{xz,1}$
$J_{yz\mathrm{R},1}$	$J_{yz,1}$
$ms_{x\mathrm{R},1}$	$ms_{x,1}$
ms_{R}	$ms_{y,1} - ms_{z,2} - ms_{z,3}$
m_{R}	$m_1 + m_2 + m_3 + m_4 + m_5 + m_6$

2. From Eq. (3.30) - (3.33) it is straightforward to understand that the $J_{\mathrm{d},i}$s, inertia of the i-th rotor that appears in the base parameter set, is involved in the computation of the torque, but it does not play any role in the forces and moments acting on each link. This means that $\boldsymbol{\kappa}^f$ has to be free of all the $J_{\mathrm{d},i}$ terms.

On the considerations made previously, $\boldsymbol{\kappa}^f$ has been arranged in Table 3.4 where, putting the focus on the grouped parameters $J_{zz\mathrm{R},1}$ and $J_{zz\mathrm{R},2}$, we can observe that they still contain respectively $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$, despite what has just been said. Here unfortunately, to get rid of these parameters, we would need to identify their value. As a result, comparing the symbolic base wrench computed in (3.37) through the Newton-Euler formulation, with its expression linear in the wrench parameters (3.40), we obtain the following systematic error $\boldsymbol{\varepsilon} \in \mathbb{R}^{6\times 1}$ shown in Tab. 3.2, computed through

$$\boldsymbol{\varepsilon}(J_{\mathrm{d},1}, J_{\mathrm{d},2}) = \boldsymbol{f}^{\mathrm{w}} - \hat{\boldsymbol{f}}^{\mathrm{w}} = \boldsymbol{f}^{\mathrm{w}} - \boldsymbol{W}^{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa}^f \ . \tag{3.41}$$

Table 3.2: Symbolic error for every wrench component, computed through (3.41).

Error Component ε_i	**Symbolic Error**
ε_1	0
ε_2	0
ε_3	0
ε_4	$-J_{\mathrm{d},2}\left(\ddot{q}_2 \sin q_1 + \dot{q}_1\dot{q}_2 \cos q_1\right)$
ε_5	$J_{\mathrm{d},2}\left(\ddot{q}_2 \cos q_1 - \dot{q}_1\dot{q}_2 \sin q_1\right)$
ε_6	$-J_{\mathrm{d},1}\ddot{q}_1$

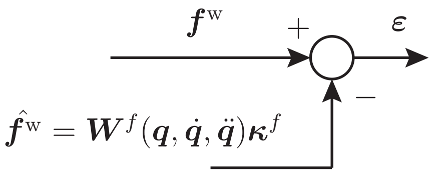

$\boldsymbol{f}^{\mathrm{w}}$
$+$
$\boldsymbol{\varepsilon}$
$-$
$\hat{\boldsymbol{f}}^{\mathrm{w}} = \boldsymbol{W}^f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa}^f$

Figure 3.3: Derivation of $\boldsymbol{\varepsilon}$.

Here we can observe that the error is due to the mentioned undesired presence of $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$, in the wrench parameter set. Summarizing, in this section we introduced two sources of uncertainty that are:

- $\boldsymbol{\kappa}^{\mathrm{w}}$ parameters actively involved in $\boldsymbol{f}^{\mathrm{w}}$, but not identified, so not involved when we numerically evaluate $\hat{\boldsymbol{f}}^{\mathrm{w}}$,
- a systematic error due to the presence of $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$, whose value is unknown.

To conclude, as soon as we can estimate all these non identified parameters above mentioned, we can get rid of the two sources of error, obtaining a consistent formulation for $\hat{\boldsymbol{f}}^{\mathrm{w}}$.

3.4 Theory of Parameter Identification

Whenever in a mathematical model, play a role one or more parameters whose value is unknown and not directly measurable, the theory of parameters identification comes in help, providing algebraic tools for estimating such parameters from measured data.

Let $\boldsymbol{\kappa} \in \mathbb{R}^r$ contain the r parameters of interest and $\boldsymbol{x} \in \mathbb{R}^N$ be the vector of the known excitation for the N measurements. The system output can be modeled through

$$\boldsymbol{y}_k = \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{\kappa}) + \boldsymbol{n}_k \quad , \quad k = 1, \ldots, N \tag{3.42}$$

that can also be stacked in vectors in the form

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\kappa}) + \boldsymbol{n} \quad , \tag{3.43}$$

where $\boldsymbol{n}$ is a random vector with an unknown probability distribution. We assume $\boldsymbol{x}_k$s deterministic, while $\boldsymbol{y}$ is not known exactly since the noise $\boldsymbol{n}$ is not known. Therefore the estimated system output $\hat{\boldsymbol{y}}$ results for an estimated parameter set $\hat{\boldsymbol{\kappa}}$ as

$$\hat{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{x}, \hat{\boldsymbol{\kappa}}) + \hat{\boldsymbol{n}} \quad . \tag{3.44}$$

The goal of the theory of parameter estimation is to find a parameter set $\hat{\boldsymbol{\kappa}}$ which gives the optimal system output $\hat{\boldsymbol{y}}$ that has to be as close as possible to the real output of the system $\boldsymbol{y}$. Further information regarding the topic, can be found in [24].

Therefore, once we have obtained an estimation $\hat{\boldsymbol{y}}$, the step next is to evaluate the goodness of the estimates performed. A first easy evaluation, can be done comparing the plots of the measurement $\boldsymbol{y}$ with the ones of the estimate $\hat{\boldsymbol{y}}$, that gives a rough and qualitative outcome. On the other side, several numeric evaluations of the estimation error can be easily applied, in particular we mention the *sum of squares due to error SSE*, defined as

$$SSE = [\boldsymbol{y} - \hat{\boldsymbol{y}}]^\top \boldsymbol{\mathcal{W}} [\boldsymbol{y} - \hat{\boldsymbol{y}}] = \sum_{i=1}^{N} \boldsymbol{\mathcal{W}}_{i,i} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2 \quad , \tag{3.45}$$

with $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{N \times N}$ optional positive definite diagonal weighting matrix. From this, it can be derived the *mean squared error MSE*

$$MSE = \frac{SSE}{M} \quad , \tag{3.46}$$

where M is the residual degree of freedom, defined as $M = N - r$, that depends on the number of measurement points N and on the fitting parameters r. A further evaluation can be done through the *root mean squared error RMSE*, also known as *standard error of the regression*, that is

$$RMSE = \sqrt{MSE} \quad . \tag{3.47}$$

3.4.1 Least Squares Method

The literature presents several approaches for the parameter identification, based on the estimation error, rather than giving the observation $\boldsymbol{y}$ more likely a posteriori. In this section is presented the least squares method, which has been used in this work. Consider the generic stacked model

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix}}_{\tilde{\boldsymbol{y}}_N} = \underbrace{\begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{x}_1, \boldsymbol{\kappa}) \\ \boldsymbol{g}_2(\boldsymbol{x}_2, \boldsymbol{\kappa}) \\ \vdots \\ \boldsymbol{g}_N(\boldsymbol{x}_N, \boldsymbol{\kappa}) \end{bmatrix}}_{\tilde{\boldsymbol{g}}_N(\boldsymbol{x}_N, \boldsymbol{\kappa})} + \sigma \underbrace{\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix}}_{\tilde{\boldsymbol{v}}_N} . \tag{3.48}$$

Let us assume that $\boldsymbol{\kappa}$ are the parameters to estimate which are unknown, as well as $\sigma^2 \in \mathbb{R}_+$, and $\boldsymbol{v}$ is a random vector with zero mean and positive definite variance matrix $\boldsymbol{R}$ with an unknown distribution. We also assume that $\boldsymbol{x}_k$s are deterministic.
Assume for one moment that $\boldsymbol{\kappa}$ and $\boldsymbol{x}_k$s are given and $\boldsymbol{y}_k$s are not given. From (3.48), the prediction error for the estimation and the measured values is given by

$$\boldsymbol{\varepsilon}_k(\boldsymbol{\kappa}) = \tilde{\boldsymbol{y}}_k - \tilde{\boldsymbol{g}}_k(\boldsymbol{x}_k, \boldsymbol{\kappa}) \quad , \tag{3.49}$$

where an estimate of $\boldsymbol{\kappa}$ is given by minimizing the fit term

$$\sum_{k=1}^{N} \boldsymbol{\varepsilon}_k^2 = \boldsymbol{\varepsilon}(\boldsymbol{\kappa})^\top \boldsymbol{\varepsilon}(\boldsymbol{\kappa}) \quad , \tag{3.50}$$

where we did not consider the $\boldsymbol{R}$ variance matrix of $\boldsymbol{v}$. Now considering the case with $\boldsymbol{R} = diag\{r_1, \dots, r_N\}$ positive definite, it can be proven (see [24]) that the minimization problem in Eq. (3.50) for the general case with $\boldsymbol{R}$ taken in an arbitrary structure, is equivalent to

$$\hat{\boldsymbol{\kappa}}_{\mathrm{LS}}(\tilde{\boldsymbol{y}}_N) = \underset{\hat{\boldsymbol{\kappa}}}{\operatorname{argmin}} \quad \frac{1}{\sigma^2} \boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}})^\top \boldsymbol{R}^{-1} \boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}}) \quad , \tag{3.51}$$

that is called the *Weighted Least-Squares (LS) estimator.*

3.4.2 LS Estimator for Linear Models

In the model theory, the linearity is an important property, in particular is wished in the parameter identification topic, for the fact that often provides in a computationally convenient way, compact form solutions for the minimization problems, like the one dealt in the LS method. We now consider a linear model at sample k, with m system inputs $\boldsymbol{x}_k \in \mathbb{R}^m$, and n system outputs $\boldsymbol{y}_k \in \mathbb{R}^n$ like

$$\boldsymbol{y}_k = \boldsymbol{W}_k(\boldsymbol{x}_k)\boldsymbol{\kappa} + \sigma \boldsymbol{v}_k \quad , \tag{3.52}$$

with $\boldsymbol{W}_k(\boldsymbol{x}_k) \in \mathbb{R}^{n \times r}$ regressor, $\boldsymbol{\kappa} \in \mathbb{R}^r$, $\boldsymbol{v}_k \in \mathbb{R}^n$ random vector with unknown probability distribution and σ s.t. $\sigma^2 \in \mathbb{R}_+$. Now observe that Eq. (3.52) represents a linear equation in r unknown parameters, which yields a solution once we provide $N \geq r$ equations in the same parameter vector $\boldsymbol{\kappa}$. In other words this means that, in order to have enough information, we need to collect $N \geq r$ measures of $\boldsymbol{y}_k$ resulting from N independent input excitations $\boldsymbol{x}_k$. Thus we can collect N measurement such that

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix}}_{\tilde{\boldsymbol{y}}_N} = \underbrace{\begin{bmatrix} \boldsymbol{W}_1(\boldsymbol{x}_1) \\ \boldsymbol{W}_2(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{W}_N(\boldsymbol{x}_N) \end{bmatrix}}_{\tilde{\boldsymbol{W}}_N(\boldsymbol{x}_N)} \boldsymbol{\kappa} + \sigma \underbrace{\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix}}_{\tilde{\boldsymbol{v}}_N} \tag{3.53}$$

where $\tilde{\boldsymbol{y}}_N \in \mathbb{R}^{n \cdot N}$, $\tilde{\boldsymbol{v}}_N \in \mathbb{R}^{n \cdot N}$ random vector with zero mean and $\boldsymbol{R}$ variance matrix, $\tilde{\boldsymbol{W}}_N \in \mathbb{R}^{n \cdot N \times r}$, $\boldsymbol{x}_N \in \mathbb{R}^{m \cdot N}$ and $\boldsymbol{\kappa} \in \mathbb{R}^r$. Therefore, the parameter identification problem in the case of linear models, is reduced to find the solution of a system of a $n \cdot N$ linear independent equations, where according to (3.53) it has to be $n \cdot N = r$ and $\operatorname{rank}(\tilde{\boldsymbol{W}}_N)(\boldsymbol{x}_N) = r$. The vector $\hat{\boldsymbol{\kappa}}^*$ that best fits the measurement values should be estimated, hence the error vector of the estimation and the measured values is defined as

$$\boldsymbol{\varepsilon} = \tilde{\boldsymbol{y}} - \tilde{\boldsymbol{W}}(\boldsymbol{x})\hat{\boldsymbol{\kappa}} \quad , \tag{3.54}$$

where we omitted the subscript N. Therefore, in this case Eq. (3.51) gives

$$\hat{\boldsymbol{\kappa}}_{\mathrm{LS}}(\tilde{\boldsymbol{y}}) = \underset{\hat{\boldsymbol{\kappa}}}{\operatorname{argmin}} \ \frac{1}{\sigma^2} \boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}})^\top \boldsymbol{R}^{-1} \boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}}) \ , \tag{3.55}$$

where we observe that the minimization in Eq. (3.55) is equivalent to

$$\hat{\boldsymbol{\kappa}}_{\mathrm{LS}}(\tilde{\boldsymbol{y}}) = \underset{\hat{\boldsymbol{\kappa}}}{\operatorname{argmin}}\ \boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}})^{\top}\boldsymbol{R}^{-1}\boldsymbol{\varepsilon}(\hat{\boldsymbol{\kappa}}) = \underset{\hat{\boldsymbol{\kappa}}}{\operatorname{argmin}}\ \psi_{\mathrm{LS}}\ \ . \tag{3.56}$$

At this point, in order to find the minimum, the gradient of ψ_{LS} with respect to $\boldsymbol{\kappa}$, has to be equal to zero, so

$$\left.\frac{\mathrm{d}\psi_{\mathrm{LS}}(\hat{\boldsymbol{\kappa}})}{\mathrm{d}\hat{\boldsymbol{\kappa}}}\right|_{\hat{\boldsymbol{\kappa}}\,=\,\hat{\boldsymbol{\kappa}}^{*}} = \frac{\mathrm{d}}{\mathrm{d}\hat{\boldsymbol{\kappa}}}\Big((\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{W}}\hat{\boldsymbol{\kappa}})^{\top}(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{W}}\hat{\boldsymbol{\kappa}})\Big) \tag{3.57}$$

$$= -2\tilde{\boldsymbol{W}}^{\top}\tilde{\boldsymbol{y}} + 2\tilde{\boldsymbol{W}}^{\top}\tilde{\boldsymbol{W}}\hat{\boldsymbol{\kappa}} = 0\ \ . \tag{3.58}$$

Furthermore, the second order derivative

$$\frac{\mathrm{d}^2\psi_{\mathrm{LS}}(\hat{\boldsymbol{\kappa}})}{\mathrm{d}\hat{\boldsymbol{\kappa}}\mathrm{d}\hat{\boldsymbol{\kappa}}^{\top}} = 2\tilde{\boldsymbol{W}}^{\top}\tilde{\boldsymbol{W}}\ \ , \tag{3.59}$$

has to be positively definite in the neighborhood of the solution. In this way ψ_{LS} is a convex function in $\boldsymbol{\kappa}$, and the local minimum found by imposing the gradient equal to zero, is also a global minimum. Finally, from (3.58), under the assumption of $\mathrm{rank}(\tilde{\boldsymbol{W}}^{\top}\tilde{\boldsymbol{W}}) = r$ we can obtain

$$\hat{\boldsymbol{\kappa}}_{\mathrm{LS}}(\tilde{\boldsymbol{y}}) = \left(\tilde{\boldsymbol{W}}^{\top}\boldsymbol{R}^{-1}\tilde{\boldsymbol{W}}\right)^{-1}\tilde{\boldsymbol{W}}^{\top}\boldsymbol{R}^{-1}\tilde{\boldsymbol{y}} \tag{3.60}$$

that represents the *Weighted Least-Squares (LS) estimator for linear models.*

3.5 Base Wrench Adjustment

After having introduced useful elements of the Parameter Identification topic in section 3.4, we want to exploit the reported results for adjust the formulation of the base wrench $\hat{\boldsymbol{f}}^{\mathrm{w}}$, trying to improve it in multiple steps.

3.5.1 Systematic Error Compensation

In order to improve the approximation of the base wrench in (3.40), we first aim to get rid of the symbolic systematic error shown in Tab. 3.2. This error comes from the fact that we tried to fit the base parameter set, built for the inverse dynamics, into the computation of the base reaction forces, which led to the undesired presence of $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ in it.
On this purpose, we make an initial assumption. Since we have $\boldsymbol{\kappa}^{f} = \boldsymbol{\kappa} \cup \boldsymbol{\kappa}^{\mathrm{w}}$, here we build a new parameter set $\boldsymbol{\kappa}^{f*} \in \mathbb{R}^{41}$ assuming that all the parameters of $\boldsymbol{\kappa}^{f}$ within $\boldsymbol{\kappa}^{\mathrm{w}}$, that correspond to the entries 1, 2, 4, 5, 6 of Table (3.4), are equal to zero.

As a consequence of this assumption, we now symbolically compute $\hat{\boldsymbol{f}}^{\mathrm{w}*} = \boldsymbol{W}^{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*}$, so Eq. (3.41) now becomes

$$\boldsymbol{\varepsilon}(J_{\mathrm{d},1},J_{\mathrm{d},2},\boldsymbol{\kappa}^{\mathrm{w}}) = \boldsymbol{f}^{\mathrm{w}} - \hat{\boldsymbol{f}}^{\mathrm{w}*} = \boldsymbol{f}^{\mathrm{w}} - \boldsymbol{W}^{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*} \tag{3.61}$$

$$= \boldsymbol{\varepsilon}_1(J_{\mathrm{d},1},J_{\mathrm{d},2}) + \boldsymbol{\varepsilon}_2(\boldsymbol{\kappa}^{\mathrm{w}})\ \ . \tag{3.62}$$

where now comes out a new symbolic term of error $\boldsymbol{\varepsilon}_2(\boldsymbol{\kappa}^{\mathrm{w}})$, due to the fact that we are now neglecting all the unknown parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$. We observe that:

- Eq. (3.61) represents the symbolic error between the desired formulation of the base wrench, which is not numerically computable, and its computable approximation $\boldsymbol{f}^{\hat{\mathrm{w}}*}$. Furthermore, giving a specific trajectory $(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$, if we take $\boldsymbol{f}^{\mathrm{w}}$ as readings from a force-torque sensor, now denoted with $\boldsymbol{f}^{\tilde{\mathrm{w}}}$, and $\boldsymbol{f}^{\hat{\mathrm{w}}*}$ from numeric simulations, this error becomes $\tilde{\boldsymbol{\varepsilon}}$ and it is measurable.

- As we are ignoring for the moment all the unknown parameters of $\boldsymbol{\kappa}^{\mathrm{w}}$, despite its symbolic description also includes terms of $\boldsymbol{\kappa}^{\mathrm{w}}$, this computed error $\tilde{\boldsymbol{\varepsilon}}$ can just be modeled as

$$\begin{aligned}\boldsymbol{\varepsilon}_1(J_{\mathrm{d},1},J_{\mathrm{d},2}) &= \boldsymbol{f}^{\tilde{\mathrm{w}}} - \boldsymbol{f}^{\hat{\mathrm{w}}*} \\ &= \boldsymbol{f}^{\tilde{\mathrm{w}}} - \boldsymbol{W}^f(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*} = -\boldsymbol{W}_{\mathrm{J}}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{\mathrm{J}} \ ,\end{aligned} \tag{3.63}$$

where $\boldsymbol{\kappa}^{\mathrm{J}} = [J_{\mathrm{d},1}, J_{\mathrm{d},2}]^{\top}$, and $\boldsymbol{W}_J \in \mathbb{R}^{6\times 2}$ consists in the over contribution of $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ in the formulation of $\boldsymbol{f}^{\hat{\mathrm{w}}*}$ in Eq. (3.40). As we expect, it is made by the columns 3 and 11 of the matrix $\boldsymbol{W}^f(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ that are the coefficients which multiply the entries in $\boldsymbol{\kappa}^{f*}$, that contain $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ in a linear form.

$$\begin{aligned}\boldsymbol{\varepsilon}_1(J_{\mathrm{d},1},J_{\mathrm{d},2}) &= -\begin{bmatrix} W^{f,3}, & W^{f,11}\end{bmatrix}\boldsymbol{\kappa}^{\mathrm{J}} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\left(\ddot{q}_2\sin q_1 + \dot{q}_1\dot{q}_2\cos q_1\right) \\ 0 & -\left(\dot{q}_1\dot{q}_2\sin q_1 - \ddot{q}_2\cos q_1\right) \\ -\ddot{q}_1 & 0 \end{bmatrix}\begin{bmatrix} J_{\mathrm{d},1} \\ J_{\mathrm{d},2}\end{bmatrix} \ .\end{aligned} \tag{3.64}$$

Therefore, the underlying idea is first to identify the parameters $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ from the computed error $\tilde{\boldsymbol{\varepsilon}}$, then evaluate the symbolic error $\boldsymbol{\varepsilon}_1(J_{\mathrm{d},1},J_{\mathrm{d},2})$ at each instant time, and from Eq. (3.63), adjusting $\boldsymbol{f}^{\hat{\mathrm{w}}*}$ s.t.

$$\boldsymbol{f}^{\bar{\mathrm{w}}} = \boldsymbol{f}^{\hat{\mathrm{w}}*} + \boldsymbol{\varepsilon}_1(\hat{J}_{\mathrm{d},1},\hat{J}_{\mathrm{d},2}) \ . \tag{3.65}$$

The followed approach is depicted in the block scheme in Fig. 3.4. The block Σ_1 estimates the parameters $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ from the measured error $\tilde{\boldsymbol{\varepsilon}}$, and Σ_2 evaluates the symbolic error (3.63) in the estimated parameters $\hat{J}_{\mathrm{d},1}$ and $\hat{J}_{\mathrm{d},2}$. This compensation term obtained, is further summed to the initial formulation of the base wrench $\boldsymbol{f}^{\hat{\mathrm{w}}*}$, giving a new $\boldsymbol{f}^{\bar{\mathrm{w}}}$, base wrench without the contribution of the first two rotors inertia. As we mentioned, the

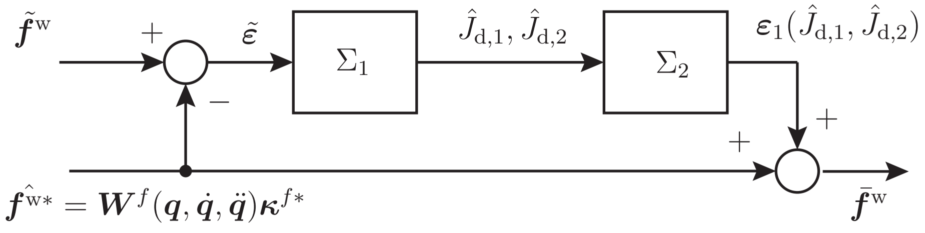

$\boldsymbol{f}^{\tilde{\mathrm{w}}}$
+
$\tilde{\boldsymbol{\varepsilon}}$
Σ_1
$\hat{J}_{\mathrm{d},1},\hat{J}_{\mathrm{d},2}$
Σ_2
$\boldsymbol{\varepsilon}_1(\hat{J}_{\mathrm{d},1},\hat{J}_{\mathrm{d},2})$
−
+
+
$\boldsymbol{f}^{\hat{\mathrm{w}}*} = \boldsymbol{W}^f(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*}$
$\boldsymbol{f}^{\bar{\mathrm{w}}}$

Figure 3.4: Systematic error adjustment scheme.

block Σ_1 takes advantage of the fact that the parameters to identify, appear in a linear form within the symbolic expression of the error $\boldsymbol{\varepsilon}(J_{\mathrm{d},1}, J_{\mathrm{d},2})$.

Therefore, whether we can collect N measurements of the error $\tilde{\boldsymbol{\varepsilon}}$, we can obtain a model according with Eq. (3.53), and through an LS estimator (3.60), we can obtain

$$\hat{\boldsymbol{\kappa}}^{\mathrm{J}}_{\mathrm{LS}}(\tilde{\boldsymbol{\varepsilon}}) = \left(\boldsymbol{W}_{\mathrm{J}}^{\top}\boldsymbol{R}^{-1}\boldsymbol{W}_{\mathrm{J}}\right)^{-1}\boldsymbol{W}_{\mathrm{J}}^{\top}\boldsymbol{R}^{-1}\tilde{\boldsymbol{\varepsilon}} \ . \tag{3.66}$$

This equation provides the required estimates $\hat{J}_{\mathrm{d},1}$ and $\hat{J}_{\mathrm{d},2}$, where we can assume $\boldsymbol{R} = \mathbf{I}_{3\times 3}$. Here we highlight the fact that this compensation error still offers a rough improvement, for the fact that in modeling the measured error $\tilde{\boldsymbol{\varepsilon}}$, the symbolic component $\boldsymbol{\varepsilon}_2(\boldsymbol{\kappa}^{\mathrm{w}})$ in (3.62) was not considered.

3.5.2 Estimation of the Non Identified Parameters

In section 3.5.1 the systematic error has been dealt starting from the assumption that all the non identified parameters were equal to zero. This led the symbolic error to be written as in Eq. (3.62), although in the compensation part, the all error was considered due to the presence of parameters $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$.
Now, for a consistent formulation of the base wrench, a more accurate modeling of the error is needed, which has also to consider the missing contribution of the $\boldsymbol{\kappa}^{\mathrm{w}}$ parameter set, fully according with (3.61)-(3.62). In this part, we want to exploit the parameter identification knowledge to estimate the non identified parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$, in order to be able to compensate also their contribution. As we aim to identify at the same time parameters that have a different nature, we want to employ an unified approach in order to take care of the combined effects of either $\boldsymbol{\kappa}^{\mathrm{w}}$ and $\boldsymbol{\kappa}^{\mathrm{J}}$ parameter set. Thus, according with (3.52), we build a linear model of type

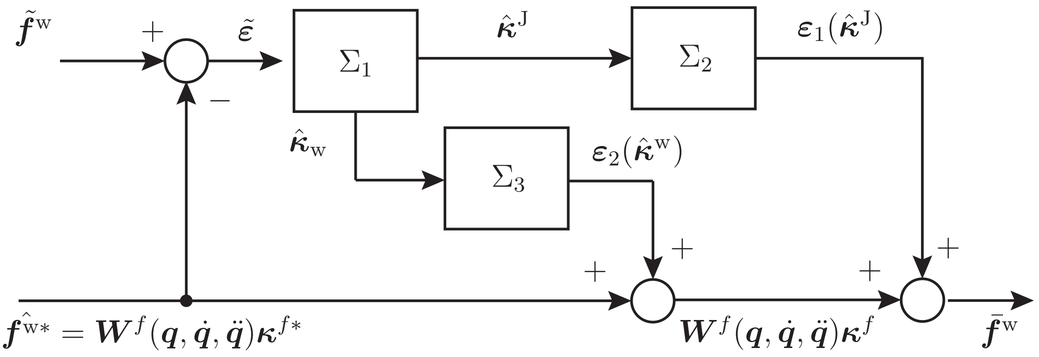

$\tilde{\boldsymbol{f}}^{\mathrm{w}}$
+
$\tilde{\boldsymbol{\varepsilon}}$
Σ_1
$\hat{\boldsymbol{\kappa}}^{\mathrm{J}}$
Σ_2
$\boldsymbol{\varepsilon}_1(\hat{\boldsymbol{\kappa}}^{\mathrm{J}})$
−
$\hat{\boldsymbol{\kappa}}_{\mathrm{w}}$
Σ_3
$\boldsymbol{\varepsilon}_2(\hat{\boldsymbol{\kappa}}^{\mathrm{w}})$
+
+
+
+
$\hat{\boldsymbol{f}}^{\mathrm{w}*} = \boldsymbol{W}^{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*}$
$\boldsymbol{W}^{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f}$
$\bar{\boldsymbol{f}}^{\mathrm{w}}$

Figure 3.5: Non identified parameters contribution is now taken into account.

$$\tilde{\boldsymbol{\varepsilon}}_k = \boldsymbol{W}_k^{\mathrm{ext}}(\boldsymbol{x}_k)\boldsymbol{\kappa}^{\mathrm{ext}} + \sigma\boldsymbol{v}_k \ , \tag{3.67}$$

where the parameter vector $\boldsymbol{\kappa}^{\mathrm{ext}}$ is extended to contain all the parameters to identify, respectively the first two rotor inertias and the components of $\boldsymbol{\kappa}^{\mathrm{w}}$, that results

$$\boldsymbol{\kappa}^{\mathrm{ext}} = \left[J_{\mathrm{d},1}, \quad J_{\mathrm{d},2}, \quad J_{xz\mathrm{R},1}, \quad J_{yz\mathrm{R},1}, \quad ms_{x\mathrm{R},1}, \quad ms_{\mathrm{R}}, \quad m_{\mathrm{R}}\right]^{\top} \ . \tag{3.68}$$

Consequently, this mixed form for the parameter vector leads the regressor to be

$$\boldsymbol{W}_k^{\mathrm{ext}} = \left[\boldsymbol{W}_{J,k}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}), \quad \boldsymbol{W}_{\mathrm{w},k}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\right] \quad , \tag{3.69}$$

where:

- $\boldsymbol{W}_J \in \mathbb{R}^{6\times 2}$ is defined in (3.64);
- $\boldsymbol{W}_{\mathrm{w}} \in \mathbb{R}^{6\times 5}$ is composed from the columns of the matrix $\boldsymbol{W}^f(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$, that multiply the parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$, s.t.

$$\boldsymbol{W}_{\mathrm{w}} = \begin{bmatrix} W^{f,1} & W^{f,2} & W^{f,4} & W^{f,5} & W^{f,6} \end{bmatrix} \quad , \tag{3.70}$$

and this represents the missing contribution of the non identified parameters in the formulation of $\boldsymbol{f}^{\mathrm{w}}$. Consequently, collecting N stacked models (3.67), through an LS estimator (3.60) we obtain

$$\hat{\boldsymbol{\kappa}}^{\mathrm{ext}}(\tilde{\boldsymbol{\varepsilon}}) = \left(\boldsymbol{W}^{\mathrm{ext}\top}\boldsymbol{R}^{-1}\boldsymbol{W}^{\mathrm{ext}}\right)^{-1}\boldsymbol{W}^{\mathrm{ext}\top}\boldsymbol{R}^{-1}\tilde{\boldsymbol{\varepsilon}} \quad , \tag{3.71}$$

which represents the estimate of the parameters in $\boldsymbol{\kappa}^{\mathrm{ext}}$. Assuming $\boldsymbol{R} = \mathbf{I}_{3\times 3}$, the combined missing contribution can now be compensated following the block scheme in Fig. 3.5, obtaining

$$\bar{\boldsymbol{f}}^{\mathrm{w}} = \boldsymbol{f}^{\hat{\mathrm{w}}*} + \boldsymbol{\varepsilon}_1(\hat{\boldsymbol{\kappa}}^{\mathrm{J}}) + \boldsymbol{\varepsilon}_2(\hat{\boldsymbol{\kappa}}^{\mathrm{w}}) \quad . \tag{3.72}$$

3.6 Adept Lynx Dynamics

Although controlling the Adept Lynx platform is not a goal of this work, in this section we want to show how to derive a dynamic model for a differentially driven platform. The dynamics of this kind of platforms is highly nonlinear, and include non-holonomic constraints mentioned in Section 2.2.3, which make difficult their modeling and analysis. First the Lagrangian formulation is presented, next the Newton-Euler approach is proposed in the case that we want to integrate the platform with an external manipulator, in order to involve its effects into the platform dynamics.

3.7 Lagrangian Approach

However, is common practice in the literature, to introduce the motion constraint into the dynamic equations, using the additional Lagrange multipliers; on this purpose, we follow the approach presented in [14]. Recalling that a vector of generalized coordinates $\boldsymbol{w} = [x_{\mathrm{F}}, \ y_{\mathrm{F}}, \ \varphi, \ \vartheta_{\mathrm{l}}, \ \vartheta_{\mathrm{r}}]^\top$ has been introduced in (2.22) to identify the platform pose, now we want to introduce another vector

$$\boldsymbol{q}_{\mathrm{plat}} = [q_1, \ q_2]^\top \quad , \tag{3.73}$$

that is the *joint space vector*, composed by the minimal set of variables that let the system be described, which now considers as q_1 and q_2, respectively the previous ϑ_{l} and ϑ_{r}, that are the left and right wheel angles. At this point assuming that mass and moment of inertia of both, casters and the driving wheels, are negligible, we can easily employ the Lagrangian formulation, writing down the Lagrange equation s.t.

$$L = \frac{1}{2} m_{\mathrm{p}} (\dot{x}_{\mathrm{G}}^2 + \dot{y}_{\mathrm{G}}^2) + \frac{1}{2} J_{\mathrm{p}} \dot{\varphi}^2 \ . \tag{3.74}$$

Here we define with m_{p} the platform mass and with J_{p} the platform moment of inertia w.r.t. the axis perpendicular to the ground, that intersects the platform center of mass. Next we derive the cartesian velocities of the center of mass $\dot{x}_{\mathrm{G}}$ and $\dot{y}_{\mathrm{G}}$, by time differentiation from the following geometric relationships, referred to Fig. 2.4(b)

$$x_{\mathrm{G}} = x_{\mathrm{F}} - l_{\mathrm{F}} \cdot \cos(\varphi) \tag{3.75}$$

$$y_{\mathrm{G}} = y_{\mathrm{F}} - l_{\mathrm{F}} \cdot \sin(\varphi) \ . \tag{3.76}$$

Once we substitute the derived $\dot{x}_{\mathrm{G}}$ and $\dot{y}_{\mathrm{G}}$ into (3.74), we obtain an expression of type $L(\dot{\boldsymbol{w}})$ function of the generalized coordinates. Therefore, we are now able to compute the constrained dynamics through

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{w}}} - \frac{\partial L}{\partial \boldsymbol{w}} + \boldsymbol{\Lambda}^{\top}(\boldsymbol{w})\lambda - \boldsymbol{\psi} = 0 \ , \tag{3.77}$$

where λ is the Lagrange multiplier that introduces the non-holonomic constraint defined in (2.47) and $\boldsymbol{\psi}$ is the external force. Observe that here, the constraint forces are responsible for not allowing the wheels to slip sideways. Hence from (3.77), the motion equations can be obtained in the form

$$\boldsymbol{M}(\boldsymbol{w})\ddot{\boldsymbol{w}} + \boldsymbol{V}(\boldsymbol{w}, \dot{\boldsymbol{w}}) = \boldsymbol{E}(\boldsymbol{w})\boldsymbol{\tau} - \boldsymbol{\Lambda}^{\top}(\boldsymbol{w})\lambda \ , \tag{3.78}$$

where $\boldsymbol{M}(\boldsymbol{w}) \in \mathbb{R}^{5\times 5}$ is the inertia matrix, $\boldsymbol{V}(\boldsymbol{w}) \in \mathbb{R}^5$ is the vector of velocity dependent forces, $\boldsymbol{\tau} \in \mathbb{R}^2$ is the torque vector, $\boldsymbol{E}(\boldsymbol{w}) \in \mathbb{R}^{5\times 2}$ is the input transformation matrix. Observe that from (2.34), it holds

$$\dot{\boldsymbol{w}} = \boldsymbol{\xi}(\boldsymbol{w})\dot{\boldsymbol{q}}_{\mathrm{plat}}(t) \ , \tag{3.79}$$

therefore from (3.79) we can obtain an expression for $\ddot{\boldsymbol{w}}$ that can be substituted in (3.78). Then, premultiplicating it by $\boldsymbol{\xi}^{\top}$ we can obtain the equivalent motion equations model

$$\boldsymbol{\xi}^{\top} \left(\boldsymbol{M}\boldsymbol{\xi}\ddot{\boldsymbol{q}}_{\mathrm{plat}}(t) + \boldsymbol{M}\dot{\boldsymbol{\xi}}\dot{\boldsymbol{q}}_{\mathrm{plat}}(t) + \boldsymbol{V} \right) = \boldsymbol{\xi}^{\top}\boldsymbol{E}\boldsymbol{\tau} - \boldsymbol{\xi}^{\top}\boldsymbol{\Lambda}^{\top}\lambda \ , \tag{3.80}$$

where now, exploiting the property in Eq. (2.48), the constraint term disappears, and the constrained dynamics can be reformulated as

$$\boldsymbol{M}^{\star}\ddot{\boldsymbol{q}}_{\mathrm{plat}} + \boldsymbol{V}^{\star} = \boldsymbol{E}^{\star}\boldsymbol{\tau} \ . \tag{3.81}$$

These new quantities are such that $\boldsymbol{M}^{\star} = \boldsymbol{\xi}^{\top}\boldsymbol{M}\boldsymbol{\xi} \in \mathbb{R}^{2\times 2}$ is the reduced mass matrix, always symmetric positively definite since $\boldsymbol{\xi}$ is full rank, $\boldsymbol{V}^{\star} = \boldsymbol{\xi}^{\top}\left(\boldsymbol{M}\dot{\boldsymbol{\xi}}\dot{\boldsymbol{q}}_{\mathrm{plat}} + \boldsymbol{V} \right) \in \mathbb{R}^2$, and $\boldsymbol{E}^{\star} = \boldsymbol{\xi}^{\top}\boldsymbol{E} = \boldsymbol{I}_{2\times 2}$ since the torque vector in (3.78) and (3.81) is kept with the same dimensions. Letting γ and δ be

$$\gamma = \frac{r^2}{4b^2} \left(-4m_{\mathrm{p}} l_{\mathrm{F}}^2 - 8 l_{\mathrm{CM}} m_{\mathrm{p}} l_{\mathrm{F}} + m_{\mathrm{p}} b^2 + 4J_{\mathrm{p}} \right) \tag{3.82}$$

$$\delta = \frac{r^2}{4b^2} \left(4m_{\mathrm{p}} l_{\mathrm{F}}^2 + 8 l_{\mathrm{CM}} m_{\mathrm{p}} l_{\mathrm{F}} + m_{\mathrm{p}} b^2 - 4J_{\mathrm{p}} \right) \tag{3.83}$$

the computed matrices result

$$\boldsymbol{M}^* = \begin{bmatrix} \gamma & \delta \\ \delta & \gamma \end{bmatrix} \tag{3.84}$$

$$\boldsymbol{V}^* = \begin{bmatrix} \frac{m_\mathrm{p} r \dot{\varphi}}{2b} \left[l_\mathrm{F} r (\dot{q}_1 - \dot{q}_2) + l_\mathrm{CM} r (\dot{q}_1 + \dot{q}_2) + (l_\mathrm{F} + l_\mathrm{CM}) b \dot{\varphi} \right] \\ \frac{m_\mathrm{p} r \dot{\varphi}}{2b} \left[l_\mathrm{F} r (\dot{q}_1 - \dot{q}_2) - l_\mathrm{CM} r (\dot{q}_1 + \dot{q}_2) + (l_\mathrm{F} + l_\mathrm{CM}) b \dot{\varphi} \right] \end{bmatrix} \quad . \tag{3.85}$$

Finally observe that summing and subtracting the two components of the model in (3.81), we can also rewrite the dynamics equations as function of the linear and the angular velocities of the platform, defined in Eq. (2.25) and (2.26), obtaining

$$m_\mathrm{p} \dot{v}_\mathrm{p} + m_\mathrm{p} l_\mathrm{CM} \dot{\varphi}^2 = \frac{(\tau_1 + \tau_2)}{r} \tag{3.86}$$

$$\left(m_\mathrm{p} l_\mathrm{CM}^2 + J_\mathrm{p} \right) \ddot{\varphi} - m_\mathrm{p} v_\mathrm{p} \dot{\varphi} l_\mathrm{CM} = \frac{(\tau_2 - \tau_1)\, b}{2r} \quad . \tag{3.87}$$

3.7.1 Payload Integration

In the previous section we have seen that applying the Lagrangian method we can derive the dynamics for a differentially driven mobile platform, in terms of wheel actuator torques. This formulation provides a systematic approach which is also helpful whether we want to integrate a payload on the top of the platform. This is the case taken into account from Papadopoulos [16], where applying the same method we can obtain an unified coupled dynamic model, which also considers the manipulator dynamic effects. In this case, the problem needs to be rearranged such that the joint space vector and the vector of generalized coordinates, are extended to contain also the manipulator joint angles, therefore $\boldsymbol{\Lambda}$ becomes

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = [\sin\varphi \quad -\cos\varphi \ \ l_\mathrm{F} \ \ 0 \ \ 0]^\top \quad , \tag{3.88}$$

and the Lagrangian equation in (3.74) needs to be extended to contain also the kinetic and potential energy of the manipulator links.

At this point we highlight the fact that, in order to follow this approach, we need to have the values of the inertia of the manipulator links, that might be unknown. If this happens, the Lagrangian formulation is not applicable, therefore an alternative method is proposed afterwards, to include the manipulator dynamic effects in the dynamic model of the platform, based on the Newton-Euler approach.

3.8 Newton-Euler Based Approach

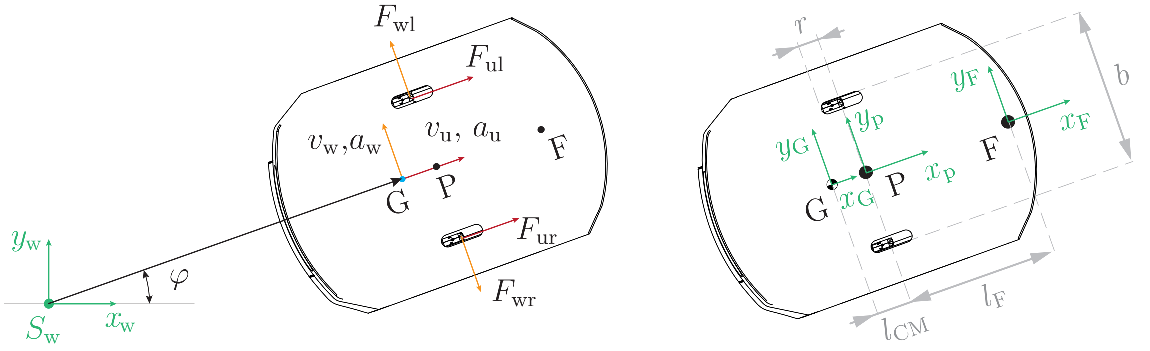
Let us assume to have a manipulator attached on the platform point F, whose interaction with the platform is modeled through ${}^\mathrm{F}\boldsymbol{f} \in \mathbb{R}^6$, wrench exerted on F given w.r.t. frame S_F in Fig. 3.6(b). Here, we first transform this wrench w.r.t. point G, through the following transformation

$${}^\mathrm{G}\boldsymbol{f} = {}^\mathrm{G}\boldsymbol{X}_\mathrm{F}\,{}^\mathrm{F}\boldsymbol{f} \quad . \tag{3.89}$$

Now, once we define the components of the new wrench vector as

$$ {}^{\mathrm{G}}\boldsymbol{f} = \left[{}^{\mathrm{G}}\mathfrak{f}_x,\ {}^{\mathrm{G}}\mathfrak{f}_y,\ {}^{\mathrm{G}}\mathfrak{f}_z,\ {}^{\mathrm{G}}\mu_x,\ {}^{\mathrm{G}}\mu_y,\ {}^{\mathrm{G}}\mu_z \right]^{\top} \ , \tag{3.90} $$

we can apply the Newton Euler formulation, that is based on the application of the Newton and Euler equations of the dynamics, where now, the underlying idea is to involve also the components of the external wrench exerted.



(a) Platform with model parameters.

(b) Platform with frames and lengths.

Figure 3.6: Lynx platform parameters and frames for Newtonian dynamic modeling.

In Fig. 3.6(a), we denote as F_{ul} and F_{ur} the longitudinal force exerted on the vehicle, by the left and right wheel respectively, and as F_{wl} and F_{wr} the exerted lateral forces. Furthermore referring to Fig. 3.6(b), we set $l_{\mathrm{G}} = l_{\mathrm{CM}} + l_{\mathrm{F}}$ and we assume that the masses of all the wheels are negligible. Now, taking J_{p} the platform moment of inertia, w.r.t. the axis perpendicular to the ground that intersects the platform center of mass, the Newton-Euler equations result

$$ m_{\mathrm{p}} a_{\mathrm{u}} = F_{\mathrm{ul}} + F_{\mathrm{ur}} + {}^{\mathrm{G}}\mathfrak{f}_x \tag{3.91} $$

$$ m_{\mathrm{p}} a_{\mathrm{w}} = F_{\mathrm{wl}} - F_{\mathrm{wr}} + {}^{\mathrm{G}}\mathfrak{f} \tag{3.92} $$

$$ J_{\mathrm{p}} \ddot{\varphi} = (F_{\mathrm{ur}} - F_{\mathrm{ul}})\frac{b}{2} + (F_{\mathrm{wl}} - F_{\mathrm{wr}}) l_{\mathrm{CM}} + {}^{\mathrm{G}}\mathfrak{f}_y \cdot l_{\mathrm{G}} + {}^{\mathrm{G}}\mu_z \ , \tag{3.93} $$

referring to a rotation about point G. Observe that the components of the external wrench applied in F have been taken into account through a force-torque transformation, that transforms the wrench from frame S_{F} to frame S_{G}. Now, considering the kinematic relationships that have been derived in Section 2.2.4 through polar coordinates, we substitute the (2.56) (2.57) in the previous equations, obtaining

$$ \dot{v}_{\mathrm{u}} = \frac{F_{\mathrm{ul}} + F_{\mathrm{ur}} + {}^{\mathrm{G}}\mathfrak{f}_x}{m_{\mathrm{p}}} + v_{\mathrm{w}}\dot{\varphi} \tag{3.94} $$

$$ \dot{v}_{\mathrm{w}} = \frac{F_{\mathrm{wl}} - F_{\mathrm{wr}} + {}^{\mathrm{G}}\mathfrak{f}_y}{m_{\mathrm{p}}} - v_{\mathrm{u}}\dot{\varphi} \tag{3.95} $$

$$ \ddot{\varphi} = \frac{1}{J_{\mathrm{p}}}\left[(F_{\mathrm{ur}} - F_{\mathrm{ul}})\frac{b}{2} + (F_{\mathrm{wl}} - F_{\mathrm{wr}}) l_{\mathrm{CM}} + {}^{\mathrm{G}}\mathfrak{f}_y \cdot l_{\mathrm{G}} + {}^{\mathrm{G}}\mu_z \right] \ , \tag{3.96} $$

where we recall that all the kinematic quantities refer to Fig.3.6(a). Therefore, we now include the non-holonomic constraint by substituting (2.58) in (3.94) and (3.95), and combining with (3.96), we obtain

$$\dot{v}_{\mathrm{u}} = \frac{F_{\mathrm{ul}} + F_{\mathrm{ur}} + {}^{\mathrm{G}}\mathfrak{f}_x}{m_{\mathrm{p}}} - l_{\mathrm{CM}}\dot{\varphi}^2 \tag{3.97}$$

$$\ddot{\varphi} = \frac{1}{J_{\mathrm{p}} + m_{\mathrm{p}} l_{\mathrm{CM}}^2}\left[(F_{\mathrm{ur}} - F_{\mathrm{ul}})\frac{b}{2} + {}^{\mathrm{G}}\mathfrak{f}_y \cdot l_{\mathrm{F}} + {}^{\mathrm{G}}\mu_z + l_{\mathrm{CM}} m_{\mathrm{p}} v_{\mathrm{u}} \dot{\varphi}\right] \ , \tag{3.98}$$

that is the constrained platform dynamics, described with the longitudinal and angular acceleration of the platform. At this point, exploiting the physical relationship between longitudinal force exerted by the wheel and torque applied to it, from (3.97) and (3.98), we can write the dynamics showing the wheel actuator torques that results

$$m_{\mathrm{p}} a_{\mathrm{u}} + m_{\mathrm{p}} l_{\mathrm{CM}} \dot{\varphi}^2 - {}^{\mathrm{G}}\mathfrak{f}_x = \frac{(\tau_{\mathrm{l}} + \tau_{\mathrm{r}})}{r} \tag{3.99}$$

$$\left(m_{\mathrm{p}} l_{\mathrm{CM}}^2 + J_{\mathrm{p}}\right)\ddot{\varphi} - m_{\mathrm{p}} v_{\mathrm{u}} \dot{\varphi} l_{\mathrm{CM}} - {}^{\mathrm{G}}\mu_z - {}^{\mathrm{G}}\mathfrak{f}_y \cdot l_{\mathrm{F}} = \frac{(\tau_{\mathrm{r}} - \tau_{\mathrm{l}})\, b}{2r} \ . \tag{3.100}$$

Finally, we have obtained a dynamic model for the platform, where the dynamic contribution of the manipulator has been considered, treated as external force.

Observe that whether we assume the wrench components to be zero, the Newton-Euler based model in (3.99) and (3.100), is equivalent to the Lagrangian one, obtained in (3.86) and (3.87) without the assumption of external payload.

Table 3.3: Base inertial parameter of Roberta P80.800, from [4].

Base Parameter $\boldsymbol{\kappa}$	**Standard parameter $\boldsymbol{\kappa}_{\mathrm{std}}$**
$J_{zz\mathrm{R},1}$	$J_{\mathrm{d},1} + J_{yy,2} + J_{yy,3} + J_{zz,1} + d_3^2\,(m_3 + m_4 + m_5 + m_6)$
$J_{xx\mathrm{R},2}$	$J_{xx,2} - J_{yy,2} - d_3^2\,(m_3 + m_4 + m_5 + m_6)$
$J_{xy\mathrm{R},2}$	$J_{xy,2}$
$J_{xz\mathrm{R},2}$	$J_{xz,2} - ms_{z,3}\,d_3$
$J_{yz\mathrm{R},2}$	$J_{yz,2}$
$J_{zz\mathrm{R},2}$	$J_{\mathrm{d},2} + J_{zz,2} + d_3^2\,(m_3 + m_4 + m_5 + m_6)$
$ms_{x\mathrm{R},2}$	$ms_{x,2} + d_3\,(m_3 + m_4 + m_5 + m_6)$
$ms_{y\mathrm{R},2}$	$ms_{y,2}$
$J_{xx\mathrm{R},3}$	$J_{xx,3} - J_{yy,3} + J_{yy,4} + 2\,ms_{z,4}\,r_4 + r_4^2\,(m_4 + m_5 + m_6)$
$J_{xy\mathrm{R},3}$	$J_{xy,3}$
$J_{xz\mathrm{R},3}$	$J_{xz,3}$
$J_{yz\mathrm{R},3}$	$J_{yz,3}$
$J_{zz\mathrm{R},3}$	$J_{yy,4} + J_{zz,3} + 2\,ms_{z,4}\,r_4 + r_4^2\,(m_4 + m_5 + m_6)$
$ms_{x\mathrm{R},3}$	$ms_{x,3}$
$ms_{y\mathrm{R},3}$	$ms_{y,3} - ms_{z,4} - r_4\,(m_4 + m_5 + m_6)$
$J_{\mathrm{dR},3}$	$J_{\mathrm{d},3}$
$J_{xx\mathrm{R},4}$	$J_{xx,4} - J_{yy,4} + J_{yy,5}$
$J_{xy\mathrm{R},4}$	$J_{xy,4}$
$J_{xz\mathrm{R},4}$	$J_{xz,4}$
$J_{yz\mathrm{R},4}$	$J_{yz,4}$
$J_{zz\mathrm{R},4}$	$J_{yy,5} + J_{zz,4}$
$ms_{x\mathrm{R},4}$	$ms_{x,4}$
$ms_{y\mathrm{R},4}$	$ms_{y,4} - ms_{z,5}$
$J_{\mathrm{dR},4}$	$J_{\mathrm{d},4}$
$J_{xx\mathrm{R},5}$	$J_{xx,5} - J_{yy,5} + J_{yy,6}$
$J_{xy\mathrm{R},5}$	$J_{xy,5}$
$J_{xz\mathrm{R},5}$	$J_{xz,5}$
$J_{yz\mathrm{R},5}$	$J_{yz,5}$
$J_{zz\mathrm{R},5}$	$J_{yy,6} + J_{zz,5}$
$ms_{x\mathrm{R},5}$	$ms_{x,5}$
$ms_{y\mathrm{R},5}$	$ms_{y,5} - ms_{z,6}$
$J_{\mathrm{dR},5}$	$J_{\mathrm{d},5}$
$J_{xx\mathrm{R},6}$	$J_{xx,6} - J_{yy,6}$
$J_{xy\mathrm{R},6}$	$J_{xy,6}$
$J_{xz\mathrm{R},6}$	$J_{xz,6}$
$J_{yz\mathrm{R},6}$	$J_{yz,6}$
$J_{zz\mathrm{R},6}$	$J_{zz,6}$
$ms_{x\mathrm{R},6}$	$ms_{x,6}$
$ms_{y\mathrm{R},6}$	$ms_{y,6}$
$J_{\mathrm{dR},6}$	$J_{\mathrm{d},6}$

Table 3.4: Wrench inertial parameter of Roberta P80.800

Entry	**Wrench Parameter $\boldsymbol{\kappa}^f$**	**Standard parameter $\boldsymbol{\kappa}_{\text{std}}$**
1	$J_{xz\mathrm{R},1}$	$J_{xz,1}$
2	$J_{yz\mathrm{R},1}$	$J_{yz,1}$
3	$J_{zz\mathrm{R},1}$	$J_{\mathrm{d},1}+J_{yy,2}+J_{yy,3}+J_{zz,1}+d_3^2\,(m_3+m_4+m_5+m_6)$
4	$ms_{x\mathrm{R},1}$	$ms_{x,1}$
5	ms_{R}	$ms_{y,1}-ms_{z,2}-ms_{z,3}$
6	m_{R}	$m_1+m_2+m_3+m_4+m_5+m_6$
7	$J_{xx\mathrm{R},2}$	$J_{xx,2}-J_{yy,2}-d_3^2\,(m_3+m_4+m_5+m_6)$
8	$J_{xy\mathrm{R},2}$	$J_{xy,2}$
9	$J_{xz\mathrm{R},2}$	$J_{xz,2}-ms_{z,3}\,d_3$
10	$J_{yz\mathrm{R},2}$	$J_{yz,2}$
11	$J_{zz\mathrm{R},2}$	$J_{\mathrm{d},2}+J_{zz,2}+d_3^2\,(m_3+m_4+m_5+m_6)$
12	$ms_{x\mathrm{R},2}$	$ms_{x,2}+d_3\,(m_3+m_4+m_5+m_6)$
13	$ms_{y\mathrm{R},2}$	$ms_{y,2}$
14	$J_{xx\mathrm{R},3}$	$J_{xx,3}-J_{yy,3}+J_{yy,4}+2\,ms_{z,4}\,r_4+r_4^2\,(m_4+m_5+m_6)$
15	$J_{xy\mathrm{R},3}$	$J_{xy,3}$
16	$J_{xz\mathrm{R},3}$	$J_{xz,3}$
17	$J_{yz\mathrm{R},3}$	$J_{yz,3}$
18	$J_{zz\mathrm{R},3}$	$J_{yy,4}+J_{zz,3}+2\,ms_{z,4}\,r_4+r_4^2\,(m_4+m_5+m_6)$
19	$ms_{x\mathrm{R},3}$	$ms_{x,3}$
20	$ms_{y\mathrm{R},3}$	$ms_{y,3}-ms_{z,4}-r_4\,(m_4+m_5+m_6)$
21	$J_{xx\mathrm{R},4}$	$J_{xx,4}-J_{yy,4}+J_{yy,5}$
22	$J_{xy\mathrm{R},4}$	$J_{xy,4}$
23	$J_{xz\mathrm{R},4}$	$J_{xz,4}$
24	$J_{yz\mathrm{R},4}$	$J_{yz,4}$
25	$J_{zz\mathrm{R},4}$	$J_{yy,5}+J_{zz,4}$
26	$ms_{x\mathrm{R},4}$	$ms_{x,4}$
27	$ms_{y\mathrm{R},4}$	$ms_{y,4}-ms_{z,5}$
28	$J_{xx\mathrm{R},5}$	$J_{xx,5}-J_{yy,5}+J_{yy,6}$
29	$J_{xy\mathrm{R},5}$	$J_{xy,5}$
30	$J_{xz\mathrm{R},5}$	$J_{xz,5}$
31	$J_{yz\mathrm{R},5}$	$J_{yz,5}$
32	$J_{zz\mathrm{R},5}$	$J_{yy,6}+J_{zz,5}$
33	$ms_{x\mathrm{R},5}$	$ms_{x,5}$
34	$ms_{y\mathrm{R},5}$	$ms_{y,5}-ms_{z,6}$
35	$J_{xx\mathrm{R},6}$	$J_{xx,6}-J_{yy,6}$
36	$J_{xy\mathrm{R},6}$	$J_{xy,6}$
37	$J_{xz\mathrm{R},6}$	$J_{xz,6}$
38	$J_{yz\mathrm{R},6}$	$J_{yz,6}$
39	$J_{zz\mathrm{R},6}$	$J_{zz,6}$
40	$ms_{x\mathrm{R},6}$	$ms_{x,6}$
41	$ms_{y\mathrm{R},6}$	$ms_{y,6}$

4 Base Wrench Validation

In this chapter we are going to show the results obtained following the base wrench adjustment steps, explained in Section 3.5. We recall that, since the real formulation in (3.37) is not numerically computable, the only way to approximate the base wrench is to adapt the base parameters that have been identified for the inverse dynamics, as explained in Section 3.3. For this reason, several formulations of the base wrench are simulated and compared with the measurements coming from the readings of a 6-axes force-torque sensor, placed underneath the Roberta P80.800 base.

The validation idea is to let the Roberta P80.800 perform a specific movement, by providing a desired velocity profile to its end-effector, and then collect N base wrench measurements $\tilde{\boldsymbol{f}}^{\mathrm{w}}$. Therefore N simulations $\hat{\boldsymbol{f}}^{\mathrm{w}}$ of the base wrench are performed on the same velocity profile, and they are compared with the N real recorded measurements.

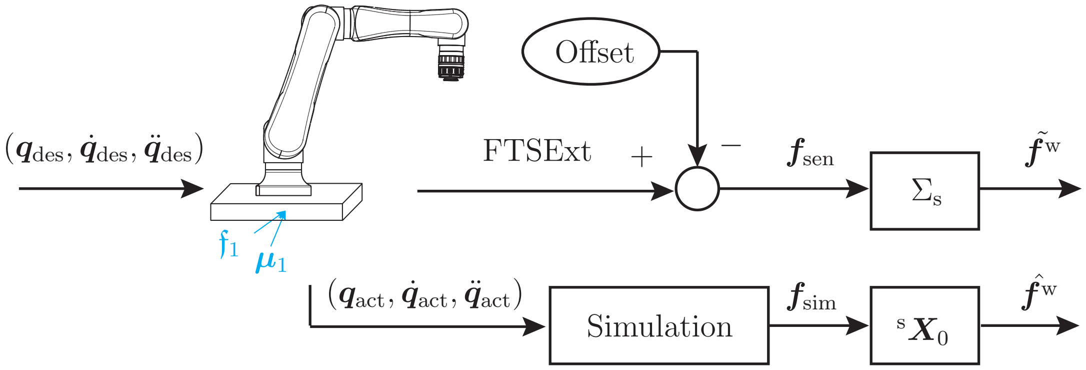

Offset
$(\boldsymbol{q}_{\mathrm{des}}, \dot{\boldsymbol{q}}_{\mathrm{des}}, \ddot{\boldsymbol{q}}_{\mathrm{des}})$
FTSExt
$+$
$-$
$\boldsymbol{f}_{\mathrm{sen}}$
Σ_{s}
$\tilde{\boldsymbol{f}}^{\mathrm{w}}$
$\mathfrak{f}_1$
$\boldsymbol{\mu}_1$
$(\boldsymbol{q}_{\mathrm{act}}, \dot{\boldsymbol{q}}_{\mathrm{act}}, \ddot{\boldsymbol{q}}_{\mathrm{act}})$
Simulation
$\boldsymbol{f}_{\mathrm{sim}}$
${}^{\mathrm{s}}\boldsymbol{X}_0$
$\hat{\boldsymbol{f}}^{\mathrm{w}}$

Figure 4.1: Base wrench validation scheme.

4.1 Preliminary Transformations

In order to be compared, the sensor measurements as well as the simulations need a preliminary transformation, to refer to an unified reference frame. Fig. 4.1 shows the base wrench validation scheme adopted.

First of all, since the force-torque sensor is affected by an instrumental offset that disturbs the sensor outputs, N force-torque measurements are collected with the sensor unloaded, without the Roberta on it. Therefore, we denote with Offset the mean value among the N recorded offset measurements, and this value has to be subtracted to every force-torque measurement.

At this point, a desired trajectory is given to the Roberta P80.800, whose end-effector movements affect the force-torque sensor at its bottom. Then, the sensor data FTSExt can be collected and set free of the Offset, so we now obtain a base wrench measurement

$\boldsymbol{f}_{\text{sen}} = [\mathfrak{f}_{\text{sen}}, \boldsymbol{\mu}_{\text{sen}}]^\top$ that refer to the frame S_{s}, as Fig. 4.2 shows. On the other side, the simulations are run on the actual trajectory, which is the real joint trajectory performed by the Roberta, giving an expression of $\boldsymbol{f}_{\text{sim}} = [\mathfrak{f}_{\text{sim}}, \boldsymbol{\mu}_{\text{sim}}]^\top$ that refer to the frame S_0. According with Fig. 4.1, two more steps are needed to compare these quantities.

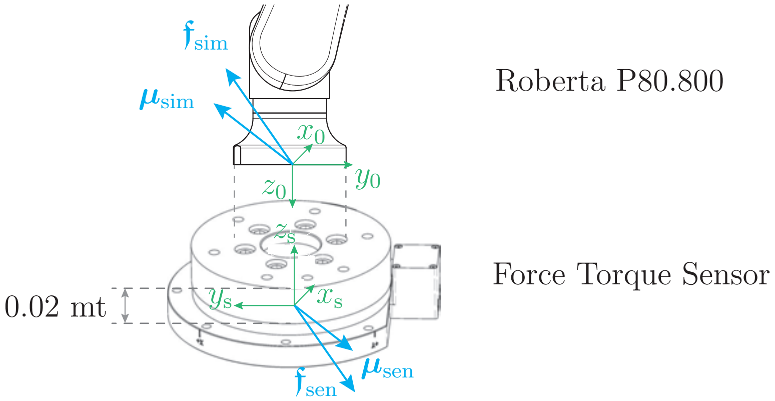

$\mathfrak{f}_{\text{sim}}$
$\boldsymbol{\mu}_{\text{sim}}$
Roberta P80.800
x_0
y_0
z_0
z_{s}
Force Torque Sensor
0.02 mt
y_{s}
x_{s}
$\boldsymbol{\mu}_{\text{sen}}$
$\mathfrak{f}_{\text{sen}}$

Figure 4.2: Base wrench simulations and measurements, w.r.t. their reference frame.

- Σ_{s} block, provides to flip the sign of the measurement such that

$$\tilde{\boldsymbol{f}}^{\text{w}} = -\boldsymbol{f}_{\text{sen}} \ . \tag{4.1}$$

- Then, a force-torque transformation is applied to move the $\mathfrak{f}_{\text{sim}}$ and $\boldsymbol{\mu}_{\text{sim}}$, simulated force and torque, from S_0 to S_{s}, that results

$$\hat{\boldsymbol{f}}^{\text{w}} = {}^{\text{s}}\hat{\boldsymbol{f}}^{\text{w}} = {}^{\text{s}}\boldsymbol{X}_0 \boldsymbol{f}_{\text{sim}} \ . \tag{4.2}$$

  where ${}^{\text{s}}\boldsymbol{X}_0$ denotes the *force-torque transformation* from S_0 to S_{s}, which refers that S_0 is rotated of a -π clockwise angle w.r.t x_{s}-axis, and it is traslated through the translation vector ${}^{\text{s}}\boldsymbol{p}_0 = [0, \ 0, \ 0.02\text{m}]^\top$.

After these two steps, the quantities $\tilde{\boldsymbol{f}}^{\text{w}}$ and $\hat{\boldsymbol{f}}^{\text{w}}$ are ready to be compared, as shown in Fig. 4.1 .

4.2 Optimal trajectory

For validating the base wrench simulations as well as the adjustment formulation, an arbitrary desired trajectory for the joints, needs to be given to the Roberta P80.800. Since the adjustment formulation also includes the parameter identification of the unknown $\boldsymbol{\kappa}^{\text{w}}$, we aim to use among all the trajectories, the optimal one that can better excite all the parameters in $\boldsymbol{\kappa}^{f}$, and consequently the ones in the unknown parameter set. This optimal trajectory is periodic, and consists of fourier series signals given to each of the manipulator axes, and it is provided from previous work done in [4] and [5]. Hence, once we identify the parameters and validate the approach using the optimal trajectory, we also consider different trajectories, in order to cross validate the formulation.

4.3 Base Parameter Set Results

First of all, we want to show the results that the formulation $\boldsymbol{W}^f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*}$ gives through $\boldsymbol{\kappa}^{f*}$ parameter set, which means adopting the base wrench parameters of the inverse dynamics, and neglecting the non identified parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$, as well as and the rotors inertia contributions $J_{\mathrm{d},i}$ for $i = 3, 4, 5, 6$.

As a result, this formulation yields a symbolic error computed in (3.61)-(3.62), which is reported in Table 4.1, where, for each error component, the dependencies on the parameters in $\boldsymbol{\kappa}^{\mathrm{w}}$ and on the inertias $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$ are highlighted. Here some initial considerations can be done.

Table 4.1: Symbolic error for every wrench component, computed w.r.t. (3.61)-(3.62). Here the g_is refer to a generic function.

Error Component ε_i	**Symbolic Error**
ε_1	$g_1(ms_{x\mathrm{R},1}, ms_{\mathrm{R}})$
ε_2	$g_2(ms_{x\mathrm{R},1}, ms_{\mathrm{R}})$
ε_3	$-9.81 \cdot m_{\mathrm{R}}$
ε_4	$g_4\left(J_{xz\mathrm{R},1}, J_{yz\mathrm{R},1}, ms_{x\mathrm{R},1}, ms_{\mathrm{R}}\right) - J_{\mathrm{d},2}\left(\ddot{q}_2 \sin q_1 + \dot{q}_1\dot{q}_2 \cos q_1\right)$
ε_5	$g_5\left(J_{xz\mathrm{R},1}, J_{yz\mathrm{R},1}, ms_{x\mathrm{R},1}, ms_{\mathrm{R}}\right) + J_{\mathrm{d},2}\left(\ddot{q}_2 \cos q_1 - \dot{q}_1\dot{q}_2 \sin q_1\right)$
ε_6	$-J_{\mathrm{d},1}\ddot{q}_1$

- Comparing to the force simulations, the moment simulations are more approximated, for the fact that they also include the terms of error coming from the inertias $J_{\mathrm{d},1}$ and $J_{\mathrm{d},2}$.
- The error of the moment components, depends on the inertia through an acceleration term, in a way that it gets higher for high $\ddot{q}_1$ $\ddot{q}_2$ accelerations.
- The magnitude of each error component depends on the magnitude of its dependencies, that are unknown so far.

Therefore, applying the validation scheme in Fig. 4.1, we obtain $\boldsymbol{f}_{\mathrm{sim}}$ through

$$\boldsymbol{f}_{\mathrm{sim}} = \boldsymbol{W}^f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\kappa}^{f*} \quad . \tag{4.3}$$

Hence, by applying the preliminary transformations in Sec. 4.1 , we can obtain the measured $\tilde{\boldsymbol{f}}^{\mathrm{w}}$ and the simulated $\hat{\boldsymbol{f}}^{\mathrm{w}}$, whose forces and moments components are compared respectively in Fig. 4.3 and 4.4.
From the simulation results, we can notice the presence of noise, that highly affect the base wrench measurements. Despite this undesired element, for every component of the base wrench, we can see that the simulations are consistent with the trend of the measurements, in particular from Fig. 4.3 and 4.4 it comes out:

- a big offset in the force along the z-component. Furthermore, taking a look on Tab. 4.1, this error is supposed to be proportional to m_{R}, which is, from Tab. 3.1, the sum of all the link masses. We can conclude that we cannot get rid of this error, as long as this parameter is unknown.

- A relevant error that concerns the measured and simulated moment in the z-component. Here one more time, Tab. 4.1 models this error as a linear dependency on $J_{\mathrm{d},1}$ through the acceleration $\ddot{q}_1$. Similarly, to compensate this error, the rotor inertia parameter needs to be identified.

After this evaluation of the first approach, we can conclude that the parameter set $\boldsymbol{\kappa}^{f*}$, that contains the entries from the original identified set $\boldsymbol{\kappa}$, provides a good base wrench formulation for the applications that do not require accuracy along the z-axis. Otherwise, in order to get a better base wrench formulation, we need to involve a parameter identification procedure.

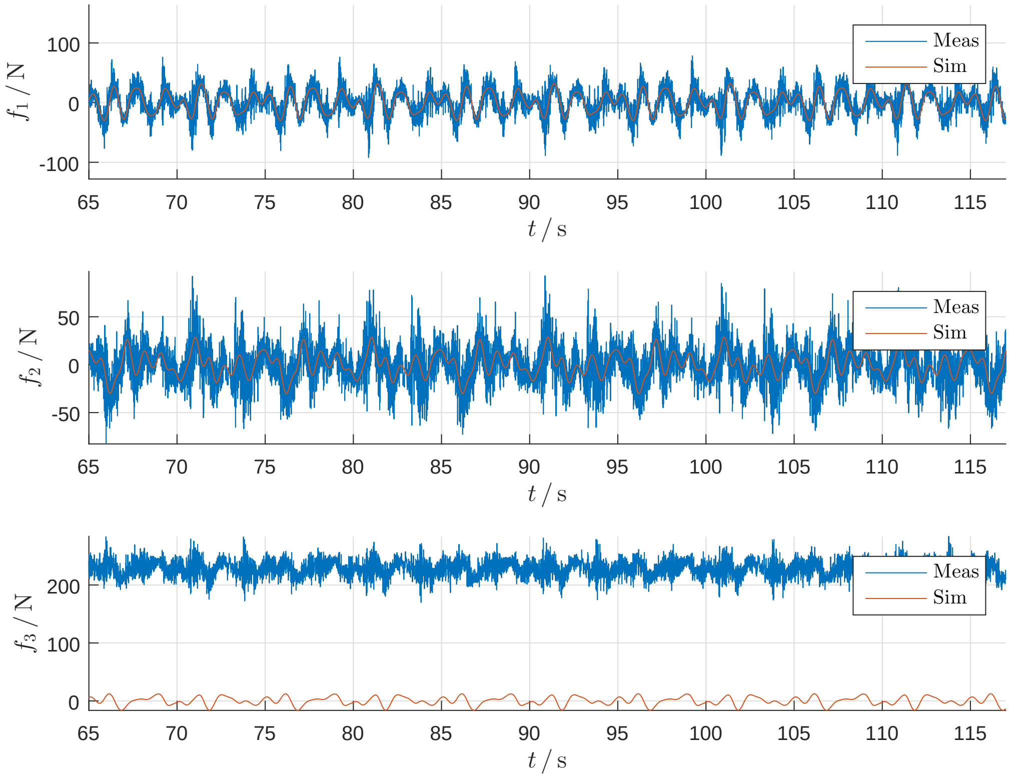


Figure 4.3: Base wrench force components, simulated with known parameters in $\boldsymbol{\kappa}$, are compared with the measurements. The plot shows 5 periods.

4.4 Results with the Systematic Error Compensation

From a symbolic evaluation, as well as from an evaluation of the outcomes provided by the $\boldsymbol{\kappa}^{f*}$ parameter set, a parameter identification is needed to improve the formulation of the base wrench. Similarly in what has been done in the previous section, we aim to apply the validation scheme in Fig. 4.1, taking $\boldsymbol{f}_{\mathrm{sim}}$ as the $\bar{\boldsymbol{f}}^{\mathrm{w}}$ obtained in Section 3.5.1.
As we expected from Tab. 4.1, by compensating the systematic error, we can improve the components of the base wrench moment. Fig. 4.5 shows the outcomes of the systematic error compensation on the moment, and as a result of this, we now have that all the error between measurements and simulations is a consequence of the missing $\boldsymbol{\kappa}^{\mathrm{w}}$ parameter set contribution.

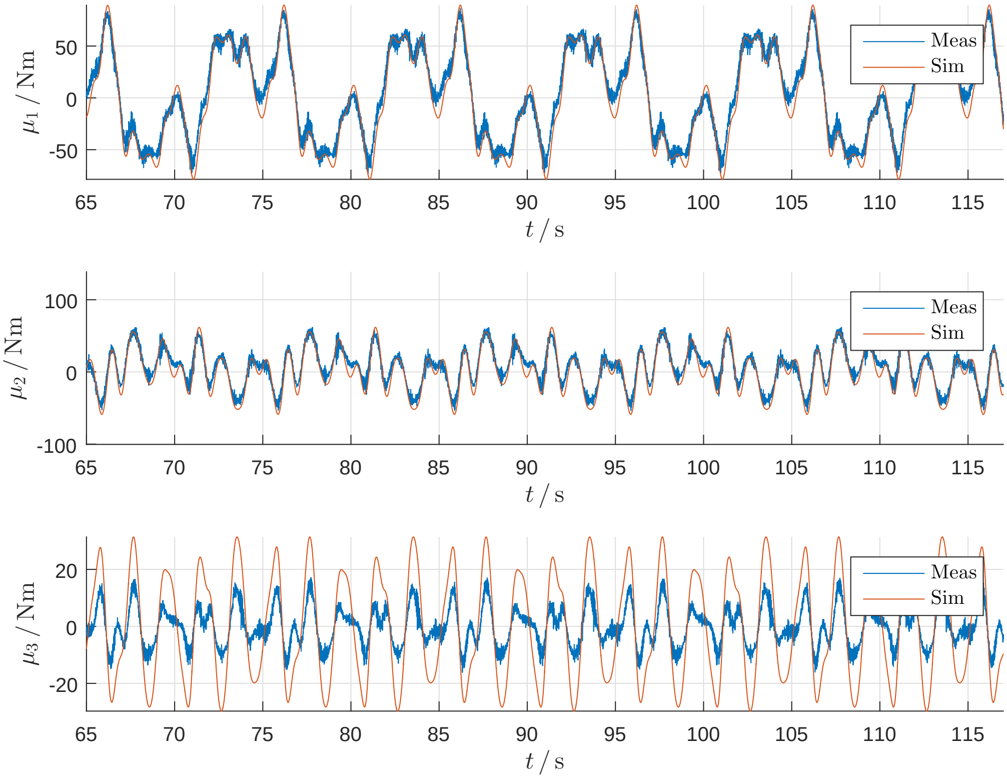


Figure 4.4: Base wrench moment components, simulated with known parameters in $\boldsymbol{\kappa}$, are compared with the measurements. The plot shows 5 periods.

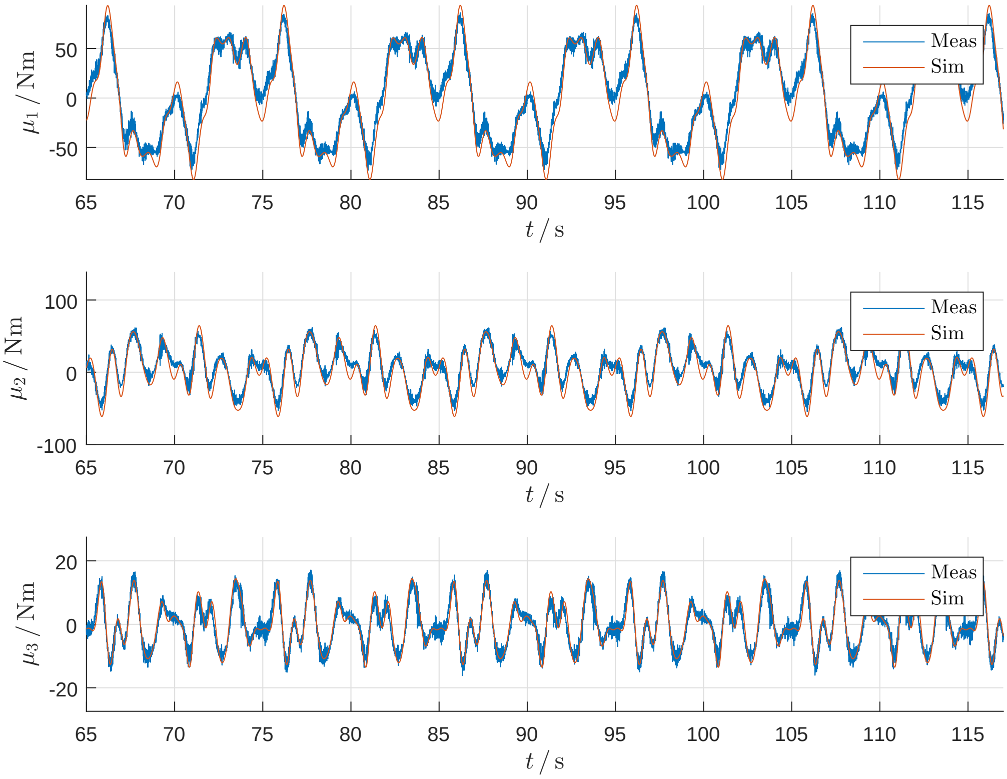


Figure 4.5: Base wrench moment components, simulated with the $\boldsymbol{\varepsilon}_1$ error compensation in (3.65), are compared with the measurements. The plot shows 5 periods.

4.5 Results with the Parameter Identification

As a result of the application of the first systematic error compensation, we now have that all the error between measurements and simulations is supposed to be modeled as a consequence of the $\boldsymbol{\kappa}^{\mathrm{w}}$. Therefore, here we want to model the error as a missing contribution of $\boldsymbol{\kappa}^{\mathrm{w}}$, as well as an over contribution from the inertia $J_{\mathrm{d},1}$ and $J_{\mathrm{d},1}$.
Then, the overall error compensation explained in Section 3.5.2, can be implemented by taking the $\boldsymbol{f}_{\mathrm{sim}}$ in Fig. 4.1 as $\bar{\boldsymbol{f}}^{\mathrm{w}}$ from formula (3.72), that is the outcome of the approach shown in Fig. 3.5. If we stick to this Figure, the block Σ_1 performs the parameter identification from 10 periods of the optimal trajectory taken, and provides the outcomes shown in Tab. 4.2.
Here we observe that the negative value of the parameter $J_{\mathrm{d},2}$, which represents an inertia, does not have a proper physical meaning. In the light of this fact, we point out that this is the outcome of a refitting procedure, that involves the identification of a parameter, whose value has already been identified in a grouped form (see entry 11 of Table 3.4). Therefore, such procedure of adjustment of the base wrench formulation, is convenient in order to rearrange the base wrench simulations, but it might not provide physically reliable outcomes.

Table 4.2: Additional base inertial parameters in $\boldsymbol{\kappa}^{\mathrm{ext}}$.

Wrench Parameter $\boldsymbol{\kappa}^{\mathrm{ext}}$	**Dynamic Parameter**	**Identified Value**
$J_{\mathrm{dR},1}$	$J_{\mathrm{d},1}$	4.063
$J_{\mathrm{dR},2}$	$J_{\mathrm{d},2}$	−1.155
$J_{xz\mathrm{R},1}$	$J_{xz,1}$	1.139
$J_{yz\mathrm{R},1}$	$J_{yz,1}$	0.701
$ms_{x\mathrm{R},1}$	$ms_{x,1}$	−0.588
ms_{R}	$ms_{y,1} - ms_{z,2} - ms_{z,3}$	−0.314
m_{R}	$m_1+m_2+m_3+m_4+m_5+m_6$	23.157

The results of the overall adjustment formulation are shown in in Fig. 4.6 and 4.7. These plots show how the identified wrench parameters can improve the simulations, by comparison with the measurements. Comparing Fig. 4.6 with Fig. 4.3, we can easily see a crucial improvement regarding the third component of the force, thanks to the identification of the wrench parameter m_{R}. Despite the third component of the force, it is quite hard to evaluate the impact of the identified parameters on the simulations. Thus we define the quantities

$$\boldsymbol{\varepsilon}_{in} = \tilde{\boldsymbol{f}}^{\mathrm{w}} - \hat{\boldsymbol{f}}^{\mathrm{w}} = \tilde{\boldsymbol{f}}^{\mathrm{w}} - {}^{\mathrm{s}}\boldsymbol{X}_0 \bar{\boldsymbol{f}}^{\mathrm{w}} \tag{4.4}$$

$$\boldsymbol{\varepsilon}_{fin} = \tilde{\boldsymbol{f}}^{\mathrm{w}} - \hat{\boldsymbol{f}}^{\mathrm{w}} = \tilde{\boldsymbol{f}}^{\mathrm{w}} - {}^{\mathrm{s}}\boldsymbol{X}_0 \hat{\boldsymbol{f}}^{\mathrm{w}*} \ , \tag{4.5}$$

plotted in Fig. 4.8 and 4.9, that are respectively, the initial error between the initial base wrench formulation, derived with the $\boldsymbol{\kappa}$ parameters, and the measurements, and the final error between the compensated formulation and measurements. Now, for each component of the wrench, we perform a numeric evaluation of the error through the $RMSE$ error, explained in (3.47) using the formulas

$$SSE = \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \ , \tag{4.6}$$

$$RMSE = \sqrt{\frac{SSE}{M}} \ . \tag{4.7}$$

Summarizing, through Eq. (4.6)-(4.7) we compute the standard error of the regression, on $\boldsymbol{\varepsilon}_{in}$ and $\boldsymbol{\varepsilon}_{fin}$, obtaining RMSE$_{in}$ and RMSE$_{fin}$.

Table 4.3: RMSE values, for each wrench component, with and without the compensation.

	$\mathfrak{f}_x$	$\mathfrak{f}_y$	$\mathfrak{f}_z$	μ_x	μ_y	μ_z
RMSE$_{in}$	12.237	14.171	227.978	8.603	8.475	10.180
RMSE$_{fin}$	12.379	14.198	10.349	11.682	9.464	1.960

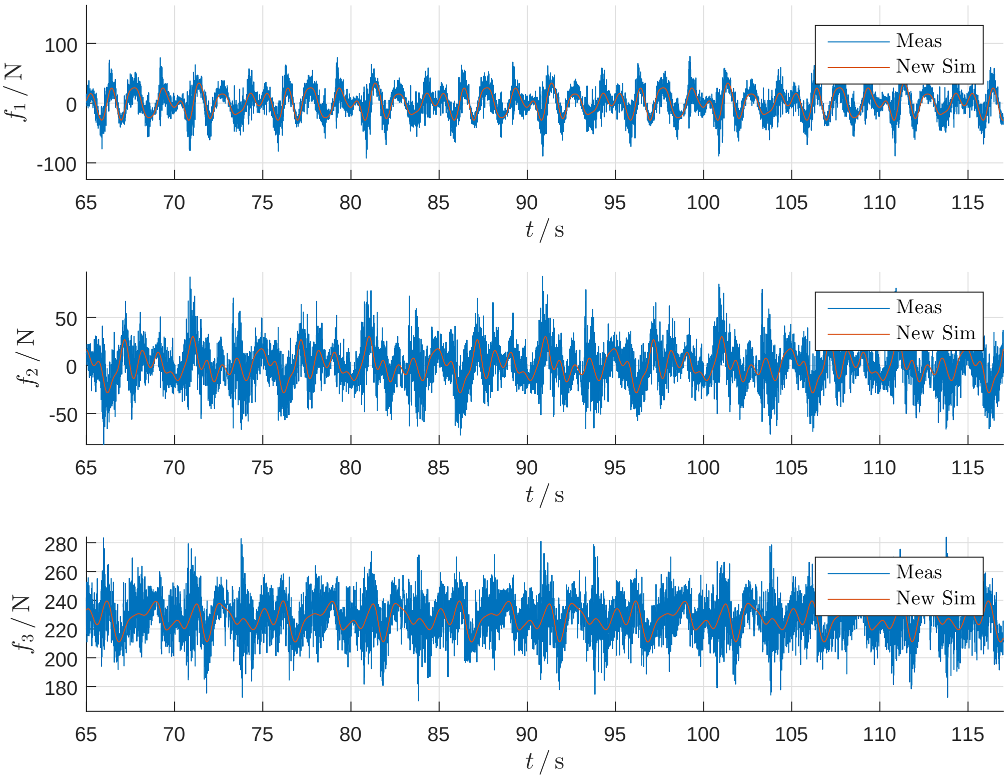


Figure 4.6: Base wrench force components, simulated with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods.

Now comparing the results shown in Table 4.3, the following considerations regarding the validity of the adopted approach, can be done.

- RMSE values show a qualitative measure of the goodness of the results. In fact this factor is highly sensitive to the noise of the measurements, as it can be seen from Section 3.4.
- The main improvements are obtained in the computation of the components along the z-axis. Indeed, the error for these components gets a relevant diminution after the compensation.

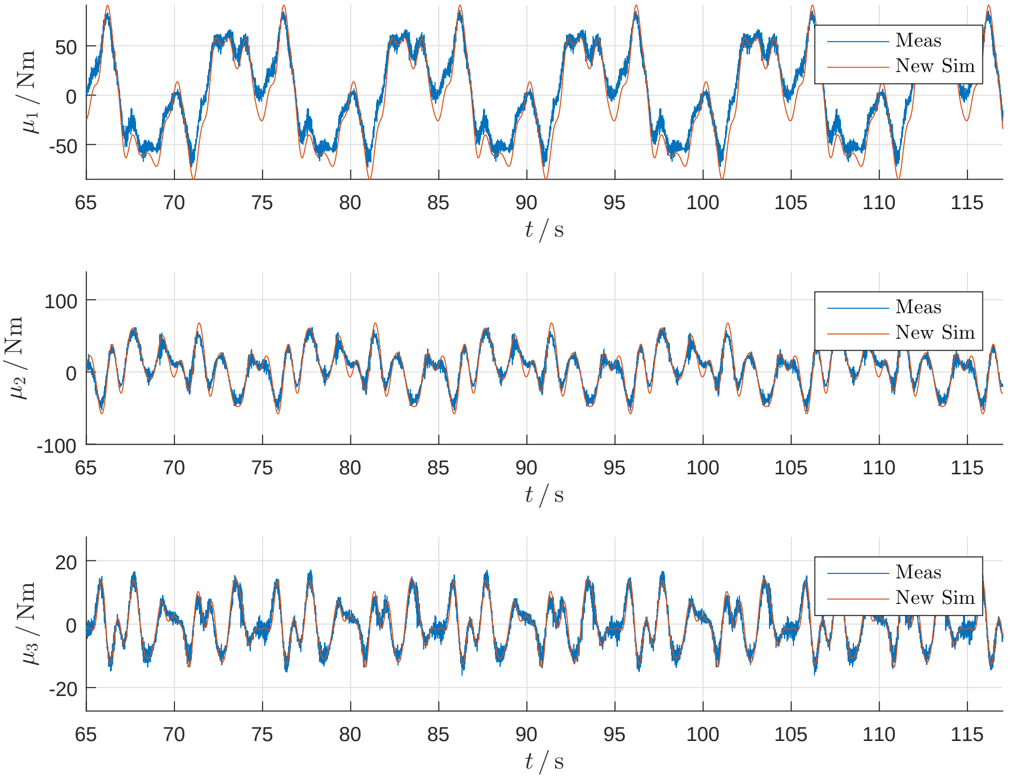


Figure 4.7: Base wrench moment components, simulated with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods.

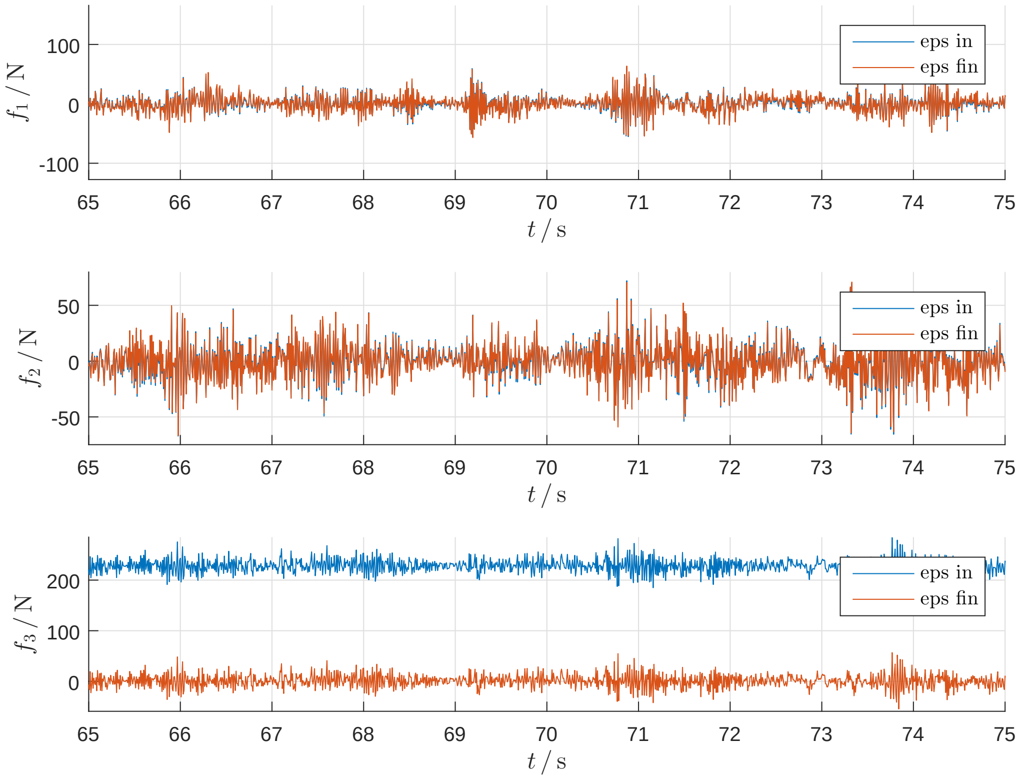


Figure 4.8: Comparison between errors $\boldsymbol{\varepsilon}_{in}$ and $\boldsymbol{\varepsilon}_{fin}$, depicted in half of a period.

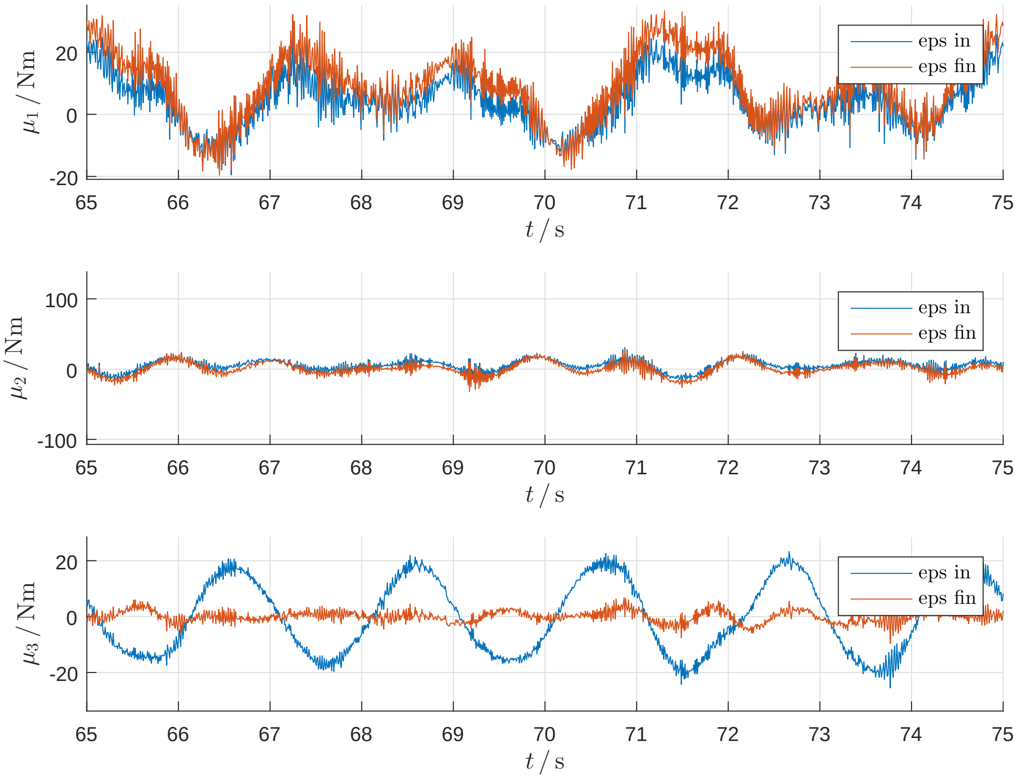


Figure 4.9: Comparison between errors $\boldsymbol{\varepsilon}_{in}$ and $\boldsymbol{\varepsilon}_{fin}$, depicted in half of a period.

- The first two components of force and moment, do not get any improvement, showing that these quantities already have a reliable trend, after the first non compensated base wrench formulation.

4.6 Cross Validation on Different Trajectories

Likewise as it has been done in Section 4.5, where the impact of the identified parameters is evaluated on a specific trajectory, the cross validation can be performed following the same evaluation criteria, by running a different trajectory instead of the optimal one. On this purpose we adopt three trajectories that are:

- *Dynamics excitation trajectory* (dyn) is designed for optimally exciting the inertia parameters. This trajectory is the one adopted by Stapf [4] for the identification of the base inertia parameters $\boldsymbol{\kappa}$.
- *Demo Red Box movement* (red) is part of a demonstration program that has been presented several times with this robot. It consists of linear and point to point movements, and it is an interesting trajectory, because it includes the most dynamic movement that occurs in the real applications.
- *PTP and LIN movement* (PL), let the robot moves repeatedly between three points with varied velocities, where some movements are performed as joint space point to point (PTP) movements and some are linear (LIN) movements in Cartesian space. This trajectory is not really employed in common applications.

Table 4.4: RMSE values for each component of the base wrench, of the three trajectories.

	$\mathfrak{f}_x$	$\mathfrak{f}_y$	$\mathfrak{f}_z$	μ_x	μ_y	μ_z
$\mathrm{RMSE}_{in}^{\mathrm{dyn}}$	14.278	14.186	215.249	7.818	9.016	12.675
$\mathrm{RMSE}_{fin}^{\mathrm{dyn}}$	14.305	14.300	17.415	10.341	11.138	2.207
$\mathrm{RMSE}_{in}^{\mathrm{red}}$	3.073	3.478	219.145	3.560	4.495	1.409
$\mathrm{RMSE}_{fin}^{\mathrm{red}}$	3.062	3.470	8.624	5.427	5.419	0.443
$\mathrm{RMSE}_{in}^{\mathrm{PL}}$	5.046	4.200	216.223	4.968	5.360	1.859
$\mathrm{RMSE}_{fin}^{\mathrm{PL}}$	5.053	4.200	11.743	6.233	4.811	0.686

Here, looking at the RMSE values of *red* and *PL* trajectories, we can notice that they have a pretty small values in comparison with the *dyn* and the optimal trajectories in Tab. 4.3. Here we recall the fact that the error of the moment depends on the identified inertias through the acceleration $\ddot{q}_1$ and $\ddot{q}_2$. In this way the trajectories with smaller acceleration for the first two joints, have a smaller error to compensate, which means that the moment simulation can be satisfying even with a low accuracy of the identified parameters. In order to verify this, the position and the acceleration for the optimal trajectory and for the *PTP and LIN*, are plotted for each joint in Fig. 4.10 and 4.11. The figures show, as we expected, that $\ddot{q}_1$ and $\ddot{q}_2$ of the optimal trajectory have a bigger magnitude comparing with the ones of the PTP and LIN, which are equal to zero for half of the entire trajectory. Consequently, PTP and LIN gives the better results.

Furthermore, for every trajectory, the RMSE error is computed for each component of the base wrench, and the results are shown in Table 4.4. Here, the same conclusions as in Section 4.5 can be deduced for each of the tested trajectory. In particular, comparing Tab. 4.4 of the cross correlation with Tab. 4.3, we observe that for each of the cross validation trajectory, the RMSE of the third component of the wrench has a greater magnitude w.r.t. the one of the other components. Furthermore plotting the results for the case of Demo Red Box trajectory, see Fig. 4.12 and 4.13, we can notice that the third component of the simulated force, seems to be affected by an offset. At this point, even though the mean of the offset is about the 3.7% of the measurements mean, this error might represent a weakness in the adjustment procedure that we performed, based on the optimal trajectory. On this purpose the recorded measurement data contains all the necessary to apply future improving measures as in [4], e.g. time domain averaging and estimation of the measurement noise, that have not been performed in this work for time reasons.

Concluding, the simple parameter identification performed to identify the $\boldsymbol{\kappa}^{\mathrm{w}}$ parameter set might not have the best accuracy, however this represents the best estimate of the base wrench, starting with the already existing information about the dynamics of the Roberta, and avoiding a new parameter identification work, that would have needed a longer time. Therefore, for what is needed for the rest of this work, the plots show that the results obtained after the adjustment procedure, are enough consistent with the trend of the measurements, so despite this margin of uncertainty, we decide to keep this base wrench formulation as reliable information for our next work topic.

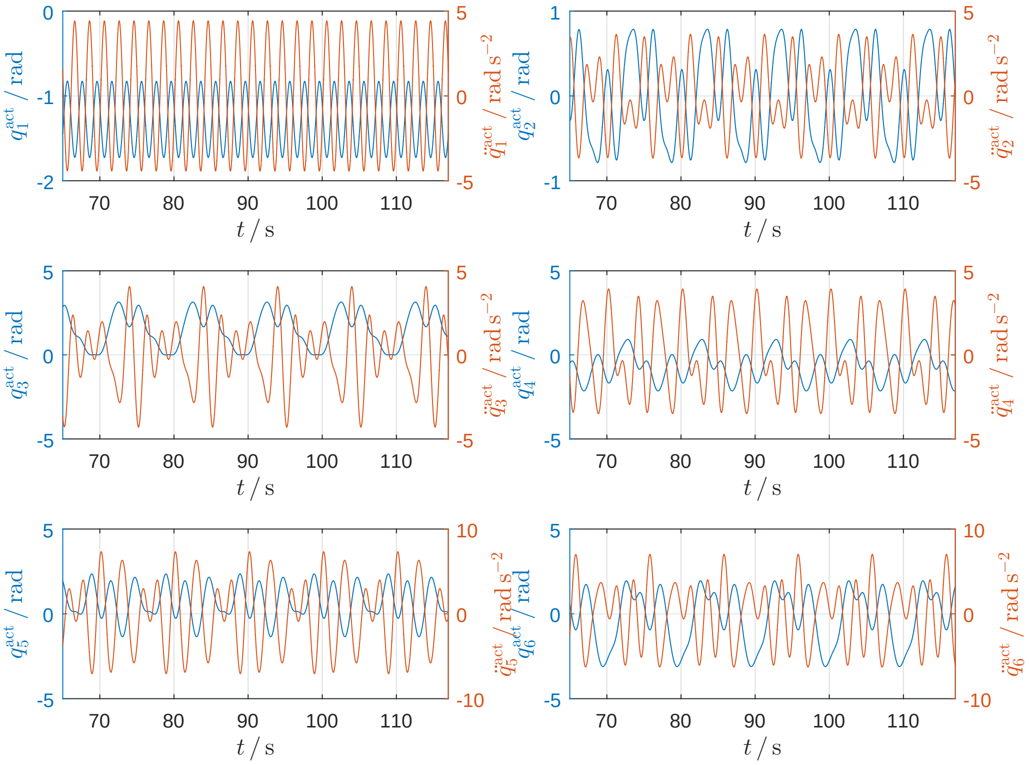


Figure 4.10: Actual position and acceleration of the Roberta joints, under the optimal trajectory: 5 periods are shown.

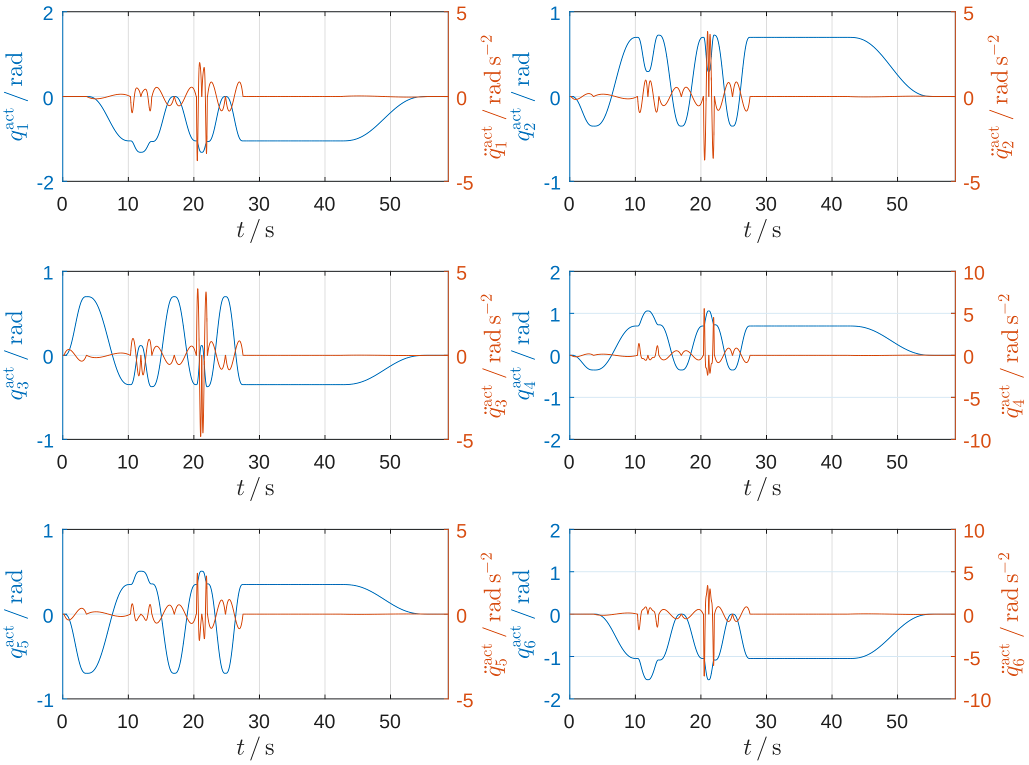


Figure 4.11: Actual position and acceleration of the Roberta joints, under the PTP and LIN. The entire trajectory is shown.

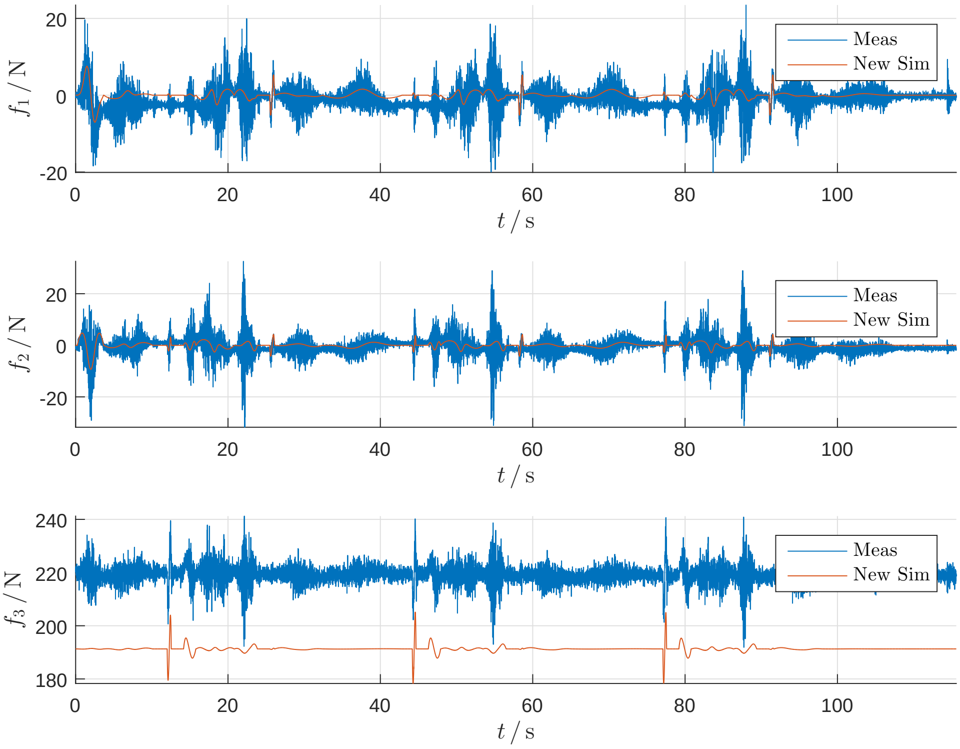


Figure 4.12: Wrench force components, simulated on Demo Red Box trajectory with error compensation in (3.72), are compared with the measures. 5 periods are shown.

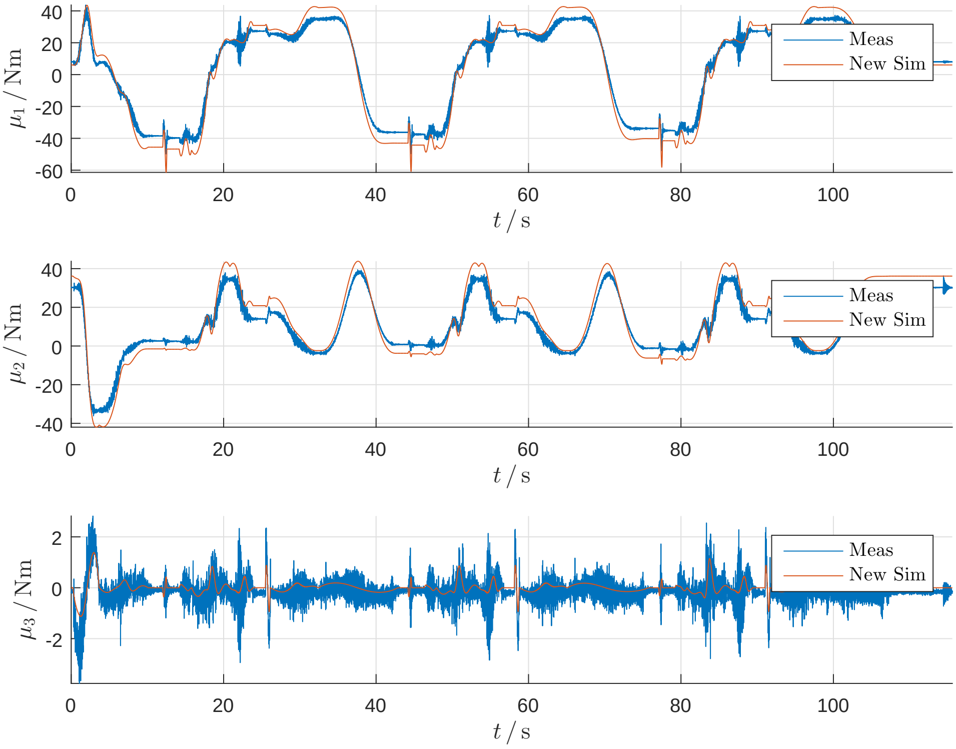


Figure 4.13: Wrench moment components, simulated on Demo Red Box trajectory with error compensation in (3.72), are compared with the measures. 5 periods are shown.

5 Tip-Over Analysis

For the final goal of evaluating dangerous configurations of the system, we have to face the problem that some information regarding the Roberta are unknown. In the previous chapters, the kinematics and the dynamics have been introduced for the robotic system in study, composed by two different components that are a 6-DoF manipulator and an autonomous mobile platform. In particular it has been found a formulation for the base wrench, considered reliable for our purpose. Hence, the underlying idea is to adopt the moment height stability measure (MHS), that is going to be introduced afterwards, which let us exploit this information extracted from the Roberta.

5.1 Moment-Height Stability Measure Introduction

The MHS measure has been chosen for this work for an important point that makes possible the initial idea of using the derived base wrench, that is to treat the system as an assembly of two separate subsystems. For the fact that needed information about the Roberta links is missing, this property is crucial to let us perform the stability analysis, without deriving a proper coupled model. Therefore, the insight here is to split up the mobile platform in the Roberta P80.800, the upper subsystem, and a lower subsystem, composed by the Adept Lynx platform and the Roberta control cabinet, with dimensions in Fig. 5.2(a), mounted on it, in a way that the base wrench reaction forces act as interaction forces between the two subsystems, as in Fig. 5.1.

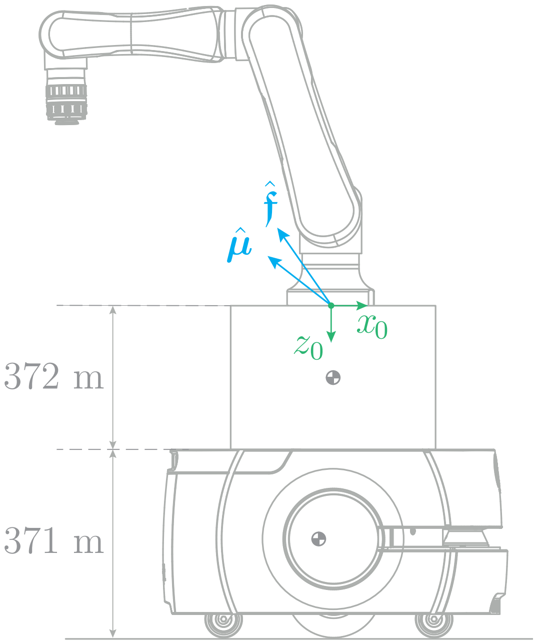
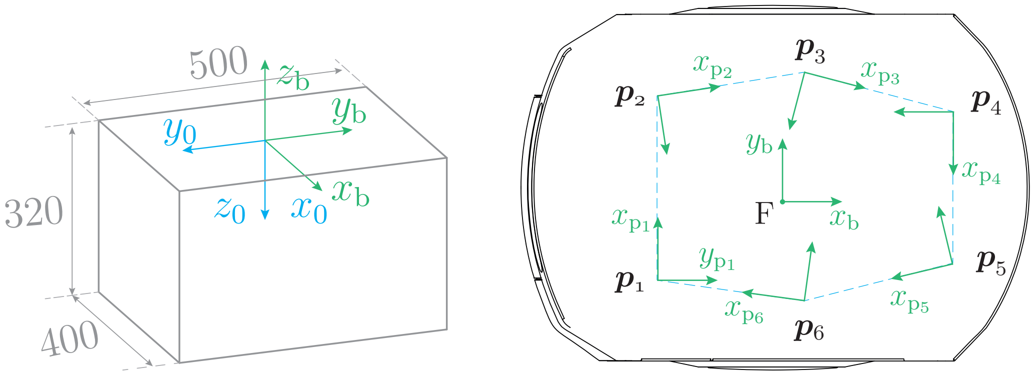


Figure 5.1: Upper and lower subsystems with the interaction forces.



(a) Control cabinet dimensions in mm, with frames S_0 and S_b.

(b) Boundary polygon and contact points are shown, with their corresponding reference frames.

Figure 5.2: Lower subsystem reference frames.

5.1.1 Geometric Model

Before starting the stability analysis, we recall that the Adept Lynx platform in use, is equipped with 2 active wheels and 4 casters, whose ideal contact points with the ground represents the vertices $\boldsymbol{p}_i$ of a convex polygon, which represents the so called *support boundary polygon*, depicted in Fig. 5.2(b).

For the analysis that is going to be performed, it is useful to name the polygon vertices from $\boldsymbol{p}_1$ to $\boldsymbol{p}_6$ in clockwise direction, and to place a reference frame S_{p_i} in correspondence of each vertex $\boldsymbol{p}_i$, whose x_{p_i}-axis points towards the next vertex $\boldsymbol{p}_{i+1}$ and the other axes y_{p_i} and z_{p_i} are chosen to complete a right-handed frame (see Fig. 5.2(b)). A further right-handed reference frame S_b is placed on the top of the upper subsystem, with x_b-axis pointing toward the front of the platform and with origin assumed to represent the point, where the Roberta P80.800 is attached. Then, we denote $\boldsymbol{e}_i$ as the edge between points $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ such that

$$\boldsymbol{e}_i = \boldsymbol{p}_{i+1} - \boldsymbol{p}_i \quad , \qquad i = 1, 2, \dots, 5 \tag{5.1}$$

$$\boldsymbol{e}_6 = \boldsymbol{p}_1 - \boldsymbol{p}_6 \quad . \tag{5.2}$$

For each edge of the support polygon, we define the unit vector $\hat{\boldsymbol{a}}_i$ with the same origin and direction of the corresponding x_{p_i} axis, such that

$$\hat{\boldsymbol{a}}_i = \frac{\boldsymbol{p}_{i+1} - \boldsymbol{p}_i}{||\boldsymbol{p}_{i+1} - \boldsymbol{p}_i||} = \frac{\boldsymbol{e}_i}{||\boldsymbol{e}_i||}, \qquad i = 1, 2, \dots, 5 \tag{5.3}$$

$$\hat{\boldsymbol{a}}_n = \frac{\boldsymbol{p}_1 - \boldsymbol{p}_6}{||\boldsymbol{p}_1 - \boldsymbol{p}_6||} = \frac{\boldsymbol{e}_6}{||\boldsymbol{e}_6||} \quad , \tag{5.4}$$

in a way that all the unit vectors make a clockwise closed loop.

Now, we highlight the fact that in studying the stability of mobile manipulators, multiple factors have to be taken into account, e.g. height of the system center of gravity, velocity of the platform, accelerations of both platform and links of the manipulator, mass moment of inertia of the mobile base, interaction forces and torques between the manipulator end-effector and the environment, and the amount of load transferred by the manipulator on the platform. Assuming that the wheels do not slip during the motion, and that the platform drives in a flat terrain, a tip-over occurs when the vehicle rotates about one of the support pattern edges, called *tip-over axis*. At the same time, when the platform experiences sudden change in its direction e.g. due to uneven ground conditions, the tip-over occurs about some combination of support pattern edges. Therefore, the MHS measure presented in [1] defines a criterion based on the stabilizing and destabilizing moments for both cases, where the yaw instability is not considered as unstable pose for the system, unlike in the car dynamics.

Therefore, as mentioned, the mass moment of inertia of the mobile base, also counts in the attitude of the platform to tip-over w.r.t. a certain axis. Hence, with the initial assumption of uniform mass distribution, the mass moment of inertia of the entire lower subsystem, has been computed with the help of a CAD-software, respect to each edge $\boldsymbol{e}_i$ of the support polygon, taking the corresponding x_{p_i} in Fig. 5.2(b) as reference. Results are shown in Tab. 5.1.

Table 5.1: Inertia values $I_{\mathrm{p}_i,x}$ with respect to x_{p_i}-axes.

	$I_{\mathrm{p}_1,x}$	$I_{\mathrm{p}_2,x}$	$I_{\mathrm{p}_3,x}$	$I_{\mathrm{p}_4,x}$	$I_{\mathrm{p}_5,x}$	$I_{\mathrm{p}_6,x}$
Inertia $(\mathrm{kg\,m^2})$	14.894	13.338	13.025	14.332	13.045	13.361

5.2 MHS Algorithm for a Stationary Base

At this point, we can first perform an analysis on the stability of the system, by testing the MHS measure for a stationary platform and a moving manipulator with no load.

Therefore in the case of a static base, all forces and torques exerted to the base body due to manipulator motion, gravitational and inertia forces, described in the S_{b} frame, are considered by taking the transformation of the base wrench obtained in Section 4. In this way the effect of manipulator arm is fully substituted by a wrench whose origin is point F, that here is assumed to be in correspondance of the base frame origin.
So first we need to extract force and torque exerted on the base, and this can be done by taking the base wrench $\bar{\boldsymbol{f}}^{\mathrm{w}}$ computed in (3.72), which is a reaction force computed with respect to frame S_0, and consequently

- flip the sign according with (3.38), in a way that we can obtain the wrench exerted by the manipulator on the platform, w.r.t. S_0;
- apply the force-torque transformation to describe it w.r.t. S_{b},

$$ {}^{\mathrm{b}}\boldsymbol{f} = {}^{\mathrm{b}}\boldsymbol{f}_{\mathrm{sim}} = {}^{\mathrm{b}}\boldsymbol{X}_0 \boldsymbol{f}_{\mathrm{sim}} \ . \tag{5.5} $$

Here we observe that the simplest choice is to mount the Roberta in a way that its frame S_0 has the same origin as the frame S_{b}, with both the x-axes pointing in the same direction, that is the platform front. Assuming that, we obtain

$$ {}^{\mathrm{b}}\boldsymbol{X}_0 = \begin{bmatrix} {}^{\mathrm{b}}\boldsymbol{A}_0 & \boldsymbol{0}_{3\times 3} \\ {}^{\mathrm{b}}\boldsymbol{S}_0 \cdot {}^{\mathrm{b}}\boldsymbol{A}_0 & {}^{\mathrm{b}}\boldsymbol{A}_0 \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{b}}\boldsymbol{A}_0 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & {}^{\mathrm{b}}\boldsymbol{A}_0 \end{bmatrix} \ , \tag{5.6} $$

where forces and moments are simply rotated of a π counter-clockwise angle, see Fig. 5.2(a).

Therefore a geometric description of the boundary polygon is given, providing first the coordinates of the six vertices of the support polygon of the platform, where the wheel contact points lay, described w.r.t. the base frame fixed in (0,0,0), that are in the following Table.

Table 5.2: Coord. of the vertices of the support polygon, w.r.t. S_{b} frame. Units in mm.

	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$	$\boldsymbol{p}_4$	$\boldsymbol{p}_5$	$\boldsymbol{p}_6$
x_{b}-coordinate	-222.62	-222.62	0	203.17	203.17	0
y_{b}-coordinate	-151.16	151.16	196.64	102.45	-102.45	-196.64
z_{b}-coordinate	-743	-743	-743	-743	-743	-743

Then, after computing edges and unit vectors through (5.1) and (5.3), we have all the needed elements to implement the algorithm.

1. From (5.5) we have that the total force and moment exerted on the base frame are represented by ${}^{\mathrm{b}}\boldsymbol{f} = \left[{}^{\mathrm{b}}\mathfrak{f}, {}^{\mathrm{b}}\boldsymbol{\mu}\right] \in \mathbb{R}^{6\times 1}$. Then, the moment of this wrench about the vertices of the support polygon can be found through

$$\boldsymbol{M}_{v_i} = -\boldsymbol{p}_i \times {}^{\mathrm{b}}\mathfrak{f} + {}^{\mathrm{b}}\boldsymbol{\mu}, \quad i = 1, 2, \ldots, 6 \ , \tag{5.7}$$

where $\boldsymbol{M}_{v_i}$ indicates the moment about $\boldsymbol{p}_i$, i-th vertex of support polygon. Here we also want to involve the lower subsystem mass contribution, thus we obtain

$$\boldsymbol{M}_{v_i} = -\boldsymbol{p}_{\mathrm{p}} \times \begin{bmatrix} 0 \\ 0 \\ -m_{\mathrm{p}}g \end{bmatrix} - \boldsymbol{p}_{\mathrm{c}} \times \begin{bmatrix} 0 \\ 0 \\ -m_{\mathrm{c}}g \end{bmatrix} - \boldsymbol{p}_i \times {}^{\mathrm{b}}\mathfrak{f} + {}^{\mathrm{b}}\boldsymbol{\mu} \ , \tag{5.8}$$

where $\boldsymbol{p}_{\mathrm{p}}$ and $\boldsymbol{p}_{\mathrm{c}}$ are the vectors pointing from the origin of frame S_{p}, respectively to the platform and to the cabinet center of gravity, m_{p}=60 kg and m_{c}=20.58 kg are the platform and cabinet masses assumed to be uniform in their volume, and g represents the gravity acceleration term.

2. These moments about the vertices, can now be projected about their edge, through the following scalar product applied on the corresponding unit vector

$$\boldsymbol{M}_i = \boldsymbol{M}_{v_i} \cdot \hat{\boldsymbol{a}}_i \ , \quad i = 1, 2, \ldots, 6 \ . \tag{5.9}$$

3. We finally have all the elements to compute the dynamic MHS measure α, by considering the most critical case that is

$$\alpha = min\{\alpha_i\} \ , \tag{5.10}$$

where α_i, is the so called *dynamic stability margin* about the i-th edge, such that

$$\alpha_i = (I_{\mathrm{p}_i,x})^{\beta_i} \boldsymbol{M}_i, \quad i = 1, 2, \ldots, 6 \ . \tag{5.11}$$

Here $I_{\mathrm{p}_i,x}$ is the base moment of inertia in Tab. 5.1, and β_i is taken as

$$\beta_i = \begin{cases} +1, & if \quad \boldsymbol{M}_i > 0 \\ -1, & otherwise. \end{cases}$$

Observe that the MHS measure α, does not take into account the height of the whole system center of gravity, therefore this measure could still be improved, through direct incorporation of the height of the manipulator center of mass, as given in [1]. Unfortunately in our work, we cannot directly estimate this manipulator parameter, consequently we improve the MHS measure by adding a normalization s.t.

$$\hat{\alpha} = \frac{min\{\alpha_i\}}{\alpha_{nom}} \ , \tag{5.12}$$

where $\hat{\alpha}$, is now the normalized dynamic stability margin, and α_{nom} refers to a specific nominal value. Observe that taking α_{nom} as the α corresponding to the most stable configuration of a specific trajectory, $\hat{\alpha}$ will vary between zero and one. Another choice would be to α_{nom} as the one of the starting configuration, in a way that the stability measure will be weighted on it. Note that this proposed normalized measure, yields a relative stability state, that does not specify an absolute value. On this purpose, additional considerations can be done.

- Observe that the inner product in Eq. (5.9) implies that when α_i is positive, the moment about the i-th edge is stabilizing, otherwise it is negative when the moment is destabilizing.
- As we mentioned, the MHS measure incorporates the mass moment of inertia of the moving base. For a stable case, the higher the I_{vi} results, the more secure the system stability will be, so β_i has to be considered equal to 1. Otherwise, in case of destabilizing moment, an higher I_{pi} causes a slower tip over, according with taking β_i = -1 in Eq. (5.11).
- If α, that is the minimum of the α_i's, is positive, the system is stable. The zero value of α represents the critical dynamic stability, and negative values of α notices that the system goes to instability.

5.3 MHS Algorithm for a Moving Base

As we mentioned, the MHS algorithm has just been addressed for the case of stationary platform, so in this section, we want to clarify how to include the case where also the platform is moving. In fact, this algorithm can be extended to the general dynamic case, if we add the dynamic effects of the moving platform, to the external applied wrench. On this purpose, this can be done by formulating the manipulator base wrench, done in Eq. (3.37) for the case of stationary base, taking the initial conditions in (3.27)-(3.28) equal to the platform angular velocity and acceleration. This case is not going to be discussed in this work.

5.4 MHS Verification

In order to verify the results of this method, we implement the algorithm with the software MATLAB, and we run it with the given geometric description of the lower subsystem, that is the platform and the Roberta control cabinet. The underlying idea is to follow the approach in Section 5.2, and run simple trajectories on the Roberta, in order to be able to expect what the result of the MHS algorithm has to be, in advance, and qualitatively determine the goodness of the method.

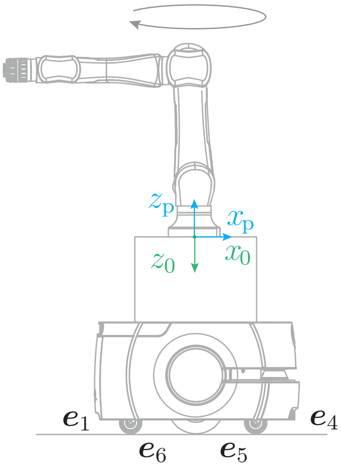


Figure 5.3: [Roberta from configuration $\boldsymbol{q}$=$\mathbf{0}$, rotate itself in clockwise direction.

5.4.1 Roberta Clockwise Rotation

First of all, we recall that we make the assumption that the frame S_p of the lower subsystem, is in correspondence of the origin of manipulator frame S_0. At this point, an intuitive verification would be to give an arbitrary rotation to the Roberta in its home configuration $\boldsymbol{q} = \mathbf{0}$, in Fig. 5.3, in order to expect the Roberta to generate destabilizing moments along every edge of the boundary polygon, in a sequence.

Under this idea, we design a fifth order polynomial trajectory, from 0 to π rad, to give to q_1, first joint of the Roberta, which is useful to avoid high accelerations at the beginning and at the end of the motion. Therefore, a 0.1 s simulation of the compensated base wrench of the Roberta is run, actuating the first joint with the designed trajectory and keeping the other joints stationary. Next, the MHS measure is applied at each instant time, on the simulated base wrench, providing the normalized stability measure computed with (5.12), where the nominal weighting value is taken as the α of the first configuration of the system, so that all the next configurations will be compared to it. Fig. 5.4 shows the results of the applied algorithm, where the time instants of interest have been labeled with t_i, for $i = 1, \ldots, 7$. From the simulation, results that the system is stable for the first time interval, enclosed between 0 and t_1, where the nominal stability measure gives positive values. Therefore, after t_1 s, the stability measure becomes negative, the system is considered to be unstable, and the information about the tip-over axis, respect to which the tip-over is occurring, is provided by the tip-over edge $\boldsymbol{e}_i$, depicted in figure within the corresponding time interval. As we expected, all the edges have been considered critical in

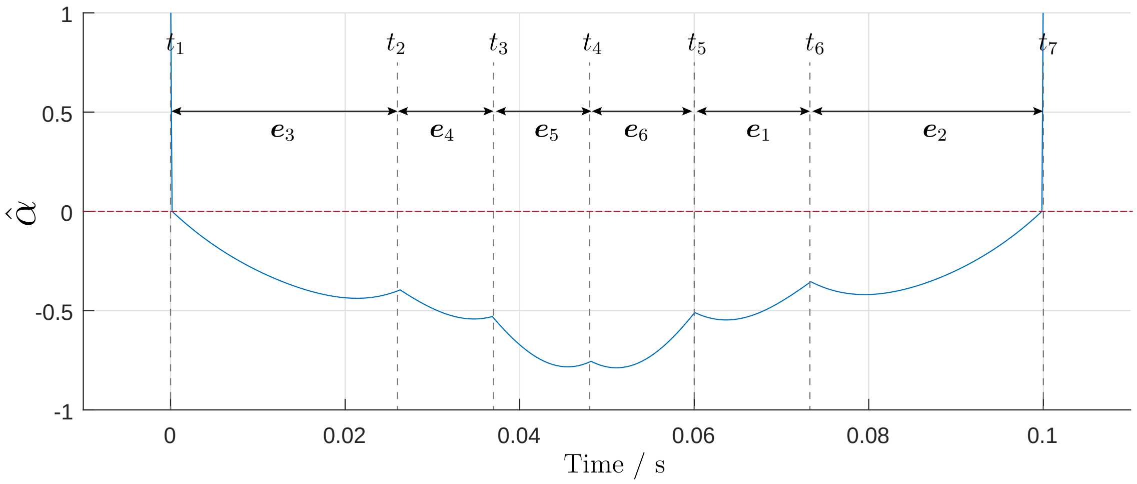


Figure 5.4: Evaluated stability measure along the time, applied on the Roberta base wrench simulations, with simulation time 0.1 s. Tip-over edges are shown when the system is unstable.

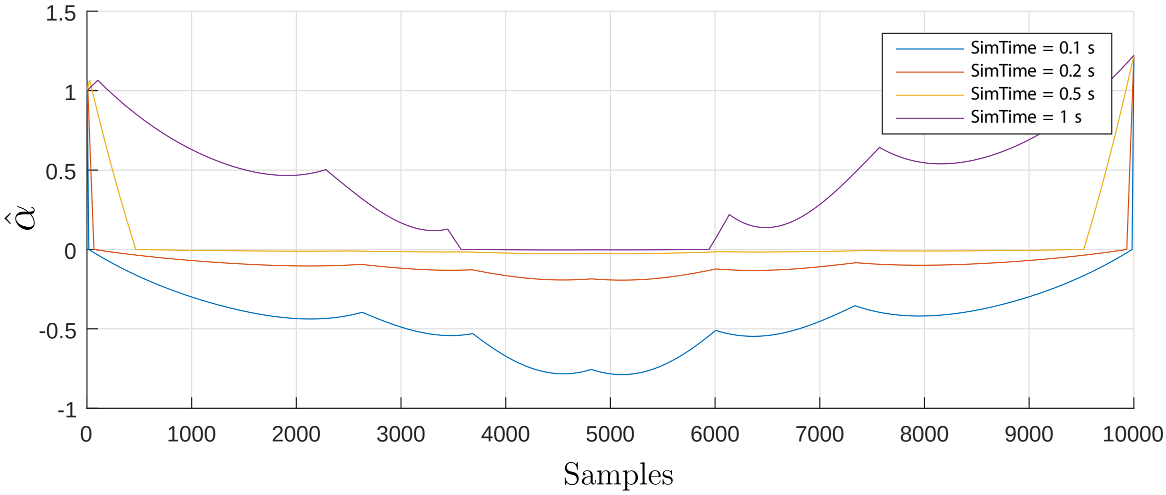


Figure 5.5: Evaluated stability measure, applied on the Roberta base wrench simulations, with different simulation time. Tip-over edges are omitted.

a clockwise loop, during the motion of the Roberta, as a result of the trajectory applied. At the end of the trajectory, after t_7 s the stability measure evaluates again the system as stable, as a consequence of the progressive diminution of the deceleration.

A further verification can be applied by operating on the acceleration/deceleration of the actuated joint q_1. On this purpose, we expect that decreasing the velocity and consequently the acceleration, through which the motion is performed, the system will result more stable. Therefore, additional simulations have been performed by increasing the simulation time, that let the fifth order polynomial trajectory, design the motion with lower acceleration values .

The results are shown in Fig. 5.5 for simulation time of 0.2, 0.5 and 1 s, that bring meaningful results for the applied method. The normalized stability measure, has been computed with nominal weighting term, taken as the α of the first configuration of the system. Here, for simulation time of 0.2 s, at each instant time, the stability measure is grater if compared to the previous one. The case of a 0.5 s simulation time bring the system to be in the critical dynamic stability for almost all the motion performed and finally the case of 1 s, shows that the system is stable for most of the time, with the exception of the middle of the trajectory, when the joint acceleration is at its maximum, that the stablity measure is at the border of the threshold.

Therefore, as we expected, the longer is the simulation time, the lower is the acceleration of the motion and as a consequence, the more stable result the overall movement.

5.5 Application on Demo-RedBox Trajectory

In this section we want to apply the MHS algorithm in order to evaluate the stability of the system in study, with a stationary platform and the Roberta moving on a realistic trajectory. For doing this, we choose the *Demo Red Box* trajectory used in Sec. 4.6 for cross-validating the base wrench formulation, that provides a series of movements that such manipulators are used to perform. Hence similarly, we run the Roberta base wrench simulation on this trajectory, and we next apply the MHS algorithm on the outcomes. Here we point out that the nominal stability measure in Eq. (5.12), has been taken as the one of the most stable configuration of the running trajectory, in a way that all the other configurations are compared w.r.t. it. Fig. 5.6 shows the stability measure computed along the time. Summarizing, the stability measure evaluates the criticality of the movement performed by the Roberta, from produced forces and moments at its base, which act on the lower subsystem as external wrench. In this case of Demo Red Box trajectory, these wrenches computed along the time, are not enough to let the MHS measure set the system as unstable. These conclusion set the basis for a proposed offline tip-over evaluation, that is presented in 5.6.

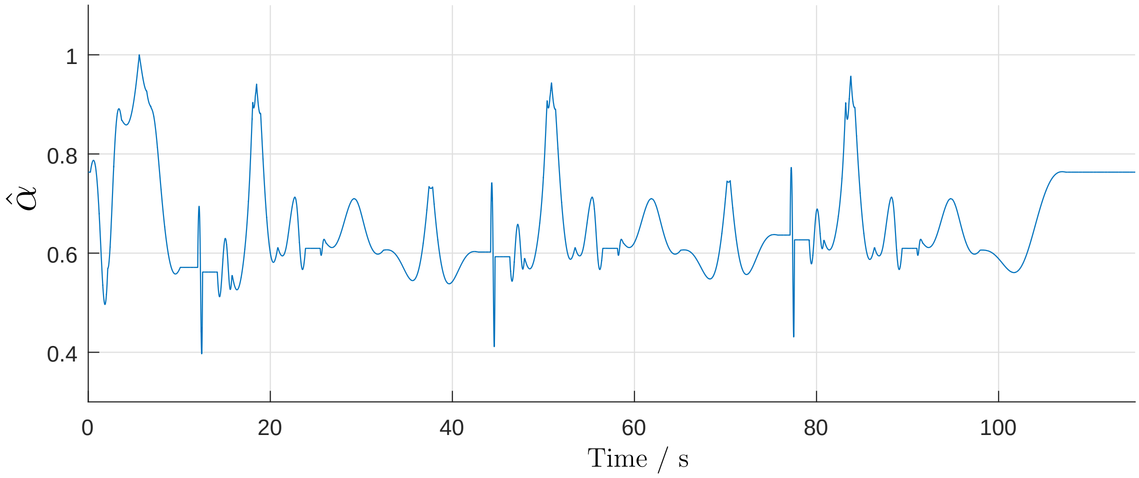


Figure 5.6: Evaluated stability measure applied on the Roberta base wrench simulations, performed on Demo Red Box trajectory.

5.6 Offline Tip-over Evaluation

At this point of the work, we can sum up all the elements treated in the previous chapters, to propose a complete model-based tip-over avoidance method, for the mobile manipulator in study. For a given trajectory to the mobile manipulator, this proposed offline analysis aims to evaluate the system safety, by simulating its stability in advance, showing the trend of the stability measure along the time, and alerting the user whether a tip-over can occur, similarly in what has been done in Section 5.4 for the case of a static base.
In the specific, we propose to plan a trajectory to the overall system by separating the one for the Adept Lynx platform, to the one for the Roberta P80.800 (differently from the literature). In this case we can estimate angular velocity and acceleration of the platform, as addressed in Section 3.6, and evaluate an expression for the Roberta base wrench, that takes into account the motion of the Roberta, as well as the platform dynamic effects of the platform, see Section 3.3.2. On this purpose we recall that in the formulation of the base wrench, can also be considered an external payload, which can be integrated in the case of gripping/lifting tasks. Consequently, the MHS measure can be applied to the base wrench simulation, an a trend of the stability measure can be provided, that can notice an eventual instability occurrence, furthermore in this case, the critical edge respect to which the tip-over is about to occur. At the end the user can evaluate the risk, and decide in case, how to rearrange the overall trajectory of the system.

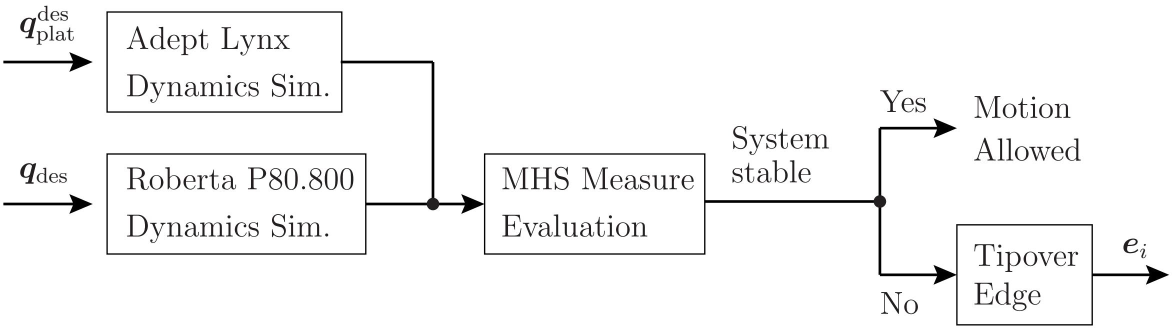


Figure 5.7: Proposed offline tip-over evaluation.

6 Conclusions and Future Perspectives

This thesis treated several topics needed to deal with a stability analysis for a mobile manipulator, built with available components in the RRRU Lab, that are the gomTec Roberta P.80.800, and the Adept Lynx autonomous mobile platform. This thesis aims to contribute to the intentions of the RRRU, to design safety measures in order to monitor the moving system, avoiding possible tip-over. For its development, results coming from previous work [4] [5], done with the purpose of developing a control cabinet for the Roberta, have been exploited.

Starting to consider the most general case of application, that is both platform and manipulator in motion, only restricting the environment to be an indoor and flat terrain, literature also offers several options for creating an unified dynamic model for the control, carrying simple examples of 2 active wheels platforms and 2-DoF manipulators with a fully provided dynamic parameters description. In these cases, the control is designed for actuating the two wheels of the platform as well as the two joints of the manipulator, that lead to a dynamic model written w.r.t. 4 variables. Anyway, since the Adept Lynx platform does not allow to apply a direct control to its single active wheel, a combined control scheme cannot be realized with the software interfaces available at the moment, so we decided to employ a separate control for both the Roberta manipulator and the Lynx platform. In addition, since the Roberta dynamic parameters (3.7) are not provided from the manufacturer, we could not follow the common Lagrangian formulation, that the literature proposes for obtaining a full coupled dynamic model. Therefore, the underlying idea was to exploit the available work in [4], which provides the dynamic parameters, identified in a grouped form (base parameter set).

Thus, because of this lack of knowledge about the Roberta, the focus was on finding a stability algorithm that could avoid this missing information, but could rather exploit the previous work done. At this point, we decided to split up the system in study in two subsystems, analyzing the interaction forces between them. In the specific, the Moment-Height Stability measure (MHS) proposes to model this interaction forces, as the manipulator reaction force and moment acting on its base (base wrench), yielding a stability measure based on the projection of this base wrench, on the platform support polygon.

As a consequence, the challenge of this work was trying to formulate an expression for the manipulator base wrench, using the unique available information regarding the Roberta dynamics, that is the link dynamic parameters in the grouped form. Here, the authors in [18] showed that exploiting the identified dynamic model of a similar 6 revolute joints manipulator, is possible to write the base wrench in a linear form w.r.t. the base parameters. However, despite the results of the authors, a certain amount of time was spent in order to conclude that, for computing the base wrench in the same way, some rearrangements are needed to exploit the previous work. In particular, few base parameters that affect the base wrench, neglected by the authors, do not appear in the identified base parameter set.

Hence at this point, another topic of this work needed to be treated, which was to find the best possible formulation for the base wrench, in order to be able to get ahead into the final goal. On this purpose we point out that to obtain a thorough base wrench formulation, a complete parameter estimation work would have been necessary. Therefore, because of time reasons and for the fact that a parameter estimation regarding the Roberta dynamics had already been done, we accorded that trying to rearrange the base parameter set from [4], represented the best trade-off between reaching the thesis goal and evaluating the base wrench with an acceptable accuracy. Then, elements of parameter identification theory have been exploited for obtaining an estimate of the unknown grouped dynamic parameters, and a formulation of the base wrench was obtained. Next, such formulation has been validated through experiments that brought satisfying results, even though the cross validation remarked that the identification could still be improved.

After having achieved this topic, we could proceed with the stability evaluation for the motion of the mobile manipulator, through the mentioned MHS measure. By simulating the base wrench for simple trajectories, the stability criterion was verified for the case of a static platform and unloaded Roberta, by bringing meaningful results. This final method could not be validated, for time reasons and for the fact that the RRRU is still not equipped with the safety measures, necessary to hold the system in case of experiments failure.

Concluding, because of this unexpected problems occurred, this approach could not be validated, so we propose some steps for additional improvements, in a future development. First of all, similarly in how it has been done in this work, we propose to formulate a base wrench expression which also considers effects of a moving base, as explained in 3.3.2 and an additional payload. Even though the results show that the model represents the real system, with an accuracy considered reliable, for what this base wrench is used in this work, the parameter identification still has potential for improvements, e.g. by applying the mentioned improving measures stated in [4].

The offline tip-over evaluation, proposed in Section 5.6, was a first interpretation in what the goal of this thesis could represent, and is supposed to be the first useful tool, that let the system in study to be monitored during the motion. Furthermore, the next step could be to implement and test this proposed evaluation, and next, to develop an online stability analysis, to be employed effectively during the motion of the system.

7 Bibliography

[1] S.A.A. Moosavian and K. Alipour. On the dynamic tip-over stability of wheeled mobile manipulators. *International Journal of Robotics and Automation*, 22(4), 2007. ISSN 1925-7090. doi: 10.2316/Journal.206.2007.4.206-3036.

[2] G. R. Nikhade, S. S. Chiddarwar, and V. S. Deshpande. Tip-over stability of omnidirectional mobile robot. In Asheesh Shukla, Abhay Chaturvedi, Atul Bansal, and Vishal Goyal, editors, *International Conference Communication, Control & Intelligent Systems (CCIS-2015)*, pages 299–303, [Piscataway, NJ] and [Piscataway, NJ], 2015. IEEE. ISBN 978-1-4673-7541-2. doi: 10.1109/CCIntelS.2015.7437928.

[3] E. G. Papadopoulos and D. A. Rey. A new measure of tipover stability margin for mobile manipulators. In *April 22 - 28, 1996, Minneapolis, Minnesota*, pages 3111–3116. IEEE, 1996. ISBN 0-7803-2988-0. doi: 10.1109/ROBOT.1996.509185.

[4] Matthias Stapf. Modeling and dynamic identification of serial robots.

[5] Oliver Kotz. Inverse kinematics, friction identification and online adaptive control of serial robots.

[6] Liang He. Tip-over avoidance algorithm for modular mobile manipulator. In Ahmed Khairy, editor, *2012 First International Conference on Innovative Engineering Systems (ICIES)*, pages 115–120, Piscataway, NJ, 2012. IEEE. ISBN 978-1-4673-4439-5. doi: 10.1109/ICIES.2012.6530855.

[7] M. J. A. Safar, K. Watanabe, S. Maeyama, and I. Nagai. Tip-over prediction for an omnidirectional mobile robot with dynamic supporting polygon. In *SICE Annual Conference (SICE), 2012 Proceedings of*, pages 149–154, 2012.

[8] Q. Huang and S. Sugano. Motion planning of stabilization and cooperation of a mobile manipulator-vehicle motion planning of a mobile manipulator. In *Robotic intelligence interacting with dynamic worlds*, pages 568–575. IEEE, 1996. ISBN 0-7803-3213-X. doi: 10.1109/IROS.1996.570852.

[9] Evangelos Papadopoulos and Daniel A. Rey. The force-angle measure of tipover stability margin for mobile manipulators. *Vehicle System Dynamics*, 33(1):29–48, 2000. doi: 10.1076/0042-3114(200001)33:1;1-5;FT029.

[10] Khalil Alipour, Arman Hasanpour, and Parisa Daemy. Comparing two online tip-over avoidance algorithms for mobile manipulators. In *International Conference on Robotics and Mechatronics Conference (ICROM 2014)*, pages 310–315, Piscataway, NJ, 2014. IEEE. ISBN 978-1-4799-6743-8. doi: 10.1109/ICRoM.2014.6990919.

[11] Muhammad Juhairi Aziz Safar, Keigo Watanabe, Shoichi Maeyama, and Isaku Nagai. Tip-over stability enhancement for omnidirectional mobile robot. *International Journal of Intelligent Unmanned Systems*, 2(2):91–106, 2014. ISSN 2049-6427. doi: 10.1108/IJIUS-12-2013-0024.

[12] Huatao Zhang and Aiguo Song. Tipover stability enhancement method for a tracked mobile manipulator. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5158–5164, [Piscataway, NJ], 2014. IEEE. ISBN 978-1-4799-3685-4. doi: 10.1109/ICRA.2014.6907616.

[13] Yuqing Wang and M. Nagai. Integrated control of four-wheel-steer and yaw moment to improve dynamic stability margin. In *Proceedings of the 35th IEEE Conference on Decision and Control: Nebent.: 1996 IEEE decision and control*, pages 1783–1784, Piscataway, NJ, 1996. IEEE Service Center. ISBN 0-7803-3590-2. doi: 10.1109/CDC.1996.572824.

[14] Rached Dhaouadi Ahmad Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics & Automation*, 02(02), 2013. ISSN 21689695. doi: 10.4172/2168-9695.1000107.

[15] W. M. Silver. On the equivalence of lagrangian and newton-euler dynamics for manipulators. *The International Journal of Robotics Research*, 1(2):60–70, 1982. ISSN 0278-3649. doi: 10.1177/027836498200100204.

[16] E. Papadopoulos and J. Poulakakis. Planning and model-based control for mobile manipulators. In *Proceedings*, pages 1810–1815, Piscataway, NJ, 2000. IEEE Operations Center. ISBN 0-7803-6348-5. doi: 10.1109/IROS.2000.895234.

[17] S. A. Moosavian and Khalil Alipour. Moment-height tip-over measure for stability analysis of mobile robotic systems. In *2006 IEEERSJ International Conference on Intelligent Robots and Systems*, pages 5546–5551, Piscataway NJ, 2006. IEEE. ISBN 1-4244-0258-1. doi: 10.1109/IROS.2006.282270.

[18] Martin Grotjahn and Bodo Heimann. Symbolic calculation of robots' base reaction-force/torque equations with minimal parameter set. In Adam Morecki, Giovanni Bianchi, and Cezary Rzymkowski, editors, *Romansy 13*, volume 422 of *International Centre for Mechanical Sciences, Courses and Lectures*, pages 59–65. Springer Vienna, Vienna, 2000. ISBN 978-3-7091-2500-7. doi: 10.1007/978-3-7091-2498-7{\textunderscore}5.

[19] Bruno Siciliano and Lorenzo Sciavicco, editors. *Robotics: Modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, London, 2009. ISBN 978-1-84628-642-1.

[20] N. Sarkar, Xiaoping Yun, and V. Kumar. Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. *The International Journal of Robotics Research*, 13(1):55–69, 1994. ISSN 0278-3649. doi: 10.1177/027836499401300104.

[21] Subir Kumar Saha and Jorge Angeles. Dynamics of nonholonomic mechanical systems using a natural orthogonal complement. *Journal of Applied Mechanics*, 58(1):238, 1991. ISSN 00218936. doi: 10.1115/1.2897157.

[22] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul. On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(2):69, 1980. ISSN 00220434. doi: 10.1115/1.3149599.

[23] W. Khalil and E. Dombre. *Modeling, identification & control of robots: Originally published: London : HPS, 2002*. Kogan Page Science, London, 2004. ISBN 1-903996-66-X.

[24] Rolf Isermann. *Identifikation Dynamischer Systeme 1: Grundlegende Methoden*. 2., neubearb. und erw. aufl. edition, 1992. ISBN 978-3-642-84680-9.

Appendices

A Cross Correlation Plots

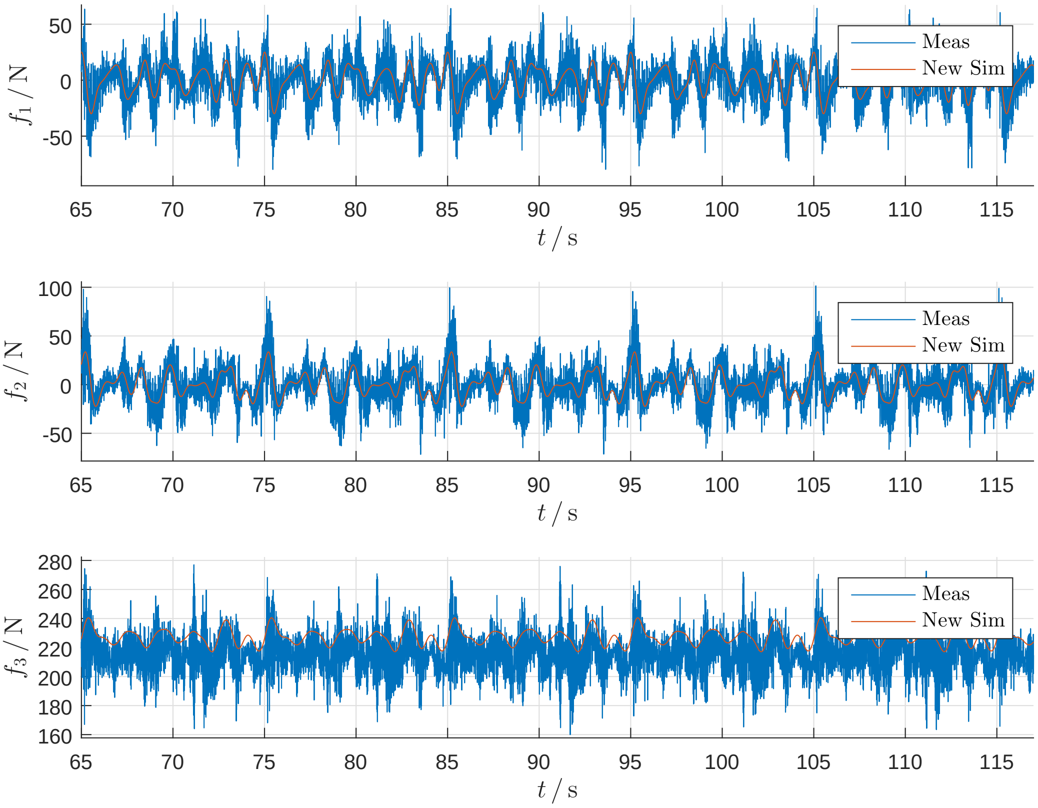


Figure A.1: Base wrench force components, simulated on the *Dynamics excitation* trajectory with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods.

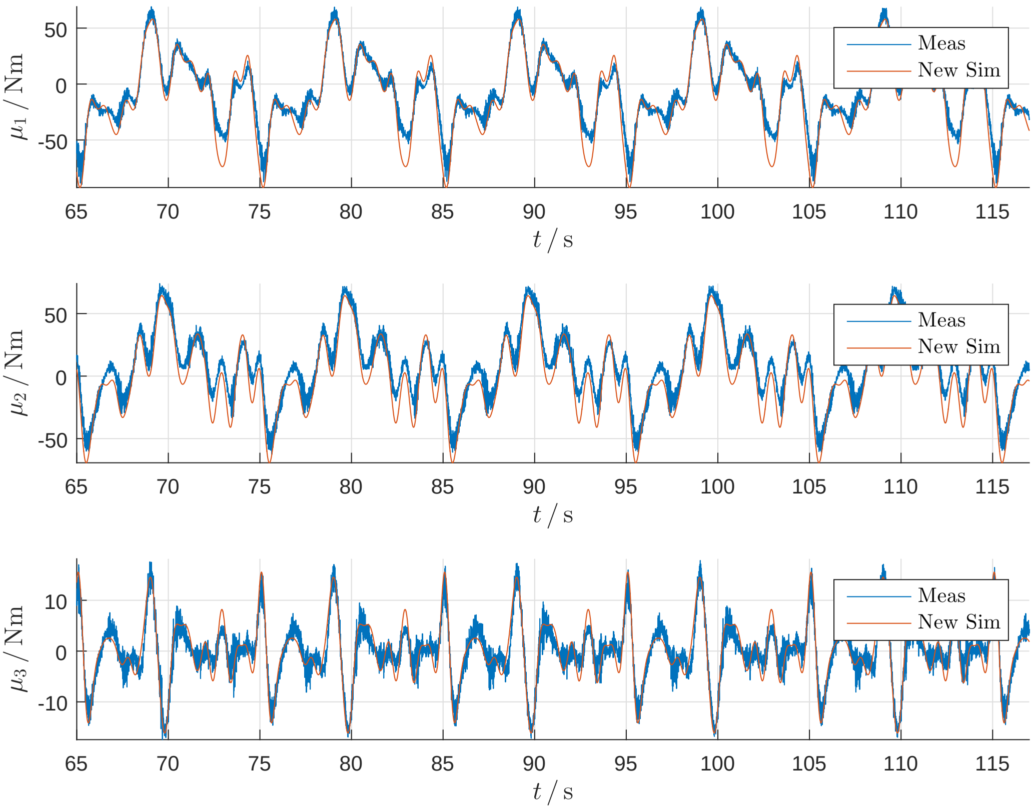


Figure A.2: Base wrench moment components, simulated on the *Dynamics excitation* trajectory with the overall error compensation in (3.72), are compared with the measurements. The plot shows 5 periods.

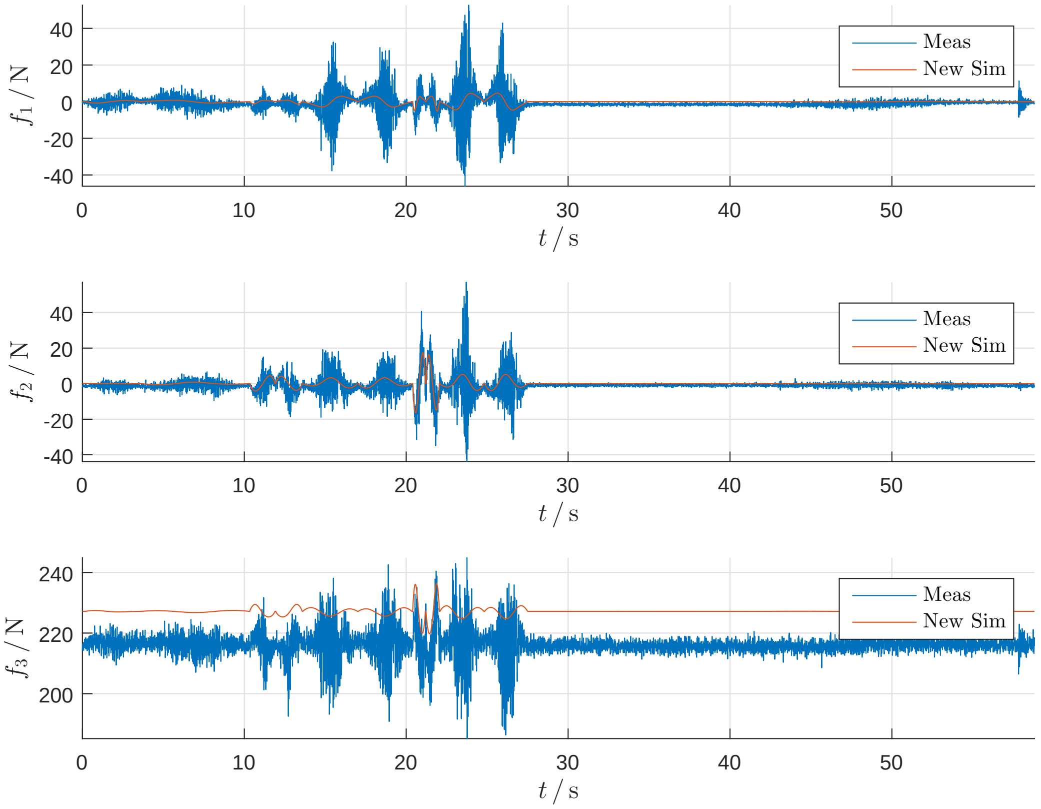


Figure A.3: Base wrench force components, simulated on the *PTP and LIN* movement with the overall error compensation in (3.72), are compared with the measurements. The plot shows the entire trajectory.

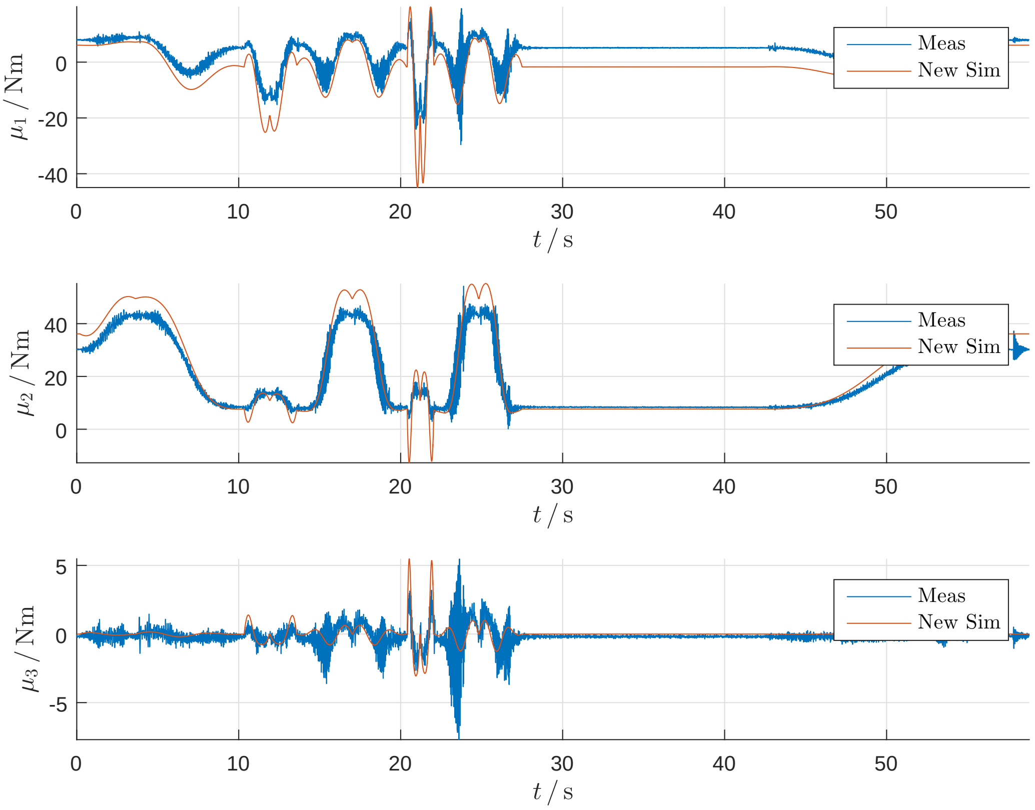


Figure A.4: Base wrench moment components, simulated on the *PTP and LIN* movement with the overall error compensation in (3.72), are compared with the measurements. The plot shows the entire trajectory.

B Lower Subsystem Mass Integration in (5.8)

In this section, we want to point out the meaning of Eq. (5.8), where we included the lower subsystem mass within the wrench moment, about the vertices of the support polygon.

From [19], for a rigid body of mass m subject to gravitational force and to active and or reaction forces $\mathfrak{f}_1, \dots \mathfrak{f}_n$, concentrated respectively at points $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$, the resultant of the external forces $\mathfrak{f}$ and the resultant moment $\boldsymbol{\mu}_\Omega$ with respect to a pole Ω are

$$\mathfrak{f} = m\boldsymbol{g} + \sum_{i=1}^{n} \mathfrak{f}_i \tag{B.1}$$

$$\boldsymbol{\mu}_\Omega = m\boldsymbol{g} \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_\mathrm{C}) + \sum_{i=1}^{n} \mathfrak{f}_i \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_i) \ , \tag{B.2}$$

where $\boldsymbol{p}_\Omega$ is the coordinate vector of pole Ω and $\boldsymbol{p}_\mathrm{C}$ is the coordinate vector of the rigid body center of mass. In the case when $\mathfrak{f}$ and $\boldsymbol{\mu}_\Omega$ are known and we want to compute the resultant moment with respect to a point Ω' different than Ω, the following relation holds

$$\boldsymbol{\mu}_{\Omega'} = \boldsymbol{\mu}_\Omega + \mathfrak{f} \times (\boldsymbol{p}_{\Omega'} - \boldsymbol{p}_\Omega) \tag{B.3}$$

that is in accord with the force-torque transformation in Eq.(3.2).

In the light of this, we want to apply this result to a simplified case, where a control cabinet of mass m_c is mounted on a wheeled platform of mass m_p, where we ignore the mass of the wheels. Here we denote with $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ the wheel contact points of the platform and we assume that a known force and moment are applied in the base frame S_b origin, respectively $\mathfrak{f}'$ and $\boldsymbol{\mu}'$. Thus, from (B.1), we obtain the resultant of the external forces

$$\mathfrak{f} = (m_\mathrm{p} + m_\mathrm{c})\boldsymbol{g} + \mathfrak{f}' \tag{B.4}$$

and similarly from (B.2) we obtain the resultant moment $\boldsymbol{\mu}_{p_1}$ with respect to pole $\boldsymbol{p}_1$

$$\boldsymbol{\mu}_{p_1} = m_\mathrm{p}\boldsymbol{g} \times \left(\boldsymbol{p}_1 - \boldsymbol{p}_\mathrm{p}\right) + m_\mathrm{c}\boldsymbol{g} \times (\boldsymbol{p}_1 - \boldsymbol{p}_\mathrm{c}) + \mathfrak{f}' \times \boldsymbol{p}_1 + \boldsymbol{\mu}' \ . \tag{B.5}$$

where $\boldsymbol{p}_\mathrm{p}$ and $\boldsymbol{p}_\mathrm{c}$ are the vectors, describing respectively the platform and cabinet center of mass w.r.t. frame S_b. Repeating this procedure for $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ at the end we have obtained the resultant of the external force and moment, w.r.t. each contact point of the platform.

Now observe that if we make the hypothesis that platform and cabinet center of mass and the origin of frame S_b lay on the same vertical axis, orthogonal w.r.t. the ground, Eq. (B.5) results equivalent to

$$\boldsymbol{\mu}_{p_1} = (m_\mathrm{p} + m_\mathrm{c})\boldsymbol{g} + \mathfrak{f}' \times \boldsymbol{p}_1 + \boldsymbol{\mu}' \ . \tag{B.6}$$

C Base Principle of MHS Measure

In this section we aim to show the base principle of the MHS measure. We consider the example in Figure C.1, composed by a control cabinet mounted on a wheeled mobile platform, with the assumption that the corresponding center of masses lay on the z_b-axis.

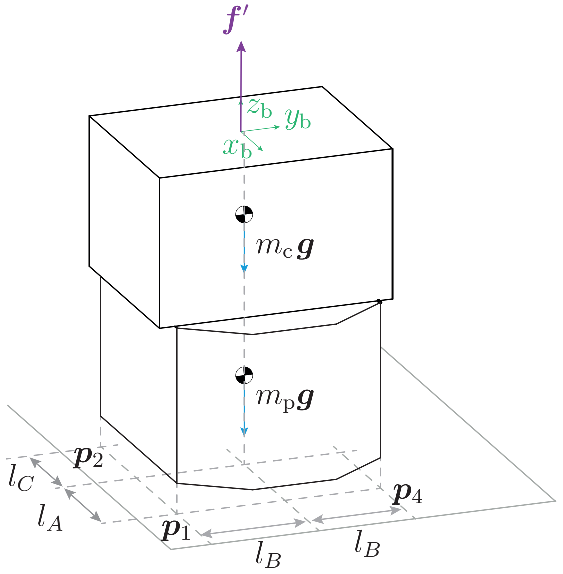


Figure C.1: Acting forces, lengths, and base reference frame of the example.

Here we assume to have a simple example of a generic vertical reaction force applied at the origin of the base frame, that is $\mathfrak{f}' = \begin{bmatrix}0, & 0, & f'_z\end{bmatrix}^\top$. From the formulas in Eq. (B.4) and (B.5) and the quantities in Fig. C.1, we have that the resultant forces and moment applied to point $\boldsymbol{p}_1$ are

$$\mathfrak{f} = \begin{bmatrix} 0 \\ 0 \\ -(m_\mathrm{p} + m_\mathrm{c})g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ f'_z \end{bmatrix} , \tag{C.1}$$

$$\boldsymbol{\mu}_{p_1} = \begin{bmatrix} 0 \\ 0 \\ -m_\mathrm{p} g \end{bmatrix} \times \begin{bmatrix} l_A \\ -l_B \\ -h_\mathrm{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_\mathrm{c} g \end{bmatrix} \times \begin{bmatrix} l_A \\ -l_B \\ -h_\mathrm{c} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ f'_z \end{bmatrix} \times \begin{bmatrix} l_A \\ -l_B \\ -h_\mathrm{b} \end{bmatrix} \tag{C.2}$$

$$= \begin{bmatrix} (f'_z - (m_\mathrm{p} + m_\mathrm{c})g)l_B \\ (f'_z - (m_\mathrm{p} + m_\mathrm{c})g)l_A \\ 0 \end{bmatrix} . \tag{C.3}$$

where we denote with h_p and h_c, respectively the height of the platform and the cabinet center of gravity, and with h_b the height of the base frame.

Now we obtained the three components of the transformed moment. Observe that the applied moment is equal to zero, when the following equations are fulfilled

$$f'_z = (m_\mathrm{p} + m_\mathrm{c})g \cdot l_B \tag{C.4}$$
$$f'_z = (m_\mathrm{p} + m_\mathrm{c})g \cdot l_A \ , \tag{C.5}$$

that represent the critical stability condition for the system, when the reaction force balance the gravitational force of the overall involved mass. Furthermore, observe that whether we have $f'_z > (m_\mathrm{p} + m_\mathrm{c})g \cdot l_B$ a positive moment w.r.t. the x_b -axis is applied, and similarly when we have $f'_z > (m_\mathrm{p} + m_\mathrm{c})g \cdot l_A$ we obtain a positive moment w.r.t. the y_b -axis, that represents the case of tip-over occurrence. Same conclusions can be done for point $\boldsymbol{p}_2$ that yields

$$\boldsymbol{\mu}_{p2} = \begin{bmatrix} (f'_z - (m_\mathrm{p} + m_\mathrm{c})g)l_B \\ (-f'_z + (m_\mathrm{p} + m_\mathrm{c})g)l_C \\ 0 \end{bmatrix} \ , \tag{C.6}$$

and for all the other points of contact.