Algorithms for Complex Bipedal Walking

by

Ing. Kamil Dolinský

supervised by *Prof. RNDr. Sergej Čelikovský, CSc.*

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Dissertation

Presented to the *Department of Control Engineering*,
Faculty of Electrical Engineering of
Czech Technical University in Prague
in Partial Fulfillment of the Requirements
for the Degree of

Doctor

in Ph.D. programme
Electrical Engineering and Information Technology
in the branch of study
Control Engineering and Robotics

Czech Technical University in Prague
August 2017

You can't cross the sea merely by standing and staring at the water.

– Rabindranath Tagore

I would like to dedicate this thesis to my parents. Thank you for the faith in me and the never ending support and love during my studies.

Nie je možné preplaviť sa cez oceán len stáním a hľadením na vodu.

– Rabindranath Tagore

Túto prácu by som chcel venovať svojím rodičom. Ďakujem Vám za to, že vo mňa vždy veríte a za Vašu nikdy nekončiacu podporu a lásku počas môjho štúdia.

Acknowledgements

I am very grateful for all the time I could spend on the Department of Control Engineering. I has been a wonderful experience. The study gave me a chance to meet many brilliant people and taught me important lessons. I am thankful for many beautiful moments. It is not possible to list here all the great people I have met there, but I would like to personally thank to those that helped me the most.

I would like to thank my supervisor Sergej Čelikovský for his guidance and support and for giving me an opportunity to study on the Department of Control Engineering. I also deeply appreciate that I had a chance to work for UTIA. It is my pleasure to thank to Noboru Sakamoto for making possible to visit the Aerospace Department of Nagoya University. I am very grateful to Zdeněk Hurák and the members of the AA4CC group for plenty of help during the start of my studies and for many good memories. I also would like to thank to the excellent company in the laboratory, namely Jiří Dostál, Pavel Otta, Radek Beňo and Peter Matisko. I would like to thank my colleagues Milan Anderle and Jaroslav Žoha for insightful advises concerning the development of hardware. And last, but not least, I am indebted to Markéta Jedličková and Dan Martinec for valuable help on the graphical side of the thesis, especially the cover of the thesis and for moral support during finishing.

Czech Technical University in Prague *Kamil Dolinský*
Jun 2017

Declaration

This doctoral thesis is submitted in partial fulfillment of the requirements for the degree of doctor (Ph.D.). The work submitted in this dissertation is the result of my own investigation, except where otherwise stated. I declare that I worked out this thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis. Moreover I declare that it has not already been accepted for any degree and is also not being concurrently submitted for any other degree.

Czech Technical University in Prague *Kamil Dolinský*
Jun 2017

Abstract

This thesis studies the problem of parameter estimation, model identification and state estimation for underactuated bipedal walking robots. Two main results were developed. The first result is a novel identification method suited for this problem. The second result is the extension of existing algorithms for state estimation to the case of the hybrid model of an underactuated walking robot.

The identification method takes advantage of the linear structure of the model with respect to estimated parameters. The resulting estimator is calculated iteratively and maximizes the likelihood of the data. The method was tested on both simulated and experimental data. Simulations were carried out for an underactuated walking robot with a distance meter to measure absolute orientation. Laboratory experiments were carried out on a leg of a laboratory walking robot model equipped with a three-axis accelerometer and gyroscope to measure absolute orientation. The method performs favorably in comparison with other benchmark estimation algorithms and both the simulations and the laboratory experiments confirmed its high potential for the use in identification of underactuated robotic walkers.

The state estimators were applied to estimate the absolute orientation of an underactuated walking robot in the presence of impacts which occur when the leg of the robot hits the ground. The proposed estimation scheme was tested on simulations of a 3-link robot and shows that proposed extensions yields improved estimation performance.

Key words: walking robots, maximum likelihood estimation, identification, state estimation.

Abstrakt

Prezentovaná dizertačná práca sa zaoberá štúdiom problému odhadu parametrov, identifikácie modelu a odhadu stavu pre podaktuované dvojnohé kráčajúce roboty. Boli dosiahnuté dva hlavné výsledky. Prvý výsledok predstavuje novú metódu identifikácie vhodnú pre tento problém. Druhý výsledok je rozšírenie existujúcich algoritmov pre odhad stavu nelineárnych systémov pre prípad hybridného modelu podaktuovaných kráčajúcich robotov.
Navrhnutá identifikačná metóda využíva lineárnu štruktúru modelu vzhľadom k odhadovaným parametrom. Výsledný odhad je počítaný iteračne a maximalizuje vierohodnosť dát. Metóda bola testovaná na simulačných, ale aj experimentálnych dátach. Simulácie boli vytvorené pre príklad robota vybaveného laserovým diaľkomerom pre meranie absolútnej pozície robota. Laboratórne experimenty boli vykonané na nohe laboratórneho prototypu kráčajúceho robota. Robot bol vybavený trojosím akcelerometrom a gyroskopom pre meranie absolútnej orientácie. V porovnaní s klasicky používanými algoritmami sa metóda chová veľmi priaznivo. Simulačné a laboratórne experimenty potvrdili vysoký potenciál navrhnutej metódy pre odhad parametrov podaktuovaných kráčajúcich robotov.
Algoritmy pre odhad stavu nelineárnych systémov boli aplikované na úlohu odhadu absolútnej orientácie podaktuovaného kráčajúceho robota. Navrhnuté rozšírenie rieši problém nárazu nohy robota na zem. Navrhnutý prístup bol testovaný na simuláciách robota. Výsledky ukazujú výrazne vylepšenie odhadu absolútnej orientácie robota.

Kľúčové slová: kráčajúce roboty, metóda maximálnej vierohodnosti, identifikácia, odhad stavu.

Acronyms

AO	Absolute Orientation
DoF	Degree(s) of Freedom
EoM	Equation(s) of Motion
EKF	Extended Kalman Filter
HEKF	Hybrid Extended Kalman Filter
IMU	Inertial Measurement Unit
IRC	Incremental Rotary Coder
LS	Least Squares
MC	Monte Carlo
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation/Estimate
NaN	Not a Number
RLS	Recursive Least Squares
UKF	Unscented Kalman Filter
WLS	Weighted Least squares
WLS-FCD	Weighted Least Squares with Filtered Central Differences

Contents

1 Introduction

1.1 Motivation

The underactuated robotic walking has been studied intensively during last decades. The possibility to use a walking like mechanism for transportation in rough terrain, desire to replace humans in hazardous occupations, possible applications in medicine and other applications make the bipedal walking a promising field that has developed substantially in last decades. Recent developments in this field resulted in increased performance of robotic walkers, shifting from slow quasi-static motions to fast and agile walking and running [Westervelt et al. (2007)]. This resulted in constructing many new walking prototypes among them the famous RABBIT [Chevallereau et al. (2003)], MABEL [Grizzle et al. (2009)], MARLO [Buss et al. (2016)] and other robots which served as testbeds for developed control algorithms. The shift towards the agile walking has been accomplished by developing controllers based on the dynamics of robot instead of heuristics or analysis of static forces. However, utilizing a controller that takes into the account the dynamics of a walking robot requires a mathematical model of the dynamics of the robot. Nowadays efficient means for determining such models are available using classical mechanics of rigid bodies, nevertheless, to obtain the parameters describing the mass distribution, geometry and the friction of the robot joints is still an open problem. The estimation problem is complicated by the interaction of the robot with the walking surface which results in hybrid model of the robot. Additionally, measuring the absolute orientation of an underactuated bipedal walking robot with respect to the ground is a nontrivial problem that cannot be solved separately from the parameter

estimation problem.

Underactuation — an intrinsic property of any agile bipedal walking robot — occurs whenever the foot of the robot is not flat on the ground. This is because the contact point between the foot and the ground can be modeled as a pivot point. However, this pivot point is not directly actuated and it is not possible to place a motor to actuate robot in that point, see the fig.1.1 and note the underactuated angle q_1. All the remaining joints of the robot can be actuated directly using motors. This fact is closely related to the problem of model identification. To identify a model of an underactuated walking robot from measurements obtained during walking, an information about the absolute orientation of the robot is required. It is possible to measure all the relative angles between the links of robot using optical encoders or encoders based on Hall effect. These provide accurate information about relative angles between the links of the robot. Further, these measurements can be used to obtain information about relative angular velocities and accelerations between links. However, due to the underactuation it is not possible to use an encoder to directly measure the absolute orientation of a walking robot. This information has to be obtained using different sensors. The human sensory system relies mostly on the sight and the inner ear. Therefore two basic concepts of the absolute orientation measurement are analyzed, the first uses a visual feedback providing the distance of the robot to a point on the ground. The second concept uses a gyroscope and an accelerometer. Unfortunately, neither of such sensors usually provides accurate information, but quite on the contrary both approaches to the measurement of the absolute angle are prone to errors. The distance sensor is prone to transformation errors when the distance is transformed to the angle. The combination of gyroscopes contain bias errors and the accelerometers are sensitive to vibrations. This problem further complicates the estimation procedure. Such an unavailability of the absolute orientation measurements and the hybrid nature of the robot model are the main differences which distinguish the problem of identifying the model of walking robot from a similar problem of identification of robotic manipulators [Janot et al. (2014a)], [Janot et al. (2014b)], [Janot et al. (2014c)].

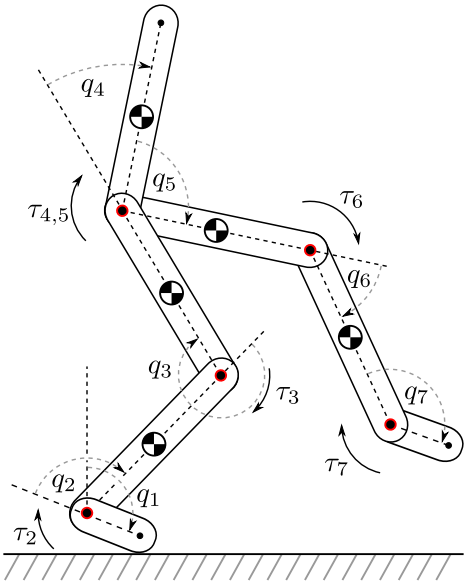


Figure 1.1 – Schematics of a walking robot.

1.2 State of the art

The problems of model identification and parameter estimation for the case of walking robots resembles to certain extent the identification and parameter estimation of models for robotic manipulators. There are several features that both problems share. The key principle inherent in both problems is the possibility to relate the torques of the joint motors with functions of configuration angles, associated angular velocities and accelerations in linear manner with respect to parameters of robot model. A robotic manipulator is, however, a fully actuated mechanical chain and therefore all of the configuration angles — including the absolute orientation angle — can be accurately measured using encoders. Associated angular velocities and accelerations can be estimated using filtered finite differences. On the contrary, the underactuation of walking robots resulting in the unavailability of encoder readings of the absolute orientation angle and the necessity of measuring the angle using sensors susceptible to errors complicates the estimation procedure. As a consequence, only

in a special case when all the torques are available and all the configuration angles are measurable with high precision the model of dynamics of a bipedal walking robot can be identified using the identification methods for robotic manipulators. Under such special circumstances both estimation problems are reduced to a linear regression problem and the method of weighted-least-squares (WLS) can be used to estimate the parameters of robot. Such a special case can be obtained by building a special walking platform. However, generally bipedal walking robots posses additional problems that complicate the use of methods tailored for identification of manipulator robots and therefore additional problems have to be solved for successful application of regression methods to parameter estimation of bipedal robots.

For early works exploiting the linear relationship in parameters between the torques and positional data see [An et al. (1985)] and [Armstrong (1987)]. The work [An et al. (1985)] studies the identifiability of a manipulator robot without full force/torque sensing. More specifically, it proposes to use ridge regression to cope with unidentifiable parameters. The work [Armstrong (1987)] studies the design of optimal excitation trajectories for manipulator robot. More recent references on the topic of trajectory optimization for manipulator robots include [Gautier and Khalil (1991)], [Swevers et al. (1996)], [Swevers et al. (1997)], [Olsen et al. (2002)], [Capisani et al. (2007)] and many others. The problem of identifiability of the parameters of the dynamical model in the case of walking robots is similar, however, if an appropriate parametrisation of robot is used then all of the parameters determining the dynamics are identifiable, provided the robot is sufficiently excited. The idea of estimating manipulator robot angular velocities and accelerations using numerical derivatives and exploiting the least-squares method was studied and experimentally verified also in [Poignet and Gautier (2000)], [Gautier and Poignet (2001)], [Gautier et al. (2013a)]. To cope with potential correlation induced by closed-loop control works [Janot et al. (2014b)] and [Janot et al. (2014c)] propose the use of the Instrumental variable method. Estimating the parameters of manipulator robots only from torque measurements was studied in [Gautier et al. (2008)] and [Gautier et al. (2013a)]. Estima-

tion of manipulator robot parameters online was studied in [Poignet and Gautier (2000)], [Gautier and Poignet (2001)], [Gautier et al. (2013b)]. Many additional important topics related to the manipulator robots identification were obtained, however, few of them can be directly applied to identify bipedal walking robots.

Successful identification of walking robots was reported in [El Yaaqoubi and Abba (2009)], where authors identify the walking robot RABBIT — both the dynamical model and the ground model. In the work [Park et al. (2011)] the authors estimate the parameters of robot MABEL. These works are both based on identification using a walking platform. Further, the identification is based on series of special experiments to identify the robot part by part.

1.3 Aim of the Thesis

The aim of the thesis is to study the problem of the model identification and state estimation of underactuated bipedal walking robots. The main problems to be solved are:

I. How to estimate the parameters of an underactuated walking robot model when direct measurement of AO is not available.

II. How to exploit the linear structure of the walking robot model with respect to the parameters when the measurements related to the AO are noisy.

III. How to online estimate the AO angle from sensors typically available for the walking robots.

Both problem I. and II. are solved in the chapter 3. Problem III. is studied and solved in chapter 4.

The issues I. and II. are solved by a novel method based on the method of ML. The method takes advantage of the linear structure of the robot

model with respect to the parameters. The noise in the measurements related to the AO of the robot are smoothed iteratively while the quality of the estimate of the parameters increases. These issues are studied on two case studies. First study is a based on simulations of a three link walking robot which uses a laser distance measurement to measure the underactuated angle. This investigation shows that the approach developed in this article can deal with the measurement errors and performs favorably in comparison with other common estimation methods. The other study is a laboratory experiment where the parameters of a leg of an underactuated walking robot are estimated and the underactuated angle is measured using a 3-axis accelerometer and gyroscope `LSM9DS1` produced by the company STMicroelectronics. The experiment shows that assumptions of the method proposed in the chapter 3 are valid and that it can be used in real application.

Problem III. — studied in chapter 4 — is solved by extending the well-known state estimation algorithms for the case of the hybrid model of the walking robot. The extended estimators yield excellent performance, even in the presents of moderate errors in the parameters of the robot model used for the estimator design. The estimators were tested on simulation study dealing with feedback control of a 3-DoF walking robot.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 covers necessary background in modeling of walking robots. It contains the derivation of a complex model containing the legs and the torso. The matrices of the model can be found in the Appendix. This model can be used to derive any other models used in the thesis. Moreover, the classical estimation algorithms — discrete EKF and UKF — are presented in this chapter. The discrete EKF and UKF algorithms will be extended to be applicable in the hybrid model of walking robots.

The main contribution of the author is described in the Chapter 3 and

the Chapter 4.

Chapter 3 deals with the offline estimation of parameters of walking robots. The proposed procedure is general and is explained on a 5-DoF walking robot. However, for the sake of clarity a simulation example is presented on the 3-link model of robot. Further, this chapter contains an experimental study of the performance of this method. The experiments were carried out on the leg of the prototype walking robot.

Chapter 4 presents the application of the classical estimation algorithms — the discrete EKF and the discrete UKF — to the hybrid model of underactuated walking robot. The extension deals with including the impact map in the estimator and using it to improve the estimation of the absolute orientation of the walking robots. The results are analyzed using Monte Carlo simulations and show excellent performance of the estimators. Thanks to the incorporating the impact map into the estimation algorithm the improved performance of the state estimators helps recover the stable behavior of the closed-loop robot controller.

Chapter 5 concludes the thesis, sums up the contribution of the author and describes some open problem connected to studied topic.

The Appendix includes the matrices of the 5-link model of a walking robot and additional equations connected to the robot model.

2 Preliminaries

2.1 Model of a planar bipedal walking robot

The model for the bipedal robot is depicted on fig. 2.1 with two different options of configuration angles. The robot consists of rigid links connected via rotary joints. Each joint is actuated. Two notable features can be observed — the fact that the model is planar and the fact that the robot model has point-feet. The choice of point-feet is a common feature of underactuated walking robots, see [Westervelt et al. (2007)] and [Chevallereau et al. (2003)], [Grizzle et al. (2009)]. The choice of the point feet is due to simplification of the gait analysis. Walking can be defined as periodic switching of two phases of the robot — a swing phase and an impact phase. Swing phase is occurring when one leg is in the air and the other is on the ground. The leg that is in the air is called the swing leg and the leg that is on the ground is called the stance leg. It is assumed that the stance leg does not slip, nor bounce of the ground. The impact phase occurs when both legs are in contact with the walking surface. Thanks to the point-feet the robot is underactuated during the whole swing phase. This fact simplifies the analysis of the robot gait and it is the main reason for the choice of point-feet model of the robot. Due to underactuation it is not possible to easily measure the AO of the robot. As mentioned before this fact complicates the control and parameter estimation of the robot. However, because the model is planar it is not capable of 3D motion and if such a model would be built it would not be able to walk without a platform stabilizing the robot in the lateral direction. The platform itself can be used to measure the AO. Nevertheless, the planar

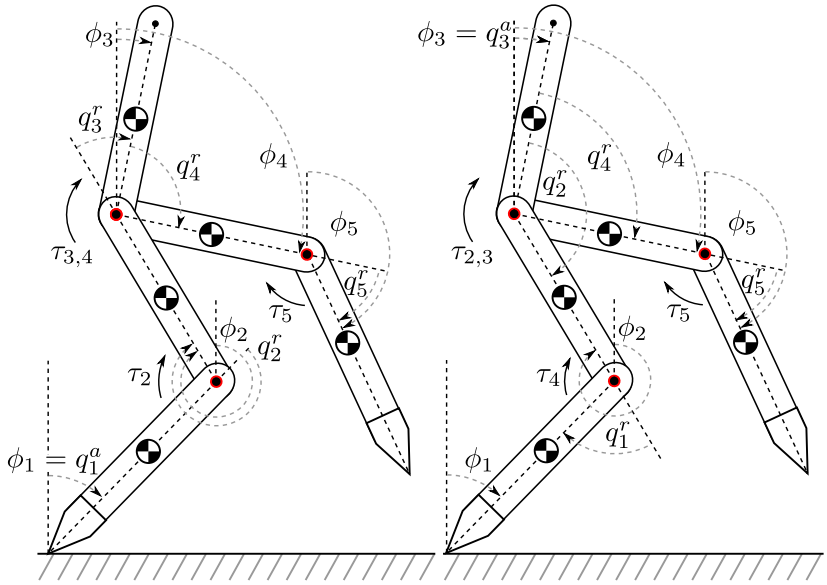


Figure 2.1 – Schematics of a 5-link robot with 5Dof. Left: AO angle q_1^a with respect to the stance leg. Right: AO angle q_3^a with respect to the torso.

robot is an excellent testbed for testing identification algorithms that do not utilize the information about the AO from the platform. Further, planar model includes all the principal challenges inherent in the problem of the identification of the underactuated walking robots. Therefore, solving the problem of AO estimation and parameter estimation when AO is not directly measured is done in 2D as a step towards 3D walking. During 3D walking robots which would walk freely without the need for support in the lateral direction. The figure 2.1 describes two special sets of configuration angles. The AO of the robot model depicted in the left part of the figure 2.1 is defined as the angle between the gravitational vector and the stance leg. The robot model depicted in the right part of the figure 2.1 is defined as the angle between the gravitational vector and the torso of the robot. These two ways of defining the AO will be used often in this work and their use is closely connected with the type and the location of sensors for the measurements of the absolute orientation.

2.2 Swing phase

During the swing phase the stance leg of the robot is rotating around the pivot point which is the point of contact between the stance leg and the walking surface. Planar bipedal walking robot can be modeled as a pinned planar open kinematic chain. The chain consisting of n_r rigid links can be modeled as a mechanical system with n_r degrees of freedom. Its motion can be described by one absolute angle q^a and its associated velocity $\dot{q}^a$ and $n_r - 1$ relative angles q_i^r and corresponding $n_r - 1$ angular velocities $\dot{q}_i^r$. All these angles at a particular time t form a vector of configuration angles $\boldsymbol{q}(t)$ and together with the vector of angular velocities $\dot{\boldsymbol{q}}(t)$ they constitute the state vector $\boldsymbol{x}(t)$. Torques corresponding to each degree of freedom are included in the vector $\boldsymbol{\tau}(t)$, vector $\boldsymbol{u}(t)$ comprises the actuator torques. Summarizing,

$$\boldsymbol{q}(t) = [q^a, \boldsymbol{q}^{rT}]^T, \tag{2.1}$$
$$\dot{\boldsymbol{q}}(t) = [\dot{q}^a, \dot{\boldsymbol{q}}^{rT}]^T, \tag{2.2}$$
$$\boldsymbol{x}(t) = [\boldsymbol{q}^T(t),\ \dot{\boldsymbol{q}}^T(t)]^T, \tag{2.3}$$
$$\boldsymbol{\tau}(t) = [\tau_1(t), \ldots, \tau_{n_r}(t)]^T, \tag{2.4}$$
$$\boldsymbol{u}(t) = [u_1(t), \ldots, u_{n_{r-1}}(t)]^T. \tag{2.5}$$

Kinematic chains can be modeled by the Euler-Lagrange's equations of motion [Landau and Lifshitz (1976)]

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathscr{L}}{\partial \boldsymbol{q}} = \boldsymbol{\tau}, \tag{2.6}$$

where $\mathscr{L}$ denotes the Lagrangian, which equals the difference between the kinetic and the potential energy of considered mechanical system, that is,

$$\mathscr{L} = K(\boldsymbol{q}, \dot{\boldsymbol{q}}) - V(\boldsymbol{q}). \tag{2.7}$$

The kinetic energy of the system can be calculated as the sum of the kinetic energies of each link,

$$K(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^{n_r} K_i(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{2.8}$$

The kinetic energy of the i-th link is

$$K_i = \frac{1}{2} m_i v_i^2 + \frac{1}{2} I_i \dot{\phi}_i^2. \tag{2.9}$$

Symbol m_i denotes the mass of the i-th link, inertia of the i-th link is denoted as I_i and ϕ_i denotes the absolute orientation of the i-th link defined with respect to the beginning of the i-th link, positive in the clock-wise direction. Angles ϕ_i expressed in configuration angles q_i for the particular configuration angles depicted in the figure 2.1 are given in the Appendix. Velocity vector $\boldsymbol{v}_i$ of the the i-th link coordinates is defined as

$$\boldsymbol{v}_i = \left[\frac{dx_{c_i}}{dt}, \frac{dy_{c_i}}{dt} \right] \tag{2.10}$$

and the square of the velocity vector is equal to

$$v_i^2 = \left(\frac{dx_{c_i}}{dt} \right)^2 + \left(\frac{dy_{c_i}}{dt} \right)^2. \tag{2.11}$$

The location of the CoG in Cartesian coordinates for i-th link is denoted as x_{c_i} and y_{c_i} and is equal to

$$x_{c_i} = l_{c_i} \sin(\phi_i) + \sum_{j=1}^{i-1} l_j \sin(\phi_j) \tag{2.12}$$

$$y_{c_i} = l_{c_i} \cos(\phi_i) + \sum_{j=1}^{i-1} l_j \cos(\phi_j). \tag{2.13}$$

The potential energy is given as

$$V(\boldsymbol{q}) = \sum_{i=1}^{n_r} V_i(\boldsymbol{q}) = \sum_{i=1}^{n_r} m_i g h_i. \tag{2.14}$$

Constant g is the gravity acceleration constant and h_i denotes the height from the ground of the i-th link. Note that the calculated Kinetic energy will have the following structure

$$K(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \mathbf{D}(\boldsymbol{q})\dot{\boldsymbol{q}},\ \mathbf{D}(\boldsymbol{q}) = \mathbf{D}^T(\boldsymbol{q}) \succ 0 \tag{2.15}$$

and the equations (2.6) will assume the following form [Westervelt et al. (2007)]

$$\mathbf{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) + \mathbf{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} = \mathbf{B}\boldsymbol{u}, \tag{2.16}$$

where

$$\mathbf{G}(\boldsymbol{q}) = \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}}, \quad \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{\partial}{\partial \boldsymbol{q}}(\mathbf{D}(\boldsymbol{q})\dot{\boldsymbol{q}}) - \frac{1}{2}(\frac{\partial}{\partial \boldsymbol{q}}(\mathbf{D}(\boldsymbol{q})\dot{\boldsymbol{q}}))^T. \tag{2.17}$$

Matrix $\mathbf{D}(\boldsymbol{q})$ is called the inertia matrix, $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the matrix of Coriolis and centrifugal forces, vector $\mathbf{G}(\boldsymbol{q})$ describes gravity effects and vector $\mathbf{F}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contains friction model terms. Matrix $\mathbf{B}$ describes how the actuator torques $\boldsymbol{u}$ are generating torques $\boldsymbol{\tau}$. For the case of 5-link robot it is calculated as

$$\mathbf{B} = \left[\frac{\partial(\phi_2 - \phi_1)}{\partial \boldsymbol{q}}, \quad \frac{\partial(\phi_3 - \phi_2)}{\partial \boldsymbol{q}}, \quad \frac{\partial(\phi_3 - \phi_4)}{\partial \boldsymbol{q}}, \quad \frac{\partial(\phi_4 - \phi_5)}{\partial \boldsymbol{q}}\right]. \tag{2.18}$$

The associated state space model can be found using (2.3) and (2.16) as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{2.19}$$

Table 2.1 – Parameters of the n_r-link robot

$l_1, \ldots, l_{n_r}$	length of $1^{\text{st}}, \ldots n_r{}^{\text{th}}$ link	$[m]$
$l_{c_1}, \ldots, l_{c_{n_r}}$	center of gravity of $1^{\text{st}}, \ldots n_r{}^{\text{th}}$ link	$[m]$
$m_1, \ldots, m_{n_r}$	mass of $1^{\text{st}}, \ldots n_r{}^{\text{th}}$ link	$[kg]$
$I_1, \ldots, I_{n_r}$	inertia of $1^{\text{st}}, \ldots n_r{}^{\text{th}}$ link	$[kg.m^2]$
g	gravitational acceleration	$[m.s^{-2}]$

where

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \mathbf{D}(\boldsymbol{q})^{-1}(-\mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \mathbf{G}(\boldsymbol{q}) - \mathbf{F}(\boldsymbol{q},\dot{\boldsymbol{q}})) \end{bmatrix}, \tag{2.20}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}(\boldsymbol{q})^{-1}\mathbf{B} \end{bmatrix}. \tag{2.21}$$

For matrices $\mathbf{D}$, $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{B}$ corresponding to various models of planar bipedal walking robots, see the Appendix. Matrices of the model can be parametrized by physical parameters of the robot which are listed in tab. 2.1. An example of definitions of lengths of robot links and the locations of Centers of Gravity (CoG) is depicted in the left part of the figure 2.2.

2.3 Impact phase

An impact occurs when the swing leg touches the walking surface. When the walking surface is rigid, the duration of impact is very short. It is common to approximate it as being instantaneous. This approximation leads to replacing the ground reaction forces by impulses, resulting in a discontinuity of velocity components of the robot state. The result of the impact model are the new initial conditions from which the single support model evolves until the next impact [Westervelt et al. (2007)]. Depending on the assumptions several impact models can be constructed [Babitsky (1998)], [Brogliato (1999)], [Grizzle et al. (2001)] and [Hurmuzlu and

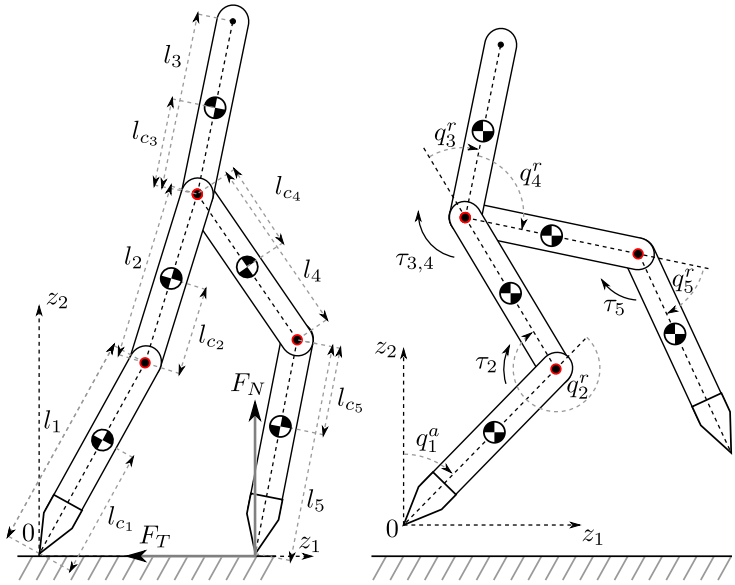


Figure 2.2 – Left: link lengths, CoG locations and ground reaction forces. Right: schematics of an unpinned 5-link walking robot with 7 DoF.

Marghitu (1994)]. As long as the assumptions are valid all of them can be used to derive the impact map. The approach taken in [Grizzle et al. (2001)] was adopted in this work for several different models of robots. The motion of the robot is analyzed only for the case that the contact of the swing leg with the ground results in no rebound and no slipping of the swing leg, and the support leg naturally lifting from the ground with no interaction [Hurmuzlu and Marghitu (1994)]. The basic premises in [Hurmuzlu and Marghitu (1994)] are that

1. the impact takes place over an infinitesimally small period of time,
2. the external forces can be represented by impulses,
3. impulsive forces may result in an instantaneous change in velocities of the generalized coordinates, but the positions remain continuous,
4. the torques supplied by the actuators are not impulsional.

To determine the impact forces and the new initial conditions for the single support phase the so-called unpinned model — depicted in the right part of the figure 2.2 — of the robot is necessary. To obtain the unpinned model, the vector of configuration angles $\boldsymbol{q}$ is extended with Cartesian coordinates z_1, z_2 of an arbitrary point on the robot. For convenience, this point is chosen to be identical with the end tip of the stance leg. The resulting model has 7 DoF and can be written as

$$\mathbf{D}_e(\boldsymbol{q}_e)\ddot{\boldsymbol{q}}_e + \mathbf{C}_e(\boldsymbol{q}_e, \dot{\boldsymbol{q}}_e)\dot{\boldsymbol{q}}_e + \mathbf{G}_e(\boldsymbol{q}_e) = \mathbf{B}\tau + \delta\boldsymbol{F}^{\text{ext}} \tag{2.22}$$

where $\boldsymbol{q}_e = [\boldsymbol{q}^T, z_1, z_2]^T$ stands for the extended configuration vector. The model 2.22 is integrated over the duration of the impact to obtain [Hurmuzlu and Marghitu (1994)]

$$\mathbf{D}_e(\boldsymbol{q}_e)\left(\dot{\boldsymbol{q}}_e^+ - \dot{\boldsymbol{q}}_e^-\right) = \boldsymbol{F}^{\text{ext}} \tag{2.23}$$

where $\boldsymbol{F}^{\text{ext}} = \int_{t^-}^{t^+} \delta\boldsymbol{F}^{\text{ext}}(\tau)d\tau$ is the integral of the contact impulse over the duration of the impact, $\dot{\boldsymbol{q}}_e^-$ is the velocity just before the impact and $\dot{\boldsymbol{q}}_e^+$ is the velocity after the impact. The positions do not change therefore $\boldsymbol{q}_e^+ = \boldsymbol{q}_e^-$.

In order to determine the vectors $\boldsymbol{q}_e^+$ and $\boldsymbol{F}^{\text{ext}}$ additional equations that describe the contact forces at the contact points are required. The first contact point is between the ground and the supporting leg. As the supporting leg is assumed to detach from the ground with no interaction the forces acting on this leg are zero. Therefore, the vector $\boldsymbol{F}^{\text{ext}}$ will be composed only of the forces acting at the end of the swing leg [Grizzle et al. (2001)]. The position of the end point of the swing leg will be denoted as Υ. For the case of 5-link robot with absolute angle defined w.r.t the stance leg or the torso — as depicted on the figure 2.1 — and parametrized as depicted in the left part of the figure 2.2 the vector Υ is given in the Appendix. The vector of external forces is

$$\boldsymbol{F}^{\text{ext}} = \mathbf{E}^T \begin{bmatrix} F_T \\ F_N \end{bmatrix}, \tag{2.24}$$

where F_T, F_N denotes the tangent and normal forces, respectively, acting at the end of the swing leg. The matrix $\mathbf{E}$ is defined as

$$\mathbf{E} := \frac{\partial \Upsilon}{\partial q_e}. \tag{2.25}$$

Thus, the system of seven equations 2.23 contains nine unknowns, $\dot{\boldsymbol{q}}_e^+$ and F_T, F_N, vector $\dot{\boldsymbol{q}}_e^-$ is known since $\dot{\boldsymbol{q}}_e^- = [\dot{\boldsymbol{q}}^{-T}, \dot{z}_1, \dot{z}_2]$ and $\dot{z}_1 = 0,\ \dot{z}_2 = 0$, since the supporting leg acts as a pivot during the swing phase. Additional set of two equations can be obtained from the assumption that the swing leg does not rebound nor slip at impact, i.e. $(d/dt)\Upsilon(q_e) = \frac{\partial \Upsilon}{\partial \boldsymbol{q}_e}\dot{\boldsymbol{q}}_e = 0$, thus

$$\mathbf{E}\dot{\boldsymbol{q}}_e^+ = 0. \tag{2.26}$$

As a consequence, the following system of equations

$$\begin{bmatrix} \mathbf{D}_e & -\mathbf{E} \\ \mathbf{E} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_e^+ \\ \boldsymbol{F}^{\text{ext}} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_e \dot{\boldsymbol{q}}_e^- \\ \mathbf{0} \end{bmatrix} \tag{2.27}$$

linear in the unknowns $\dot{\boldsymbol{q}}_e^+$ and $\boldsymbol{F}$ is obtained. Note that $\boldsymbol{F}^{\text{ext}} = (F_T,\ F_N)^T$. The solvability of the (2.27) requires the invertibility of the matrix on the left hand side. The invertibility of the left hand side matrix follows from the fact that $\mathbf{D}_e$ is positive definite and $\mathbf{E}$ has full rank [Grizzle et al. (2001)]. The solution of the (2.27) yields the vector $\dot{\boldsymbol{q}}_e^+ = [\dot{\boldsymbol{q}}^+, \dot{\boldsymbol{z}}]$ and values $\dot{\boldsymbol{q}}^+$ should be used to reinitialize the model. However, before reinitialisation a change of coordinates is necessary as the former support leg is now swing leg and vice versa. The state vector for new swing phase will be $\boldsymbol{x} = \left[\tilde{\boldsymbol{q}}^T,\ \dot{\tilde{\boldsymbol{q}}}^T\right]^T$. Where $\tilde{\boldsymbol{q}}$ and $\dot{\tilde{\boldsymbol{q}}}$ denotes the angles and associated angular velocities after the impact and relabeling. This operation will be denoted as

$$\mathbf{x}^+ = \boldsymbol{\Delta}(\mathbf{x}^-). \tag{2.28}$$

The extended matrix of the 7DoF model with AO defined with respect to the stance leg required for impact calculation is given in the Appendix.

The relabeling maps for the 5-link with AO defined w.r.t the stance leg or the torso are given in the Appendix.

2.4 Discrete model of the robot

The prediction of state and measurements between the sampling instant is based on the discrete model of the robot dynamics and measurements

$$\boldsymbol{x}(t_{l+1}) = \boldsymbol{f}_d(\boldsymbol{x}(t_l), \boldsymbol{u}(t_l)), \tag{2.29}$$

$$\boldsymbol{y}(t_k) = \boldsymbol{h}_d(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)). \tag{2.30}$$

The vector field $\boldsymbol{f}_d$ can be calculated using explicit integration methods. Using the Euler's method results in $\boldsymbol{f}_d$ given as

$$\boldsymbol{f}_{eul} = \boldsymbol{x}(t_l) + T_i(\boldsymbol{f}(\boldsymbol{x}(t_l)) + \boldsymbol{g}(\boldsymbol{x}(t_l))\boldsymbol{u}(t_l)), \tag{2.31}$$

where T_i is the integration duration. Euler's integration method is very simple. The simplicity of the method results in lower precision of the method. Therefore, this method has to be applied several times between the measurement samples. Due to these reasons a more sophisticated integration method — the method of Rung-Kutta of fourth order — has been tested. The vector field $\boldsymbol{f}_d$ for this method is given as

$$\boldsymbol{f}_{rk4} = \boldsymbol{x}(t_l) + \frac{\boldsymbol{k}_1}{6} + \frac{\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{3} + \frac{\boldsymbol{k}_4}{6}, \tag{2.32}$$

where, provided $\boldsymbol{u}(t_l)$ is held constant during one sample,

$$\begin{aligned}
\boldsymbol{k}_1 &= T_i(\boldsymbol{f}(\boldsymbol{x}(t_l)) + \boldsymbol{g}(\boldsymbol{x}(t_l))\boldsymbol{u}(t_l)), \\
\boldsymbol{k}_2 &= T_i(\boldsymbol{f}(\boldsymbol{x}(t_l) + \frac{k_1}{2}) + \boldsymbol{g}(\boldsymbol{x}(t_l) + \frac{k_1}{2})\boldsymbol{u}(t_l)), \\
\boldsymbol{k}_3 &= T_i(\boldsymbol{f}(\boldsymbol{x}(t_l) + \frac{k_2}{2}) + \boldsymbol{g}(\boldsymbol{x}(t_l) + \frac{k_2}{2})\boldsymbol{u}(t_l)), \\
\boldsymbol{k}_4 &= T_i(\boldsymbol{f}(\boldsymbol{x}(t_l) + k_3) + \boldsymbol{g}(\boldsymbol{x}(t_l) + k_3)\boldsymbol{u}(t_l)).
\end{aligned} \tag{2.33}$$

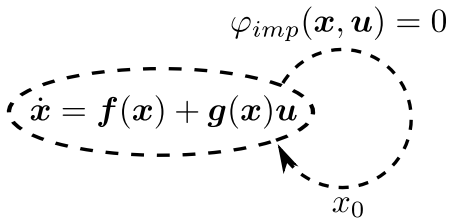


Figure 2.3 – Robot model with impact.

Every time the robot's swing leg touches the ground an impact occurs. To model the impact mathematically it only means that the robot model is reinitialized with new initial conditions whenever the impact condition $\varphi_{imp}(\boldsymbol{x}, \boldsymbol{u}) = 0$ as shown on figure 2.3. The condition is usually defined as the height of the end tip of the swing leg.

2.5 Estimation algorithms

2.5.1 Discrete Extended Kalman Filter

The algorithm of the Extended Kalman Filter (EKF) can be used to estimate the state of a nonlinear dynamical system [Simon (2006)]. The algorithm is composed of two steps: the time update step and the measurement update step. During the time update a one step ahead prediction of the state $\hat{\boldsymbol{x}}(t_{k+1}|t_k)$ is calculated. This prediction is updated during measurement update once a new measurement is available and results in the corrected estimate $\hat{\boldsymbol{x}}(t_{k+1}|t_{k+1})$. The stochastic discrete model of the walking robot can be written as

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{f}_d(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), \boldsymbol{w}(t_k)), \tag{2.34}$$
$$\boldsymbol{y}(t_k) = \boldsymbol{h}_d(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), \boldsymbol{v}(t_k)), \tag{2.35}$$
$$\boldsymbol{w}(t_k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_Q), \tag{2.36}$$
$$\boldsymbol{v}(t_k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_R). \tag{2.37}$$

Vector $\boldsymbol{w}(t_k)$ is a Gaussian random process representing disturbances acting on the modeled system. Vector $\boldsymbol{v}(t_k)$ is a Gaussian random process

representing the measurement noise. Both $\boldsymbol{w}(t_k)$ and $\boldsymbol{v}(t_k)$ are uncorrelated with each other and their past samples. Matrix $\boldsymbol{\Sigma}_Q$ denotes the covariance matrix of the random disturbances $\boldsymbol{w}(t_k)$ acting on the system and the matrix $\boldsymbol{\Sigma}_R$ stands for covariance of measurement noise $\boldsymbol{v}(t_k)$. The algorithm of the EKF can be summarized in following steps.

1. The EKF filter is initialized with

$$\hat{\boldsymbol{x}}(t_0) = \mathbb{E}[\boldsymbol{x}(t_0)] \tag{2.38}$$
$$\mathbf{P}(t_0) = \mathbb{E}[(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))^T] \tag{2.39}$$

2. For $k = 1 \dots N$ carry out following steps.

   (a) Calculate Jacobians associated with linearisation of the dynamic equation of robot.

$$\mathbf{F}(t_k) = \left.\frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_k) \\ \boldsymbol{w}=\mathbf{0}}}, \quad \mathbf{L}(t_k) = \left.\frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_k) \\ \boldsymbol{w}=\mathbf{0}}} \tag{2.40}$$

   (b) Time update: perform the covariance update and state prediction.

$$\mathbf{P}(t_{k+1}|t_k) = \mathbf{F}(t_k)\mathbf{P}(t_k|t_k)\mathbf{F}(t_k)^T + \mathbf{L}(t_k)\boldsymbol{\Sigma}_Q\mathbf{L}(t_k)^T \tag{2.41}$$
$$\hat{\boldsymbol{x}}(t_{k+1}|t_k) = \boldsymbol{f}_d(\hat{\boldsymbol{x}}(t_k|t_k), \boldsymbol{u}(t_k), \mathbf{0}) \tag{2.42}$$

   (c) Calculate Jacobians associated with linearisation of the robot's measurement model.

$$\mathbf{H}(t_{k+1}) = \left.\frac{\partial \boldsymbol{h}_d}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_{k+1}|t_k) \\ \boldsymbol{w}=\mathbf{0}}}, \quad \mathbf{M}(t_{k+1}) = \left.\frac{\partial \boldsymbol{h}_d}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_{k+1}|t_k) \\ \boldsymbol{w}=\mathbf{0}}} \tag{2.43}$$

   (d) Measurement update: calculate the Kalman gain and the current

measurement estimate

$$\mathbf{K}(t_{k+1}) = \mathbf{P}(t_{k+1}|t_k)\mathbf{H}(t_{k+1})^T(\mathbf{H}(t_{k+1})\mathbf{P}(t_{k+1}|t_k)\mathbf{H}(t_{k+1})^T + \mathbf{M}(t_{k+1})\mathbf{\Sigma}_R\mathbf{M}(t_{k+1})^T)^{-1}, \tag{2.44}$$

$$\hat{\boldsymbol{y}}(t_{k+1}) = \boldsymbol{h}_d(\hat{\boldsymbol{x}}(t_{k+1}|t_k), \boldsymbol{u}(t_{k+1}), \mathbf{0}). \tag{2.45}$$

Perform the correction of the covariance and state estimates using new measurement.

$$\hat{\boldsymbol{x}}(t_{k+1}|t_{k+1}) = \hat{\boldsymbol{x}}(t_{k+1}|t_k) + \mathbf{K}_{k+1}(\boldsymbol{y}(t_{k+1}) - \hat{\boldsymbol{y}}(t_{k+1})) \tag{2.46}$$

$$\mathbf{P}(t_{k+1}|t_{k+1}) = (\mathbb{1} - \mathbf{K}(t_{k+1})\mathbf{H}(t_{k+1}))\mathbf{P}(t_{k+1}|t_k) \tag{2.47}$$

2.5.2 Discrete Unscented Kalman Filter

In comparison with the EKF, the UKF does not require the calculation of the linearized model of the robot. The algorithm of UKF can be summarized in the following steps.

1. The UKF is initialized by following estimates

$$\hat{\boldsymbol{x}}(t_0) = \mathbb{E}[\boldsymbol{x}(t_0)], \tag{2.48}$$

$$\mathbf{P}(t_0) = \mathbb{E}[(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))^T]. \tag{2.49}$$

   For $k = 1 \dots N$ carry out following steps.
2. Time update.
   (a) Generate sigma points for one-step-ahead state prediction:

$$\boldsymbol{x}^{(i)}(t_k) = \hat{\boldsymbol{x}}(t_k|t_k) + \tilde{\boldsymbol{x}}^{(i)}(t_k) \qquad i = 1, \dots, 2n \tag{2.50}$$

$$\tilde{\boldsymbol{x}}^{(i)}(t_k) = \left(\sqrt{n\mathbf{P}(t_k|t_k)}\right)_i^T \qquad i = 1, \dots, n \tag{2.51}$$

$$\tilde{\boldsymbol{x}}^{(n+i)}(t_k) = -\left(\sqrt{n\mathbf{P}(t_k|t_k)}\right)_i^T \qquad i = n+1, \dots, 2n \tag{2.52}$$

   (b) Propagate sigma points using the robot model (2.29), that is

$$\boldsymbol{x}^{(i)}(t_{k+1}) = \boldsymbol{f}_d(\hat{\boldsymbol{x}}^{(i)}(t_k), \boldsymbol{u}(t_k)). \tag{2.53}$$

(c) Calculate one-step-ahead prediction of the state as

$$\hat{\boldsymbol{x}}(t_{k+1}|t_k) = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{x}^{(i)}(t_{k+1}). \tag{2.54}$$

(d) Estimate the covariance of the one-step-ahead state estimate

$$\mathbf{P}(t_{k+1}|t_k) = \frac{1}{2n}\sum_{i=1}^{2n} \left(\left(\boldsymbol{x}^{(i)}(t_{k+1}) - \hat{\boldsymbol{x}}(t_{k+1}|t_k)\right) \left(\boldsymbol{x}^{(i)}(t_{k+1}) - \hat{\boldsymbol{x}}(t_{k+1}|t_k)\right)^T \right) + \boldsymbol{\Sigma}_Q \tag{2.55}$$

3. Measurement update.

(a) Generate sigma points for the estimation of output:

$$\boldsymbol{x}^{(i)}(t_{k+1}|t_k) = \hat{\boldsymbol{x}}(t_{k+1}|t_k) + \tilde{\boldsymbol{x}}^{(i)}(t_{k+1}) \quad i = 1, \ldots, 2n \tag{2.56}$$

$$\tilde{\boldsymbol{x}}^{(i)}(t_{k+1}) = \left(\sqrt{n\mathbf{P}(t_{k+1}|t_k)}\right)_i \quad i = 1, \ldots, n \tag{2.57}$$

$$\tilde{\boldsymbol{x}}^{(n+i)}(t_{k+1}) = -\left(\sqrt{n\mathbf{P}(t_{k+1}|t_k)}\right)_i \quad i = 1, \ldots, n \tag{2.58}$$

(b) Propagate sigma points using the robot measurement model

$$\boldsymbol{y}^{(i)}(t_{k+1}) = \boldsymbol{h}_d(\boldsymbol{x}^{(i)}(t_{k+1}), \boldsymbol{u}(t_{k+1})). \tag{2.59}$$

(c) Calculate current estimate of the outputs as

$$\hat{\boldsymbol{y}}(t_{k+1}) = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{y}^{(i)}(t_{k+1}). \tag{2.60}$$

(d) Estimate the covariance of the predicted measurements

$$\mathbf{P}_y(t_{k+1}) = \frac{1}{2n}\sum_{i=1}^{2n} \left(\left(\boldsymbol{y}^{(i)}(t_{k+1}) - \hat{\boldsymbol{y}}(t_{k+1})\right) \left(\boldsymbol{y}^{(i)}(t_{k+1}) - \hat{\boldsymbol{y}}(t_{k+1})\right)^T \right) + \boldsymbol{\Sigma}_R. \tag{2.61}$$

(e) Estimate the cross-covariance between $\hat{\boldsymbol{x}}(t_{k+1}|t_k)$ and $\hat{\boldsymbol{y}}(t_{k+1})$

$$\mathbf{P}_{xy}(t_{k+1}) = \frac{1}{2n}\sum_{i=1}^{2n}\left(\left(\boldsymbol{x}^{(i)}(t_{k+1}) - \hat{\boldsymbol{x}}(t_{k+1}|t_k)\right)\left(\boldsymbol{y}^{(i)}(t_{k+1}) - \hat{\boldsymbol{y}}(t_{k+1})\right)^T\right). \tag{2.62}$$

(f) Perform the measurement update

$$\mathbf{K}(t_{k+1}) = \mathbf{P}_{xy}(t_{k+1})\mathbf{P}_y^{-1}(t_{k+1}), \tag{2.63}$$

$$\hat{\boldsymbol{x}}(t_{k+1}|t_{k+1}) = \hat{\boldsymbol{x}}(t_{k+1}|t_k) + \mathbf{K}(t_{k+1})(\boldsymbol{y}(t_{k+1}) - \hat{\boldsymbol{y}}(t_{k+1})), \tag{2.64}$$

$$\mathbf{P}(t_{k+1}|t_{k+1}) = \mathbf{P}(t_{k+1}|t_k) - \mathbf{K}(t_{k+1})\mathbf{P}_y(t_{k+1})\mathbf{K}(t_{k+1})^T. \tag{2.65}$$

2.5.3 Hybrid Extended Kalman Filter for parameter estimation

It is possible to estimate both the states $\boldsymbol{x}$ and the parameters $\boldsymbol{\beta}$ of the system (2.19) using HEKF. Vector of parameters β can be obtained by reparametrization of the model as will be seen in the following chapter. Unknown robot parameters can be considered as state variables and the state vector can be augmented as

$$\bar{\boldsymbol{x}} = [\boldsymbol{x}^T, \boldsymbol{\beta}^T]^T. \tag{2.66}$$

Robot (2.19)and measurement model $\boldsymbol{h}$ are augmented as

$$\bar{\boldsymbol{x}} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}, \boldsymbol{u}, \bar{\boldsymbol{w}}) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_\beta \end{bmatrix}, \tag{2.67}$$

$$\boldsymbol{y}(t_k) = \bar{\boldsymbol{h}}(\bar{\boldsymbol{x}}(t_k), \bar{\boldsymbol{v}}(t_k)). \tag{2.68}$$

Vectors $\boldsymbol{w}_x \sim \mathcal{N}(0, \boldsymbol{\Sigma}_Q^c)$, $\bar{\boldsymbol{v}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_R^c)$, $\boldsymbol{w}_\beta \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\beta)$. The state vector is propagated between the measurements according to augmented

state equation and the covariance matrix of the estimation error is propagated according to

$$\dot{\mathbf{P}} = \mathbf{AP} + \mathbf{PA} + \mathbf{L}\boldsymbol{\Sigma}_Q\mathbf{L}^T, \tag{2.69}$$

where

$$\mathbf{A} = \left.\frac{\partial \bar{\boldsymbol{f}}}{\partial \bar{\boldsymbol{x}}}\right|_{\hat{\bar{\boldsymbol{x}}}^+(t_{k-1})}, \quad \mathbf{L} = \left.\frac{\partial \bar{\boldsymbol{f}}}{\partial \boldsymbol{w}}\right|_{\hat{\bar{\boldsymbol{x}}}^+(t_{k-1})}.$$

The prior state estimate $\boldsymbol{x}(t_k|t_{k-1})$ and the prior covariance matrix estimate $\mathbf{P}(t_k|t_{k-1})$ are updated at each measurement time t_k

$$\mathbf{K} = \mathbf{P}(t_k|t_{k-1})\mathbf{H}^T(\mathbf{H}\mathbf{P}(t_k|t_{k-1})\mathbf{H}^T + \mathbf{M}\boldsymbol{\Sigma}_R\mathbf{M}^T)^{-1} \tag{2.70}$$

$$\hat{\bar{\boldsymbol{x}}}(t_k|t_k) = \hat{\bar{\boldsymbol{x}}}(t_k|t_{k-1}) + \mathbf{K}(\boldsymbol{y} - \bar{\boldsymbol{h}}(\hat{\bar{\boldsymbol{x}}}(t_k|t_{k-1}))) \tag{2.71}$$

$$\begin{aligned}\mathbf{P}(t_k|t_k) = {} & (\mathbb{1} - \mathbf{KH})\mathbf{P}(t_k|t_{k-1})(\mathbb{1} - \mathbf{KH})^T \\ & + \mathbf{KM}\boldsymbol{\Sigma}_R\mathbf{M}^T\mathbf{K}^T,\end{aligned} \tag{2.72}$$

where the time indices are dropped to improve the readability.

$$\mathbf{H}(t_k) = \left.\frac{\partial \boldsymbol{h}}{\partial \bar{\boldsymbol{x}}}\right|_{\hat{\bar{\boldsymbol{x}}}(t_k|t_{k-1})}, \quad \mathbf{M}(t_k) = \left.\frac{\partial \bar{\boldsymbol{h}}}{\partial \boldsymbol{v}}\right|_{\hat{\bar{\boldsymbol{x}}}(t_k|t_{k-1})}.$$

2.5.4 Recursive Least Squares

The recursive least squares can be implemented as follows.
1. Initialization: Initialize the parameter estimate $\hat{\boldsymbol{\beta}}(0)$ with some initial estimate. Initialize the covariance matrix $\mathbf{P}(0)$ of the parameter estimate $\hat{\boldsymbol{\beta}}(0)$.
2. Estimation: For $t_k = 1 \ldots N$, perform following:
(a) Calculate the estimated robot joint torques using parameters $\hat{\boldsymbol{\beta}}(t_{k-1})$

$$\hat{\boldsymbol{\tau}}(t_k) = \mathbf{Z}(\hat{\boldsymbol{q}}_{fcd}(t_k), \hat{\dot{\boldsymbol{q}}}_{fcd}(t_k), \hat{\ddot{\boldsymbol{q}}}_{fcd}(t_k))\hat{\boldsymbol{\beta}}(t_k), \tag{2.73}$$

where $\mathbf{Z}$ is the design matrix composed of nonlinear regressors consisting of the relative and absolute configuration angles and their first and second derivatives. The $\hat{\boldsymbol{q}}_{fcd}(t_k)$ stands for digitally filtered measurements of $\boldsymbol{q}(t_k)$. Vectors $\hat{\dot{\boldsymbol{q}}}_{fcd}(t_k)$, $\hat{\ddot{\boldsymbol{q}}}_{fcd}(t_k)$ stands for approximations calculated using central differences from the measurements of $\boldsymbol{q}(t_k)$.
(b) Calculate the gain matrix $\mathbf{K}(t_k)$ and update the parameter estimates and the associated covariance matrix

$$\mathbf{K}(t_k) = \mathbf{P}(t_{k-1})\mathbf{Z}^T(t_k)(\mathbf{Z}(t_k)\mathbf{P}(t_{k-1})\mathbf{Z}(t_k) + \mathbf{R})^{-1}, \tag{2.74}$$

$$\hat{\boldsymbol{\beta}}(t_k) = \hat{\boldsymbol{\beta}}(t_{k-1}) + \mathbf{K}(t_k)(\boldsymbol{y}_\tau(t_k) - \hat{\boldsymbol{\tau}}(t_k)), \tag{2.75}$$

$$\begin{aligned}\mathbf{P}(t_k) = {} & (\mathbb{1} - \mathbf{K}(t_k)\mathbf{Z}(t_k))\mathbf{P}(t_{k-1})(\mathbb{1} - \mathbf{K}(t_k)\mathbf{Z}(t_k)) \\ & + \mathbf{K}(t_k)\mathbf{R}(t_k)\mathbf{K}(t_k).\end{aligned} \tag{2.76}$$

Matrix $\mathbf{R} = \mathbf{B}\boldsymbol{\Sigma}_u\mathbf{B}^T$ is of rank 2. Therefore a regularization with some small number is necessary in order to prevent numerical instability of the algorithm.

3 Offline identification for bipedal walking robots

This chapter contains the key results of the thesis. In particular, the following sections describe how it is possible to include inertial measurements or distance measurements in the design matrix of the underlying regression problem. The regression problem is closely connected to the problem of the parameter estimation. This chapter also develops the algorithm which removes the noise contained in these measurements and calculates the parameter estimates. The chapter is organized as follows. First, the problem of parameter estimation is introduced. Section 3.1 describes the estimation of relative velocities that are required in the associated calculations. Section 3.2 describes the process of elimination of the unknown absolute angle from the regression equation. In section 3.3 these results are used to develop the optimization algorithm that estimates the unknown robot parameters by maximization of the likelihood of the measured data. In the section 3.4 a simulated example dealing with the identification of a 3-link robot model is presented. Finally, in the section 3.5 an experiment focused on estimation of parameters of the leg of the prototype walking robot is given.

Using the link masses, the link inertia, the link lengths and other physical parameter — described in table 2.1 — to parametrize the motion equations (2.16) of a mechanical chain is quite natural. For the purpose of parameter estimation it is, however, more advantageous to use a parametrization that allows a model linear in parameters,

$$\mathbf{B}\boldsymbol{u}(t) = \boldsymbol{\tau}(t) = \mathbf{Z}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t))\boldsymbol{\beta}, \tag{3.1}$$

where the matrix $\mathbf{Z}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t))$ can be formed by manipulating the left-hand side of equation (2.16). The structural identifiability of parameters $\boldsymbol{\beta}$ can be easily ensured by choosing a parametrization that makes the corresponding matrix $\mathbf{Z}$ of full rank. To be able to exploit the linearity in the parameters, all variables entering the design matrix $\mathbf{Z}$, that is both relative position and absolute position, must be available. Then the problem of parameter estimation is reduced to the problem of multivariate multiple linear regression [Armstrong (1987)]. This case can be achieved using special platform where the absolute orientation of robot can be measured with respect to the platform itself. However, a walking robot freely moving in environment would have to obtain its absolute orientation using inertial measurements. These measurements, however, are likely to contain measurement noise. In this chapter the pursued idea is estimating the parameters in the presence of imperfect knowledge of absolute orientation from potentially noisy measurements.

3.1 Estimating relative angular velocities and accelerations

A typical situation encountered in practice is that the measurements of actuator torques and relative angles are available from motor-current sensors and rotary encoders, respectively. The encoder data are usually very accurate, and after digital filtering, practically noise free. For dataset consisting of N samples one obtains

$$\boldsymbol{y}_\tau(t_k) = \boldsymbol{\tau}(t_k) + \boldsymbol{e}_\tau(t_k) = \mathbf{B}\boldsymbol{y}_u(t_k), \tag{3.2}$$

$$\boldsymbol{y}_u(t_k) = \boldsymbol{u}(t_k) + \boldsymbol{e}_u(t_k), \tag{3.3}$$

$$\boldsymbol{y}_q^r(t_k) = \boldsymbol{q}^r(t_k), \tag{3.4}$$

where $k = 1 \dots N$. Actuator-torques measurement errors $\boldsymbol{e}_u(t_k)$ are assumed to be independent, identically distributed, following normal distribution and to have zero mean. That is

$$\mathbb{E}[\boldsymbol{e}_u(t_k)\boldsymbol{e}_u(t_k)^T] = \boldsymbol{\Sigma}_u, \tag{3.5}$$

$$\mathbb{E}[\boldsymbol{e}_u(t_k)\boldsymbol{e}_u(t_l)^T] = 0 \quad \text{for} \quad t_l \neq t_k. \tag{3.6}$$

As a consequence, $\boldsymbol{e}_\tau \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\tau)$, where $\boldsymbol{\Sigma}_\tau = \mathbf{B}\boldsymbol{\Sigma}_u\mathbf{B}^T$. On the other hand, the angular velocities and accelerations are usually not available, but in the case of relative angles, they can be estimated from angular data using central differences

$$\hat{\dot{\boldsymbol{q}}}^r_{fcd}(t_k) = \frac{\hat{\boldsymbol{q}}^r_f(t_k + T_s) - \hat{\boldsymbol{q}}^r_f(t_k - T_s)}{2T_s}, \tag{3.7}$$

$$\hat{\ddot{\boldsymbol{q}}}^r_{fcd}(t_k) = \frac{\hat{\boldsymbol{q}}^r_f(t_k + 2T_s) - 2\hat{\boldsymbol{q}}^r_f(t_k) + \hat{\boldsymbol{q}}^r_f(t_k - 2T_s)}{4T_s^2}, \tag{3.8}$$

where $\hat{\boldsymbol{q}}^r_f$ denotes digitally-filtered relative angular data, T_s the sampling rate and the filtering is used to reduce the noise amplification introduced by numerical differences. By using the estimates (3.7), (3.8) the dependence of (3.1) on relative velocities is effectively removed resulting in

$$\mathbf{Z}_{fcd}(\boldsymbol{q}^r_f, q_a, \dot{q}_a, \ddot{q}_a) = \mathbf{Z}(q_a, \boldsymbol{q}, \dot{q}_a, \dot{\boldsymbol{q}}^r, \ddot{q}_a, \ddot{\boldsymbol{q}}^r)\Big|_{\substack{\dot{\boldsymbol{q}}^r = \hat{\dot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r) \\ \ddot{\boldsymbol{q}}^r = \hat{\ddot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)}}, \tag{3.9}$$

where the time index was dropped to improve the readability. It can be seen that the only unknown variables are the absolute angle and its associated first and second derivatives, since the relative angles are assumed to be measured.

3.2 Eliminating the dependence on the absolute angle

There are various sensors that can measure various quantities that are directly related to the absolute angle. In this work only the distance from the ground and linear accelerations measured by an accelerometer combined with gyroscope measurements will be studied. These quantities can be used to eliminate the dependence on the absolute angle q^a and its derivatives $\dot{q}_a$, $\ddot{q}_a$ from equation (3.1). Achieving this would eliminate the

remaining unknown variables from the design matrix $\mathbf{Z}$ and would allow the model identification.

3.2.1 Using distance measurement model

Absolute orientation of the walking robot can be inferred from the distance measurement between some known point on robot and the some point on the ground. In this work, only flat surface is studied. One way of mounting the distance measuring sensor is depicted in the left part of fig. 3.1. The distance meter is rigidly fixed to the stance leg and is pointing towards the ground. Note that d_2 denotes the distance from the point where the stance leg touches the ground to the sensor location. Value of d_2 is known and constant. The distance d_1 is measured by the sensor. Further, the angle γ_3 between the stance leg and the optical axis of the sensor is also known and constant. In the following text the model of the 5-link robot with AO referenced to the stance leg will be used to demonstrate derivations of necessary equations. However, deriving the equations is possible also using other ways of AO angle definition. The underactuated angle q_1^a can be obtained using equations

$$q_1^a = \frac{\pi}{2} - \arcsin\left(\frac{d_1}{d_3}\sin(\gamma_3)\right), \qquad \text{for } d_1 \leqslant \frac{d_2}{\cos(\gamma_3)}, \tag{3.10}$$

$$q_1^a = -\frac{\pi}{2} + \arcsin\left(\frac{d_1}{d_3}\sin(\gamma_3)\right), \qquad \text{for } d_1 > \frac{d_2}{\cos(\gamma_3)}, \tag{3.11}$$

where

$$d_3 = \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos(\gamma_3)}. \tag{3.12}$$

Once the angle q_1^a is determined from the distance measurement d_1, the estimates for $\dot{q}_1^a$ and $\ddot{q}_1^a$ can be easily calculated using central differences. Using the formulas for the filtered central derivatives and formulas (3.10), (3.11) eliminates the dependence on absolute angle and all derivatives

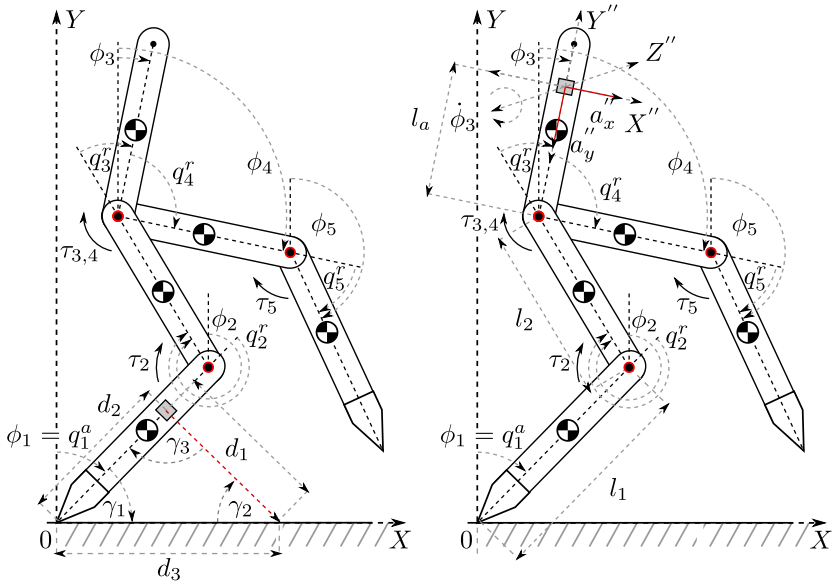


Figure 3.1 – Left: Laser distance-meter measurements. Right: IMU measurements.

from the design matrix $\mathbf{Z}$ resulting in

$$\mathbf{Z}_{d,fcd}(d_1, \boldsymbol{q}^r) = \mathbf{Z}(q^a, \boldsymbol{q}^r, \dot{q}_a, \dot{\boldsymbol{q}}^r, \ddot{q}_a, \ddot{\boldsymbol{q}}^r)\Big|_{\substack{q^a = h^d(d_1) \\ \dot{q}^a = \hat{\dot{q}}^a_{fcd}(q^a) \\ \ddot{q}^a = \hat{\ddot{q}}^a_{fcd}(q^a) \\ \dot{\boldsymbol{q}}^r = \hat{\dot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r) \\ \ddot{\boldsymbol{q}}^r = \hat{\ddot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)}}, \tag{3.13}$$

It is assumed that the distance measurements d_1 measured by the laser distance meter are corrupted with noise as in the following model

$$\boldsymbol{y}_{d_1}(t_k) = d_1(t_k) + \boldsymbol{e}_{d_1}(t_k), \tag{3.14}$$

where $e_{d_1}(t_k) \sim \mathcal{N}(0, \Sigma_{d_1})$. Measurement errors $\boldsymbol{e}_{\tau,d_1} = [\boldsymbol{e}_\tau^T, e_{d_1}^T]^T$ are assumed to be normally distributed with zero mean and to be uncorrelated

across time. Thus,

$$\mathbb{E}[\boldsymbol{e}_{\tau,d_1}(t_k)\boldsymbol{e}_{\tau,d_1}(t_k)^T] = \boldsymbol{\Sigma}_{\tau,d_1} = \begin{bmatrix} \boldsymbol{\Sigma}_\tau & 0 \\ 0 & \Sigma_{d_1} \end{bmatrix}, \tag{3.15}$$

$$\mathbb{E}[\boldsymbol{e}_{\tau,d_1}(t_k)\boldsymbol{e}_{\tau,d_1}(t_l)^T] = 0 \quad \text{for} \quad t_l \neq t_k. \tag{3.16}$$

The covariance of errors $\boldsymbol{e}_{d_1}$ is either constant or dependent on the distance d_1, a typical example is when the error of the distance measurement grows with the distance. The model for a whole dataset containing only measurable variables can be written as

$$\boldsymbol{y}_{\tau,1\ldots N} = \mathbf{Z}_{d,fcd,1\ldots N}(y_{d_1}, \boldsymbol{y}_{q^r})\boldsymbol{\beta} + \boldsymbol{e}_{\tau,1\ldots N}, \tag{3.17}$$

3.2.2 Using accelerometer and gyroscope model

The model of a simple IMU (Inertial Measurement Unit) composed of two axis accelerometer and one axis gyroscope is satisfactory for the purposes of estimation in the case of planar robot. The following model will be derived for the case when the IMU is located on the torso of the robot, but it can be easily adapted for different location of the sensor. The axes of both sensors — accelerometer and gyroscope — are identical and are denoted as X'', Y'', Z'' respectively. Sensors axes X'' and Y'' lie in the plane created by the inertial frame axes X and Y — where the motion of the robot takes place. Axis Y'' is aligned with robot's torso and the axis X'' is perpendicular to the robot's torso. Axis Z'' is perpendicular to the plane $X''Y''$ and XY.

Axes X'' and Y'' are aligned with body coordinate system $X'Y'$ of the robot's torso, body coordinate system is not shown in the fig. 3.1 for the sake of readability. The origin of the body coordinate system $X'Y'$ lies in the center of mass of the robot's torso. The origin of the sensor coordinate system lies in the center of the sensor, see the right part of the figure 3.1. Both body coordinate system $X'Y'$ and sensor coordinate system $X''Y''$ are rotated in a clock-wise direction by an angle $\phi_3 = q_1 + q_2 + q_3$ with respect to the base inertial coordinate system XY. The output of the

accelerometer will be the proper acceleration vector $\boldsymbol{a}_{ps}$ of the sensor decomposed into components acting along the accelerometer axes X'' and Y''. Note that the absolute angles $\boldsymbol{\phi}$, depicted on the figure 3.1, are given as

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_1 + q_2 \\ q_1 + q_2 + q_3 \\ q_1 + q_2 + q_3 + q_4 \\ q_1 + q_2 + q_3 + q_4 + q_5 \end{bmatrix} \tag{3.18}$$

The acceleration of the accelerometer can be calculated from its XY coordinates

$$x_a = l_1 \sin(\phi_1) + l_2 \sin(\phi_2) + l_a \sin(\phi_3), \tag{3.19}$$
$$y_a = l_1 \cos(\phi_1) + l_2 \cos(\phi_2) + l_a \cos(\phi_3). \tag{3.20}$$

By differentiating the coordinates of accelerometer the velocities are obtained as

$$\dot{x}_a = l_1 \cos(\phi_1)\dot{\phi}_1 + l_2 \cos(\phi_2)(\dot{\phi}_2) + l_a \cos(\phi_3)(\dot{\phi}_3), \tag{3.21}$$
$$\dot{y}_a = -l_1 \sin(\phi_1)\dot{\phi}_1 - l_2 \sin(\phi_2)(\dot{\phi}_2) - l_a \sin(\phi_3)(\dot{\phi}_3), \tag{3.22}$$

and the second derivatives of positions give the accelerations as

$$\begin{aligned} \ddot{x}_a = & -l_1 \sin(\phi_1)\dot{\phi}_1^2 + l_1 \cos(\phi_1)\ddot{\phi}_1 - l_2 \sin(\phi_2)(\dot{\phi}_2)^2 + \\ & + l_2 \cos(\phi_2)(\ddot{\phi}_2) - l_3 \sin(\phi_3)(\dot{\phi}_3)^2 + l_3 \cos(\phi_3)(\ddot{\phi}_3), \end{aligned} \tag{3.23}$$
$$\begin{aligned} \ddot{y}_a = & -l_1 \cos(\phi_1)\dot{\phi}_1^2 - l_1 \sin(\phi_1)\ddot{\phi}_1 - l_2 \cos(\phi_2)(\dot{\phi}_2)^2 + \\ & - l_2 \sin(\phi_2)(\ddot{\phi}_2) - l_3 \cos(\phi_3)(\dot{\phi}_3)^2 - l_3 \sin(\phi_3)(\ddot{\phi}_3). \end{aligned} \tag{3.24}$$

Define a clock-wise rotation matrix $\mathbf{R}^+(\phi)$ and the inverse-clock wise ro-

tation matrix $\mathbf{R}^{-}(\phi) = \text{inv}(\mathbf{R}^{+}(\phi))$ given as

$$\mathbf{R}^{+}(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}, \qquad \mathbf{R}^{-}(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{3.25}$$

The acceleration of the accelerometer (mounted on the torso) in the XY coordinates, given that $\boldsymbol{q}$ and $\boldsymbol{\phi}$ are defined as on fig. 3.1, can be written as

$$\begin{bmatrix} \ddot{x}_a \\ \ddot{y}_a \end{bmatrix} = \mathbf{R}^{+}(\phi_1) l_1 \begin{bmatrix} \ddot{\phi}_1 \\ -\dot{\phi}_1^2 \end{bmatrix} + \mathbf{R}^{+}(\phi_2) l_2 \begin{bmatrix} \ddot{\phi}_2 \\ -\dot{\phi}_2^2 \end{bmatrix} + \mathbf{R}^{+}(\phi_3) l_a \begin{bmatrix} \ddot{\phi}_3 \\ -\dot{\phi}_3^2 \end{bmatrix} \tag{3.26}$$

The proper acceleration vector $\boldsymbol{a}_{ps}$ of the accelerometer — the acceleration vector relative to the free fall — is

$$\boldsymbol{a}_{ps} = \boldsymbol{a}_s - \boldsymbol{g} = \begin{bmatrix} \ddot{x}_a \\ \ddot{y}_a \end{bmatrix} - \begin{bmatrix} 0 \\ g \end{bmatrix}. \tag{3.27}$$

The output of the accelerometer will be the proper acceleration $\boldsymbol{a}_{ps}^{''}$ of accelerometer with respect to the inertial frame $X^{''}Y^{''}$. Since the inertial frame $X^{''}Y^{''}$ is rotated by the angle ϕ_3 in a clockwise direction with respect to the coordinate frame XY, vector $\boldsymbol{a}_{ps}^{''}$ can be calculated by rotating the vector $\boldsymbol{a}_{ps}$ by the angle ϕ_3 in the counter-clockwise direction in order to obtain the same vector but in different coordinate system. Therefore

$$\boldsymbol{a}_{ps}^{''} = \mathbf{R}^{-}(\phi_3)\boldsymbol{a}_{ps} = \mathbf{R}^{-}(\phi_3)(\boldsymbol{a}_s - \boldsymbol{g}) = \mathbf{R}^{-}(\phi_3)\left(\begin{bmatrix} \ddot{x}_a \\ \ddot{y}_a \end{bmatrix} - \begin{bmatrix} 0 \\ g \end{bmatrix}\right) \tag{3.28}$$

and by substituting (3.26) into (3.28) the proper acceleration is obtained as

$$\boldsymbol{a}_{ps}^{''} = \mathbf{R}^{-}(\phi_3)\left(\mathbf{R}^{+}(\phi_1) l_1 \begin{bmatrix} \ddot{\phi}_1 \\ -\dot{\phi}_1^2 \end{bmatrix} + \mathbf{R}^{+}(\phi_2) l_2 \begin{bmatrix} \ddot{\phi}_2 \\ -\dot{\phi}_2^2 \end{bmatrix} +$$

$$+ \mathbf{R}^{+}(\phi_3) l_a \begin{bmatrix} \ddot{\phi}_3 \\ -\dot{\phi}_3^2 \end{bmatrix} - \begin{bmatrix} 0 \\ g \end{bmatrix} \Bigg). \tag{3.29}$$

Noticing that

$$\mathbf{R}^{+}(\phi_1) = \mathbf{R}^{+}(q_1), \tag{3.30}$$
$$\mathbf{R}^{+}(\phi_2) = \mathbf{R}^{+}(q_1 + q_2) = \mathbf{R}^{+}(q_1)\mathbf{R}^{+}(q_2), \tag{3.31}$$
$$\mathbf{R}^{+}(\phi_3) = \mathbf{R}^{+}(q_1 + q_2 + q_3) = \mathbf{R}^{+}(q_1)\mathbf{R}^{+}(q_2)\mathbf{R}^{+}(q_3), \tag{3.32}$$

and using the fact that $\mathbf{R}^{+}(q_1)\mathbf{R}^{+}(q_2)\mathbf{R}^{+}(q_3) = \mathbf{R}^{+}(q_3)\mathbf{R}^{+}(q_2)\mathbf{R}^{+}(q_1)$, relation (3.29) can be written as

$$\begin{aligned} \boldsymbol{a}_{ps}^{''} = {} & \mathbf{R}^{-}(q_3)\mathbf{R}^{-}(q_2) l_1 \begin{bmatrix} \ddot{q}_1 \\ -\dot{q}_1^2 \end{bmatrix} + \mathbf{R}^{-}(q_3) l_2 \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ -(\dot{q}_1 + \dot{q}_2)^2 \end{bmatrix} + \\ & + l_a \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3 \\ -(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \end{bmatrix} - \mathbf{R}^{-}(q_3)\mathbf{R}^{-}(q_2)\mathbf{R}^{-}(q_1) \begin{bmatrix} 0 \\ g \end{bmatrix}. \end{aligned} \tag{3.33}$$

From (3.33) it is clear that by a simple manipulation one can obtain the following relation

$$\begin{bmatrix} -\sin(q_1) \\ \cos(q_1) \end{bmatrix} g = \begin{bmatrix} \cos(q_1) & -\sin(q_1) \\ \sin(q_1) & \cos(q_1) \end{bmatrix} \begin{bmatrix} 0 \\ g \end{bmatrix} = \mathbf{R}^{-}(q_1) \begin{bmatrix} 0 \\ g \end{bmatrix} = \bar{\boldsymbol{h}}_a. \tag{3.34}$$

where function $\bar{\boldsymbol{h}}_a$ is defined as

$$\begin{aligned} \bar{\boldsymbol{h}}_a = \begin{bmatrix} \bar{h}_1^a \\ \bar{h}_2^a \end{bmatrix} := {} & -\mathbf{R}^{+}(q_3)\mathbf{R}^{+}(q_2) \Bigg(\boldsymbol{a}_{ps}^{''} - \mathbf{R}^{-}(q_3)\mathbf{R}^{-}(q_2) l_1 \begin{bmatrix} \ddot{q}_1 \\ -\dot{q}_1^2 \end{bmatrix} + \\ & + \mathbf{R}^{-}(q_3) l_2 \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ -(\dot{q}_1 + \dot{q}_2)^2 \end{bmatrix} + l_a \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3 \\ -(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \end{bmatrix} \Bigg). \end{aligned} \tag{3.35}$$

Finally,

$$\begin{bmatrix} \sin(q_1) \\ \cos(q_1) \end{bmatrix} = \boldsymbol{h}_a = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} := \frac{1}{g} \begin{bmatrix} -\bar{h}_1^a \\ \bar{h}_2^a \end{bmatrix}. \tag{3.36}$$

The resulting equation (3.36) can be used to calculate the sine and cosine of the absolute orientation angle q_1. Using $\sin(q_1)$ and $\cos(q_1)$ eliminates the variable q_1 from eq. (3.1) and (3.9), since the variable q_1 always appears in the motion equations as an argument of either sine or cosine. There still remains, however, the undetermined values of $\dot{q}_1$ and $\ddot{q}_1$, since the relative velocities and accelerations can be eliminated using (3.7) and (3.8). Variables $\dot{q}_1$ and $\ddot{q}_1$ are required for both calculating (3.36) and also in (3.1) or (3.9). However, the angular velocity $\dot{q}_1$ of the absolute orientation angle can be measured using gyroscope. And the angular acceleration $\ddot{q}_1$ can be calculated — in the case of offline estimation — using finite differences from the gyroscope measurements. After using all these information, the design matrix $\mathbf{Z}$ takes the following form

$$\mathbf{Z}_{a,fcd}(\overset{''}{\boldsymbol{a}}_{ps}, \boldsymbol{q}^r, \dot{q}_a) = \mathbf{Z}(q^a, \boldsymbol{q}^r, \dot{q}_a, \dot{\boldsymbol{q}}^r, \ddot{q}_a, \ddot{\boldsymbol{q}}^r)\Big|_{\substack{\sin(q_1)=h_1^a(\overset{''}{\boldsymbol{a}}_{ps},\boldsymbol{q}^r,\dot{q}^a),\\ \cos(q_1)=h_2^a(\overset{''}{\boldsymbol{a}}_{ps},\boldsymbol{q}^r,\dot{q}^a)\\ \dot{\boldsymbol{q}}^r=\hat{\dot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)\\ \ddot{\boldsymbol{q}}^r=\hat{\ddot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)\\ \ddot{q}^a=\hat{\ddot{q}}^a_{fcd}(\dot{q}^a)}} \tag{3.37}$$

where

$$\boldsymbol{h}_a(\overset{''}{\boldsymbol{a}}_{ps}, \boldsymbol{q}^r, \dot{q}^a) = \boldsymbol{h}(\overset{''}{\boldsymbol{a}}_{ps}, \boldsymbol{q}^r, \dot{q}^a, \dot{\boldsymbol{q}}^r, \ddot{q}^a, \ddot{\boldsymbol{q}}^r,)\Big|_{\substack{\dot{\boldsymbol{q}}^r=\hat{\dot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)\\ \ddot{\boldsymbol{q}}^r=\hat{\ddot{\boldsymbol{q}}}^r_{fcd}(\boldsymbol{q}^r)\\ \ddot{q}^a=\hat{\ddot{q}}^a_{fcd}(\dot{q}^a)}} \tag{3.38}$$

It is assumed that both, the proper acceleration $\overset{''}{\boldsymbol{a}}_{ps}$ measured by the accelerometer and the angular velocity $\dot{q}^a$ measurements measured by the gyroscope, are corrupted with noise as in the following model

$$\boldsymbol{y}_a(t_k) = \overset{''}{\boldsymbol{a}}_{ps}(t_k) + \boldsymbol{e}_a(t_k), \tag{3.39}$$

$$y_{\dot{q}^a}(t_k) = \dot{q}^a(t_k) + e_{\dot{q}^a}(t_k). \tag{3.40}$$

For a compact reference the following vectors are defined

$$\boldsymbol{y}_{imu}(t_s) = [\boldsymbol{y}_a(t_k), y_{\dot{q}^a}(t_k)]^T, \tag{3.41}$$

$$\boldsymbol{e}_{imu}(t_s) = [\boldsymbol{e}_a(t_k), e_{\dot{q}^a}(t_k)]^T, \tag{3.42}$$

where $\boldsymbol{e}_{imu}(t_k) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_a)$. Measurement errors $\boldsymbol{e}_{\tau,imu} = [\boldsymbol{e}_\tau^T, \boldsymbol{e}_{imu}^T]^T$ are assumed to be normally distributed with zero mean and to be uncorrelated across time. Thus,

$$\mathbb{E}[\boldsymbol{e}_{\tau,imu}(t_k)\boldsymbol{e}_{\tau,imu}(t_k)^T] = \boldsymbol{\Sigma}_{\tau,imu} = \begin{bmatrix} \boldsymbol{\Sigma}_\tau & 0 \\ 0 & \boldsymbol{\Sigma}_{imu} \end{bmatrix}, \tag{3.43}$$

$$\mathbb{E}[\boldsymbol{e}_{\tau,imu}(t_k)\boldsymbol{e}_{\tau,imu}(t_l)^T] = 0 \quad \text{for} \quad t_l \neq t_k. \tag{3.44}$$

The model for a whole dataset containing only measured variables can be written as

$$\boldsymbol{y}_{\tau,1\ldots N} = \mathbf{Z}_{a,fcd,1\ldots N}(\boldsymbol{y}_{a''_{ps}}, \boldsymbol{y}_{q^r}, y_{\dot{q}_a})\boldsymbol{\beta} + \boldsymbol{e}_{\tau,1\ldots N}. \tag{3.45}$$

3.2.3 Linear least-squares regression

For the sake of generality following variables are introduced,

$$\boldsymbol{y}_\eta(t_k) = \boldsymbol{\eta}(t_k) + \boldsymbol{e}_\eta(t_k) \tag{3.46}$$

$$\boldsymbol{y}_\alpha(t_k) = \boldsymbol{\alpha}(t_k). \tag{3.47}$$

where

$$\boldsymbol{y}_\eta(t_k) = [y_{\eta_1}(t_k), \ldots, y_{\eta_{n_\eta}}(t_k)]^T, \tag{3.48}$$

$$\boldsymbol{y}_\alpha(t_k) = [y_{\alpha_1}(t_k), \ldots, y_{\eta_{n_\alpha}}(t_k)]^T, \tag{3.49}$$

$$\boldsymbol{\eta}(t_k) = [\eta_1(t_k), \ldots, \eta_{n_\eta}(t_k)]^T, \tag{3.50}$$

$$\boldsymbol{\alpha}(t_k) = [\alpha_1(t_k), \ldots, \alpha_{n_\alpha}(t_k)]^T, \tag{3.51}$$

$$\boldsymbol{e}_\eta(t_k) = [e_{\eta_1}(t_k), \ldots, e_{\eta_{n_\eta}}(t_k)]^T. \tag{3.52}$$

Variable $\boldsymbol{y}_\eta(t_k)$ stands for measurements of variables $\boldsymbol{\eta}(t_k)$ containing noise $\boldsymbol{e}_\eta(t_k)$. Variable $\boldsymbol{y}_\alpha(t_k)$ stands for measurements of variables $\boldsymbol{\alpha}(t_k)$ which are noise-free. The measurement error $\boldsymbol{e}_\eta(t_k) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\eta)$. Measurement errors $\boldsymbol{e}_{\tau,\eta} = [\boldsymbol{e}_\tau^T, \boldsymbol{e}_\eta^T]^T$ are assumed to be normally distributed

with zero mean and to be uncorrelated across time. Thus,

$$\mathbb{E}[\boldsymbol{e}_{\tau,\eta}(t_k)\boldsymbol{e}_{\tau,\eta}(t_k)^T] = \boldsymbol{\Sigma}_{\tau,\eta} = \begin{bmatrix} \boldsymbol{\Sigma}_\tau & 0 \\ 0 & \boldsymbol{\Sigma}_\eta \end{bmatrix}, \tag{3.53}$$

$$\mathbb{E}[\boldsymbol{e}_{\tau,\eta}(t_k)\boldsymbol{e}_{\tau,\eta}(t_l)^T] = 0 \quad \text{for} \quad t_l \neq t_k. \tag{3.54}$$

The general model that can accommodate different sensors (distance measurement model or IMU model) can now be written as

$$\boldsymbol{y}_{\tau,1\ldots N} = \mathbf{Z}_{\eta 1\ldots N}(\boldsymbol{\eta},\boldsymbol{\alpha})\boldsymbol{\beta} + \boldsymbol{e}_{\tau,1\ldots N}, \tag{3.55}$$

where the design matrix $\mathbf{Z}_\eta(\boldsymbol{\eta},\boldsymbol{\alpha})$ is given either by the design matrix based on the distance measurements or by the design matrix based on the IMU measurements. In the case of distance measurements the only noisy variable is the distance d_1, therefore $\eta = d_1$. In the case of IMU the noisy variables are both the accelerations and the angular velocity, therefore $\boldsymbol{\eta} = [(\boldsymbol{a}''_{ps})^T, (\dot{\boldsymbol{q}}_1^a)^T]^T$. Since the structure of the identified system can be described accurately by the Euler-Lagrange equations, it is assumed that the errors due to unmodeled dynamics can be neglected. If there are any additional dynamic phenomena like friction or gear dynamics they should be included in the model. In the case when the absolute orientation positional data are accurate, the discrepancies in the design matrix as well as potential correlation due to closed-loop, can both be neglected and the problem of parameter estimation reduces to the linear regression problem, similarly as in the case of manipulator robots, see [Poignet and Gautier (2000)], [Olsen and Petersen (2001)], [Olsen et al. (2002)] and references therein. In such a case the design matrix $Z_\eta(\boldsymbol{\alpha})$ is composed only from accurate measurements and $\boldsymbol{\beta}$ can be estimated using weighted least-squares estimator

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}_{\eta,1\ldots N}^T \mathbf{W} \mathbf{Z}_{\eta,1\ldots N})^{-1} (\mathbf{Z}_{\eta,1\ldots N}{}^T \mathbf{W} \boldsymbol{y}_{1\ldots N}) \ , \tag{3.56}$$

The weighting matrix $\mathbf{W}$ is equal to

$$\mathbf{W} = \boldsymbol{\Sigma}_\tau \otimes \mathbb{1}_{N\times N}, \tag{3.57}$$

where the symbol $\otimes$ denotes the Kronecker product. Thanks to elimination of variables that are usually not measured, the method of linear least-squares can be used to estimate the parameters of the walking robot. However, this method does not explicitly account for the measurement noise in the absolute orientation data and therefore does not provide optimal estimates when measurements are noisy. Unfortunately, both the distance meter and IMU tend to provide measurements corrupted with noise and simple digital filtering does not provide optimal results.

3.3 Maximum likelihood estimation of parameters from open-loop noisy data

Due to errors in the noisy measurements $\boldsymbol{y}_\eta$ of the robot's absolute orientation the design matrix

$$\mathbf{Z}_\eta(\boldsymbol{y}_\eta, \boldsymbol{y}_\alpha) \neq \mathbf{Z}_\eta(\boldsymbol{\eta}, \boldsymbol{\alpha}) \tag{3.58}$$

and the resulting estimate of parameters will be biased. Therefore one would like to find estimates of $\hat{\boldsymbol{\eta}}_{mle}(t_k)$, which most likely correspond to the values of $\boldsymbol{\eta}(t_k)$, for $k = 1, \ldots, N$. It is assumed that the error properties (3.43) are known. To be more specific, matrix $\boldsymbol{\Sigma}_{\tau,\eta}$, or matrices $\boldsymbol{\Sigma}_{\boldsymbol{\tau}}$ and $\boldsymbol{\Sigma}_\eta$, are known up to a scalar multiple. Estimates of these matrices can be obtained from specially designed repeated experiments. The distribution of measurements $\boldsymbol{y}_\tau(t_k)$ and $\boldsymbol{y}_\eta(t_k)$ is given by multivariate normal distribution

$$f(\boldsymbol{y}_\tau(t_k), \boldsymbol{y}_\eta(t_k)) = \\ \frac{1}{\sqrt{(2\pi)^{2r}|\boldsymbol{\Sigma}_{\tau,\eta}|}} \exp\left(-\frac{1}{2}\boldsymbol{e}(t_k)^T \boldsymbol{\Sigma}_{\tau,\eta}^{-1} \boldsymbol{e}(t_k)\right). \tag{3.59}$$

The likelihood function of measurements $\boldsymbol{y}_\tau(t_k)$ and $\boldsymbol{y}_\eta(t_k)$ is a function of $\boldsymbol{\tau}(t_k)$ and $\boldsymbol{\eta}(t_k)$ and is proportional to (3.59). By using the model (3.55),

estimator (3.56) one concludes that ℓ for a whole dataset is given as

$$\ell(\boldsymbol{\eta}_{1\dots N}) = \prod_{k=1}^{k=N} \exp\left(-\frac{1}{2}\boldsymbol{e}(t_k)^T \boldsymbol{\Sigma}_{\tau,\eta}^{-1} \boldsymbol{e}(t_k)\right). \tag{3.60}$$

Maximization of $\ell(\boldsymbol{\eta}_{1\dots N})$ can be obviously achieved by minimizing the sum of squares

$$S(\boldsymbol{\eta}_{1\dots N}) = \sum_{k=1}^{k=N} \left(\boldsymbol{e}(t_k)^T \boldsymbol{\Sigma}_{\tau,\eta}^{-1} \boldsymbol{e}(t_k)\right). \tag{3.61}$$

3.3.1 Implementation of the minimization procedure

Minimizing the criterion (3.61) is a nonconvex optimization problem and therefore the solution has to be sought iteratively. However, the sum of squares can be calculated extremely efficiently thanks to utilizing the WLS estimator in conjunction with the central differences. In particular, this approach does not require to integrate the equations of motion. The implementation of the identification procedure can be performed using the following steps, with some small differences which depend on whether the absolute orientation was measured using the distance meter or using the combination of accelerometers and gyroscopes.

1. *Initialization*:
(a) Set the iteration counter to $i = 1$.
(b) For $k = 1 \dots N$, filter the actuator torques, relative angles, the quantity $\boldsymbol{\eta}$ related to the absolute angle:

$$\hat{\boldsymbol{u}}_f(t_k) = \text{lowpass}(\boldsymbol{y}_u(t_k)), \tag{3.62}$$
$$\hat{\boldsymbol{q}}_f^r(t_k) = \text{lowpass}(\boldsymbol{y}_q^r(t_k)), \tag{3.63}$$
$$\hat{\boldsymbol{\eta}}_f(t_k) = \text{lowpass}(\boldsymbol{y}_\eta(t_k)). \tag{3.64}$$

(c) For $k = 1 \dots N$ calculate $\boldsymbol{y}_\tau(t_k) = \mathbf{B}\hat{\boldsymbol{u}}_f(t_k)$ and form the vector $\boldsymbol{y}_{\tau,1\dots N}$.

(d) Initialize the Maximum Likelihood (ML) estimator of $\boldsymbol{\eta}$ as

$$\hat{\boldsymbol{\eta}}_{mle,1}(t_k) = \hat{\boldsymbol{\eta}}_f(t_k), \tag{3.65}$$

2. *Calculation of minimization criterion*:
(a) In the case of distancemeter, estimate the angle $\hat{q}^a_{mle}(t_k)$ using the estimate $\hat{\boldsymbol{\eta}}_{mle,i}(t_k)$ and for $k = 1 \dots N$ calculate

$$\hat{\dot{q}}^a_{mle,i}(t_k) = \frac{\hat{q}^a_{mle,i}(t_k + T_s) - \hat{q}^a_{mle,i}(t_k - T_s)}{2T_s}. \tag{3.66}$$

In the case of accelerometers and gyroscopes, this step is omitted.
(b) For $k = 1 \dots N$, calculate the central differences based estimate from the current ML estimate $\hat{\dot{q}}^a_{mle,i}(t_k)$ as

$$\hat{\ddot{q}}^a_{mle,i}(t_k) = \frac{\hat{\dot{q}}^a_{mle,i}(t_k + T_s) - \hat{\dot{q}}^a_{mle,i}(t_k - T_s)}{2T_s}. \tag{3.67}$$

(c) Form the design matrix $\mathbf{Z}_\eta$ for each $k = 1 \dots N$ as,

$$\mathbf{Z}_\eta(t_k) = \mathbf{Z}_\eta(\hat{\boldsymbol{\eta}}_{mle,i}(t_k), \hat{\boldsymbol{q}}^r(t_k)). \tag{3.68}$$

(d) Form matrices $\mathbf{Z}_{1\dots N}$ and $\mathbf{W}$ and calculate the estimate of parameters as

$$\hat{\boldsymbol{\beta}} = ({\mathbf{Z}_{\eta,1\dots N}}^T \mathbf{W} \mathbf{Z}_{\eta,1\dots N})^{-1} (\mathbf{Z}^T_{\eta,1\dots N} \mathbf{W} \boldsymbol{y}_{\tau,1\dots N}). \tag{3.69}$$

(e) For $k = 1 \dots N$, calculate the joint torques predicted by the ML estimates

$$\boldsymbol{\tau}_{mle,i}(t_k) = \mathbf{Z}_\eta(\hat{\boldsymbol{\eta}}_{mle,i}(t_k), \hat{\boldsymbol{q}}^r(t_k))\boldsymbol{\beta}. \tag{3.70}$$

(f) For $k = 1 \dots N$, calculate residuals

$$\boldsymbol{e}_\tau(t_k) = \boldsymbol{y}_\tau(t_k) - \boldsymbol{\tau}_{mle,i}(t_k), \tag{3.71}$$
$$\boldsymbol{e}_\eta(t_k) = \boldsymbol{y}_\eta(t_k) - \hat{\boldsymbol{\eta}}_{mle,i}(t_k). \tag{3.72}$$

Form the error to be optimized as $\boldsymbol{e}(t_k) = [\boldsymbol{e}^T_\tau(t_k), \boldsymbol{e}_\eta(t_k)]^T$. Form the

sum of squares to be minimized

$$S = \sum_{k=1}^{k=N} \left(\boldsymbol{e}(t_k)^T \boldsymbol{\Sigma}_{\tau,\eta}^{-1} \boldsymbol{e}(t_k) \right) . \tag{3.73}$$

3. *Perform Gauss-Newton update*:
(a) Form the numerical gradient $\boldsymbol{g}_{mle}$ of S.
(b) Form the numerical Hessian $\mathbf{H}$ of S.
(c) Obtain the new ML estimate as

$$\hat{\boldsymbol{\eta}}_{mle,i,1\ldots N} = \mathbf{H}^{-1} \boldsymbol{g}_{mle}. \tag{3.74}$$

(d) Increase iteration counter i by one and continue to step 2 or end if a stopping criterion of the optimization procedure is satisfied.

The steps (3.a) to (3.d) are only illustrative and can be replaced by a particular implementation of nonlinear least-squares minimization procedure.

3.4 Application to parameter estimation of a planar bipedal walking robot

This section studies the simulation example of one of the simplest mechanical systems capable of underactuated planar bipedal walking [Grizzle et al. (2001)]. The robot is composed of three rigid links - two legs and a torso - connected by two actuated rotary joints. Schematics of robot are depicted on fig. 3.2. The set of configuration angles $\boldsymbol{q}$ is composed of one absolute angle q_1^a and two relative angles q_2^r, q_3^r. It is assumed that the relative angles would be measured by incremental encoders and that the absolute orientation of the robot would be obtained by a laser distance sensor. The distance meter is rigidly fixed to the stance leg and is pointing towards the ground, see fig. 3.2. Note that d_2 denotes the distance from the point where the stance leg touches the ground to the sensor location. Value of d_2 is known and constant. The distance d_1 is measured by the sensor. Further, the angle γ_3 between the stance leg and the optical axis

Table 3.1 – Parameters of the 3-link robot

l_1, l_2, l_3	length of 1^{st}, 2^{nd} and 3^{rd} link	$[m]$
l_{c_1}, l_{c_2}, l_{c_3}	center of gravity of 1^{st}, 2^{nd} and 3^{rd} link	$[m]$
m_1, m_2, m_3	mass of 1^{st}, 2^{nd} and 3^{rd} link	$[kg]$
I_1, I_2, I_3	inertia of 1^{st}, 2^{nd} and 3^{rd} link	$[kg.m^2]$
τ_1, τ_2	torque generated by 1^{st} and 2^{nd} motor	$[N.m]$
g	gravitational acceleration	$[m.s^{-2}]$

of the sensor is also known and constant. The underactuated angle q_1^a can be obtained using equations

$$q_1^a = \frac{\pi}{2} - \arcsin\left(\frac{d_1}{d_3}\sin(\gamma_3)\right), \qquad \text{for } d_1 \leqslant \frac{d_2}{\cos(\gamma_3)}, \tag{3.75}$$

$$q_1^a = -\frac{\pi}{2} + \arcsin\left(\frac{d_1}{d_3}\sin(\gamma_3)\right), \qquad \text{for } d_1 > \frac{d_2}{\cos(\gamma_3)}, \tag{3.76}$$

where

$$d_3 = \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos(\gamma_3)}. \tag{3.77}$$

Once the angle q_1^a is determined from the distance measurements d_1, the estimates for $\dot{q}_1^a$ and $\ddot{q}_1^a$ can be easily calculated using central differences. In this simulated study the conversion from d_1 to q_1^a was omitted for simplicity and it is assumed that directly q_1^a is measurable, thus the following measurement model was used both in the simulations and in the estimation procedure

$$y_{q_1^a} = q_1^a + e_{q_1^a}, \tag{3.78}$$

$$e_{q_1^a} \sim N(0, \sigma^2_{q_1^a}), \tag{3.79}$$

where the variance $\sigma^2_{q_1^a}$ is known. Physical parameters of the robot are listed in tab. 3.1. Values of physical robot parameters are based on a numerical example of [Grizzle et al. (2001)] where a different parametrization

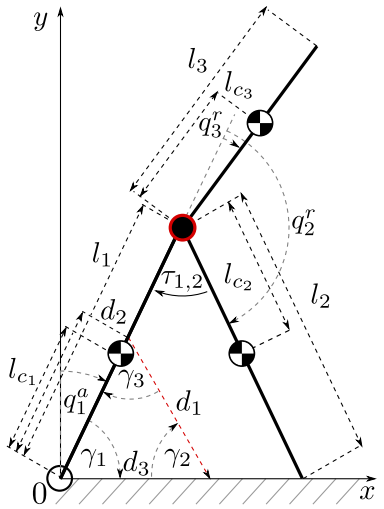


Figure 3.2 – Schematics of a 3-link robot.

is used. Parameters are as follows

$$\begin{aligned} l_1 &= 1, & l_2 &= 1, & l_3 &= 0.5, \\ l_{c_1} &= 0.5, & l_{c_2} &= 0.5, & l_{c_3} &= 0.2, \\ m_1 &= 5, & m_2 &= 5, & m_3 &= 25, \\ I_1 &= 0, & I_2 &= 0, & I_3 &= 1.5\,. \end{aligned} \tag{3.80}$$

Lagrangian formalism results in the following model matrices

$$\mathbf{D} = \begin{bmatrix} \beta_1+\beta_2+\beta_3+2\beta_4\mathrm{c}_2^r+2\beta_5\mathrm{c}_3^r, & \beta_2+\beta_4\mathrm{c}_2^r, & \beta_3+\beta_5\mathrm{c}_3^r \\ \beta_2+\beta_4\mathrm{c}_2^r, & \beta_2, & 0 \\ \beta_3+\beta_5\mathrm{c}_3^r, & 0, & \beta_3 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} -\beta_4\dot{q}_2^r\mathrm{s}_2^r-\beta_5\dot{q}_3^r\mathrm{s}_3^r, & -\beta_4\mathrm{s}_2^r(\dot{q}_1^a+\dot{q}_2^r), & -\beta_5\mathrm{s}_3^r(\dot{q}_1^a+\dot{q}_3^r) \\ \beta_4\dot{q}_1^a\mathrm{s}_2^r, & 0, & 0 \\ \beta_5\dot{q}_1^a\mathrm{s}_3^r, & 0, & 0 \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} -\beta_6\mathrm{s}_1^a-\beta_7\mathrm{s}_{1,2}^{a+r}-\beta_8\mathrm{s}_{1,3}^{a+r} \\ -\beta_7\mathrm{s}_{1,2}^{a+r} \\ -\beta_8\mathrm{s}_{1,3}^{a+r} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix},$$

using the following abbreviations: $\mathrm{c}_2^r = \cos(q_2^r)$, $\mathrm{c}_3^r = \cos(q_3^r)$, $\mathrm{s}_1^a = \sin(q_1^a)$, $\mathrm{s}_2^r = \sin(q_2^r)$, $\mathrm{s}_3^r = \sin(q_3^r)$, $\mathrm{s}_{1,2}^{a+r} = \sin(q_1^a + q_2^r)$, $\mathrm{s}_{1,3}^{a+r} = \sin(q_1^a + q_3^r)$. Substitution

$$\begin{aligned} \beta_1 &= I_1 + l_1^2 m_2 + l_1^2 m_3 + l_{c_1}^2 m_1, \quad & \beta_2 &= I_2 + m_2 l_{c_2}^2, \\ \beta_3 &= I_3 + m_3 l_{c_3}^2, & \beta_4 &= l_1 l_{c_2} m_2, \\ \beta_5 &= l_1 l_{c_3} m_3, & \beta_6 &= g(l_1 m_2 + l_1 m_3 + l_{c_1} m_1), \\ \beta_7 &= g l_{c_2} m_2, & \beta_8 &= g l_{c_3} m_3, \end{aligned} \tag{3.81}$$

makes the model linear with respect to the parameters $\boldsymbol{\beta}$. The state space vector is given as

$$\boldsymbol{x} = (q_1^r, q_2^r, q_3^a, \dot{q}_1^r, \dot{q}_2^r, \dot{q}_3^a)^T. \tag{3.82}$$

3.4.1 Monte Carlo analysis

Since the measurement of the absolute angle is corrupted with noise, a Monte Carlo simulations were utilized to analyze the performance of the estimation algorithms. Four cases with different variance $\sigma_{q_1^a}^2$ were analyzed, see tab. 3.2. Furthermore the case when $\sigma_{q_1^a}^2 = 10^{-3}$ was also studied. In this case the data begin to be too noisy with respect to their curvature and the resulting estimates ceases to be reliable due to premature termination of the optimization procedure. This case is used to demonstrate the effects of local minima. As a validation of proposed estimation method a 10^3 Monte Carlo simulations were performed for each setting of the variance $\sigma_{q_1^a}^2$ totaling in 4×10^3 simulations for the four cases described in the tab. 3.2 and additional 10^3 simulations were analyzed for the case when $\sigma_{q_1^a}^2$ equals 10^{-3}. The short duration of the experiments is due to the open-loop controller and the unstable nature of walking.

The MLE algorithm was compared with the WLS estimator based on the filtered central differences. Filtering of the absolute angle measurements $y_{q_1}^a$ was realized using Matlab built in function `filtfilt` which performs forward and backward filtering to reduce the distortion of data. The

designed filter was a Butterworth filter of 5^{th} order with a half power frequency, well tuned for each $\sigma^2_{q_1^a}$ separately according to the tab.3.2. The resulting filtered data are denoted as $\hat{q}^a_{1,f}$. Relative angular data $\boldsymbol{q}^r$ were not filtered, since they are available noise-free. The estimates of velocities and accelerations are calculated using central differences based on data $\hat{\boldsymbol{q}}_f = [\boldsymbol{q}^{rT}, \hat{q}^a_f]^T$as

$$\hat{\dot{\boldsymbol{q}}}_{fcd}(t_k) = \frac{\hat{\boldsymbol{q}}_f(t_k + n_{cd}T_s) - \hat{\boldsymbol{q}}_f(t_k - n_{cd}T_s)}{2n_{cd}T_s}, \tag{3.83}$$

$$\hat{\ddot{\boldsymbol{q}}}_{fcd}(t_k) = \frac{\hat{\dot{\boldsymbol{q}}}_{fcd}(t_k + n_{cd}T_s) - \hat{\dot{\boldsymbol{q}}}_{fcd}(t_k - n_{cd}T_s)}{2n_{cd}T_s}. \tag{3.84}$$

Parameter $n_{cd} \geqslant 1$ is related to the number of data samples used in the central-difference approximation of a derivative. Making the parameter $n_{cd} > 1$ results in a smoother estimates of the derivatives for the specific values refer to the tab.3.2. This algorithm will be denoted as WLS with Filtered Central Differences (WLS-FCD). The MLE algorithm initial estimates of angles, angular velocities and accelerations are calculated using WLS-FCD algorithm. For comparison purposes the Recursive Least Squares (RLS) and Hybrid Extended Kalman filter (HEKF) were tested as well. Their detailed implementation is provided in the chapter 2 describing preliminaries. The results of all estimation algorithms are summarized in the tab.3.3. By studying the tab.3.3, one can see that the performance of the MLE method is superior to all other approaches, especially for the cases where the noise has larger variance. The RLS method however works also very well. The problem of HEKF lies in high nonlinearity of the system and therefore the filter is often unstable and fails to integrate.

3.5 Parameter estimation of a Leg of Laboratory Walking Robot

The aim of this section is to verify the proposed estimation procedure, described in the section 3.3, by estimating parameters of the leg of a prototype underactuated walking robot which was firmly attached to a rigid

Table 3.2 – Four studied cases and most relevant parameters

Simulation parameters	Case 1	Case 2	Case 3	Case 4
$\sigma^2_{q_1}$	10^{-10}	10^{-8}	10^{-6}	10^{-4}
$\sigma^2_{u_1}$, $\sigma^2_{u_2}$	10^{-1}	10^{-1}	10^{-1}	10^{-1}
Sampling Period T_s $[s]$	0.01	0.01	0.01	0.01
Experiment Duration T $[s]$	0.68	0.68	0.68	0.68
WLS-FCD parameters				
Half-power freq.	1	0.5	0.25	0.1
C.D. samples n_{cd} [-]	2	3	5	7

Table 3.3 – Sample mean and variance of summed squared simulation errors of the 3-link robot

MLE	Case 1	Case 2	Case 3	Case 4
Mean	1.242	1.231	1.195	1.149
Variance	0.009	0.010	0.011	2.567
Failed simulations	0	0	0	2
WLS-FCD	Case 1	Case 2	Case 3	Case 4
Mean	1.753	1.884	2.480	$6 * 10^4$
Variance	0.016	0.055	5.023	$1.9 * 10^{12}$
Failed simulations	0	0	0	132
HEKF	Case 1	Case 2	Case 3	Case 4
Mean	6.804	214.1	10^3	196.1
Variance	455.9	10^7	$2 * 10^8$	$2.98 * 10^6$
Failed simulations	25	228	246	160
RLS	Case 1	Case 2	Case 3	Case 4
Mean	1.245	0.5952	0.5136	7.1788
Variance	0.009	0.0279	0.1169	173.1557
Failed simulations	0	0	0	4

platform. The identified system is depicted in fig. 3.3. The leg consists of two rigid links and two joints, the joint between the links is actuated, the other joint is left intentionally unactuated and so it plays the role of the pivot. Therefore, only one motor is installed and the motor is equipped with an optical encoder to measure the relative angle q_2 between the links. Further, both joint angles q_1 and q_2 can be measured using a contactless hall encoders, which are used in order to provide benchmark measurements of both the actuated and the underactuated angle. However, these measurements are not used by the parameter estimation procedure. Therefore this system possesses principal challenges that are characteristic for identification of underactuated walking robots. Finally, the motor torque τ is measured using current shunt monitor and the leg is equipped with a 3-axis accelerometer - measuring proper acceleration a_x parallel with the leg link and a_y perpendicular to the upper link, both in the sagittal plane and a_z in the lateral plane - and a gyroscope - measuring angular velocity $\dot{q}_1$. The joint angle q_1 is defined with respect to the horizontal axis passing through the center of the joint, parallel with the sagittal plane of the robot. Angle q_2 is defined as a relative angle between the links. Corresponding physical parameters of the robot are listed in the tab. 3.4. A substitution rendering the model linear in parameters was used resulting in matrices of the model given by (3.86).

Table 3.4 – Parameters of the 2-link robot leg

l_1, l_2	length of 1$^{\text{st}}$ and 2$^{\text{nd}}$ link	$[m]$
l_{c_1}, l_{c_2}	center of gravity of 1$^{\text{st}}$ and 2$^{\text{nd}}$ link	$[m]$
m_1, m_2	mass of 1$^{\text{st}}$ and 2$^{\text{nd}}$ link	$[kg]$
I_1, I_2	inertia of 1$^{\text{st}}$ and 2$^{\text{nd}}$ link	$[kg.m^2]$
μ_1, μ_2	viscous friction parameters	$[N.s/m]$
μ_3, μ_4	Coulomb friction parameters	$[N.s/m]$
g	gravitational acceleration	$[m.s^{-2}]$

Figure 3.3 – Photo of the prototype walking robot - front leg is identified.

$$
\begin{aligned}
\beta_1 &= m_1 l_{c_1}^2 + m_2 l_1^2 + I_1, & \beta_6 &= \mu_1, \\
\beta_2 &= m_2 l_{c_2}^2 + I_2, & \beta_7 &= \mu_2, \\
\beta_3 &= m_2 l_1 l_{c_2}, & \beta_8 &= \mu_3, \\
\beta_4 &= m_1 l_{c_1} + m_2 l_1, & \beta_9 &= \mu_4, \\
\beta_5 &= m_2 l_{c_2}. &&
\end{aligned}
\tag{3.85}
$$

$$
\boldsymbol{D} = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos(q_2); & \beta_2 + \beta_3 \cos(q_2) \\ \beta_2 + \beta_3 \cos(q_2); & \beta_2 \end{bmatrix}, \tag{3.86}
$$

$$\boldsymbol{C} = \begin{bmatrix} -2\beta_3 \sin(q_2)\dot{q_2}; & -\beta_3 \sin(q_2)\dot{q_2} \\ \beta_3 \sin(q_2)\dot{q_1}; & 0 \end{bmatrix}, \tag{3.87}$$

$$\boldsymbol{G} = \begin{bmatrix} \beta_4 g \sin(q_1) + \beta_5 g \sin(q_1 + q_2) \\ \beta_5 g \sin(q_1 + q_2) \end{bmatrix}, \tag{3.88}$$

$$\mathbf{F} = \begin{bmatrix} \beta_6 \dot{q}_1 + \beta_8 \text{sign}(\dot{q}_1) \\ \beta_7 \dot{q}_2 + \beta_9 \text{sign}(\dot{q}_2) \end{bmatrix}, \tag{3.89}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.90}$$

The absolute orientation angle can be related with measurements using the accelerometer data a_x, a_y, angular velocity $\dot{q}_1$ measured by gyroscope and the angular acceleration $\ddot{q}_1$ - which can be easily obtained by filtered differences from $\dot{q}_1$. This results in the following relation

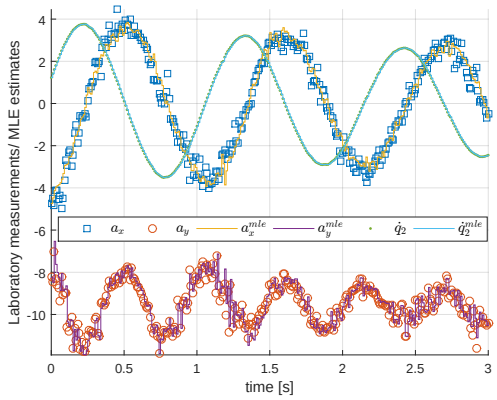
$$h = \begin{bmatrix} \sin(q_1) \\ \cos(q_1) \end{bmatrix} = \begin{bmatrix} \dfrac{l_s \ddot{q_1} - a_{x,s}}{g} \\ \dfrac{-a_{y,s} - l_s \dot{q}_1^2}{g} \end{bmatrix}. \tag{3.91}$$

Since $\sin(q_1 + q_2) = \sin(q_1)\cos(q_2) + \cos(q_1)\sin(q_2)$ and by combining with the optical incremental encoder measurements of the angle q_2 and by approximating its derivatives $\dot{q}_2$, $\ddot{q}_2$ using filtered central differences, all the data required for constructing matrices $\mathbf{D}$, $\mathbf{C}$, $\mathbf{G}$ are available. By using also the torque measurements it is possible to use the MLE method to estimate parameters $\boldsymbol{\beta}$. Table 3.5 shows summed squared errors between response of simulated model obtained by identification and the real validation data - not the dataset used for identification. Data were collected at sampling period of $0.01[s]$. Three methods were compared, WLS method based only on Hall sensor measurements for benchmark, then both WLS

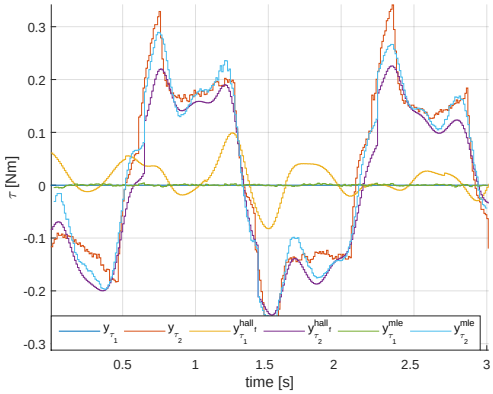
Table 3.5 – Summed squared errors for identified model of the robot leg simulated on validation data from real laboratory experiment

Method	$\sum e_{q_1}^2$	$\sum e_{q_2}^2$	$\sum e_{\dot{q}_1}^2$	$\sum e_{\dot{q}_2}^2$
WLS-Hall	12.927	17.064	18.284	7.762
WLS-IMU	10.216	19.258	17.341	7.620
MLE-IMU	0.919	1.748	25.927	6.028

and MLE methods based on gyroscope and accelerometer data. It can be seen that the MLE based model is superior in predicting the angles q_1 and q_2, which is exactly what is most important, even in comparison with the benchmark WLS method based on Hall measurements. The covariance matrices of data, which are required for algorithm to work, were tuned manually within several minutes. This further boost the practical usefulness of the algorithm. Fig. 3.4a shows ML estimates for accelerometer and gyroscope data — recall that data related to q_2 are accurate and are not optimized. Fig. 3.4b shows torque prediction for benchmark model and the MLE model. Note especially excellent quality of the MLE prediction for the torque τ_1 on unactuated joint - which should be zero.



(a)



(b)

Figure 3.4 – (a) Accelerometer and gyroscope measurements and their MLE estimates. (b) Current sensor measurements, WLS model predictions of the current based on hall sensor measurements, MLE based current predictions based on accelerometer and gyroscope data.

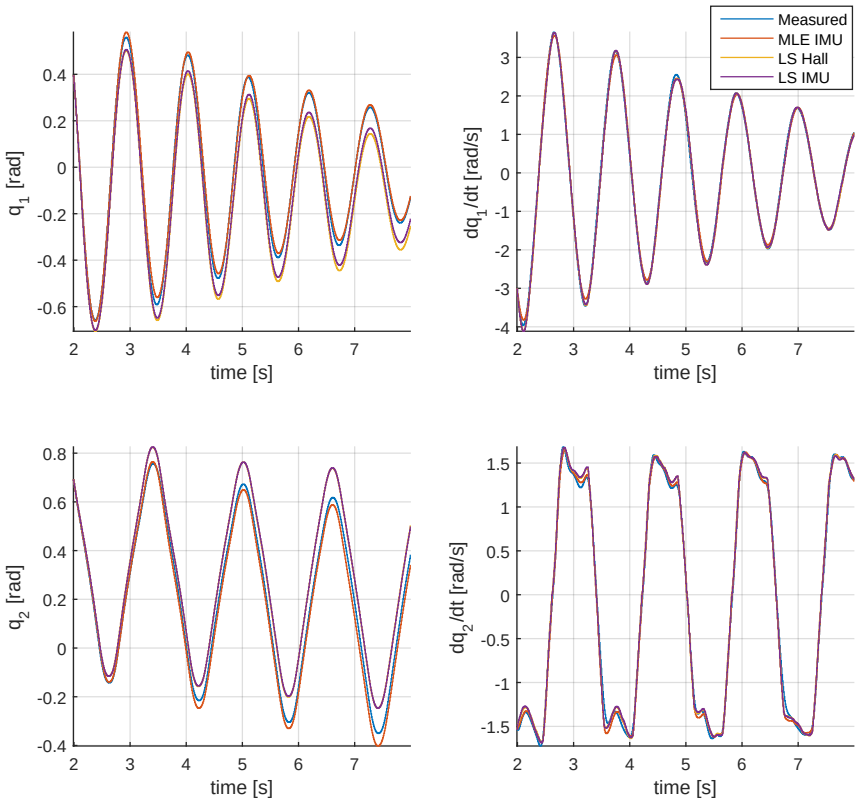


Figure 3.5 – Measurement of q and $\dot{q}$ and its predictions from various models.

4 Online state estimation

Previous chapter was dedicated to the offline estimation of the absolute-orientation angle and parameters of the robot's model. However, when the absolute orientation is required in a feedback control loop, it is necessary to estimate it online. It will be shown — similarly as in previous chapter — that various measurement instruments can be used to improve the absolute orientation estimate. The main problem is to properly detect the impact event. A closely related problem is to achieve a timely impact by lowering the swing leg at the right moment. Relative angles in combination with walking robot model provide sufficient information for estimation of the absolute angle, for experimental verification, see the work of [Lebastard et al. (2006)]. However for the practical implementation it must be possible to detect when the swing leg hits the ground and the impact event occurs. This is due to the hybrid nature of the robot model. Each time the impact event occurs the impact map is utilized and the swing phase model is reinitialized with new initial conditions. Moreover the roles of the legs are swapped. However, if the impact is not detected, then the model no longer describes the reality and the absolute angle estimator based on such a model will not work properly.

Apart from the proper detection of the time instant during which the impact occurs the quality of the absolute angle estimate has direct effect on the stability of the closed loop. Walking gait of the robot is usually encoded using the absolute angle of the stance leg. For successful walking the swing leg must be lifted from the ground into the air, high enough to avoid scuffing. To initiate the impact, the swing leg must be lowered

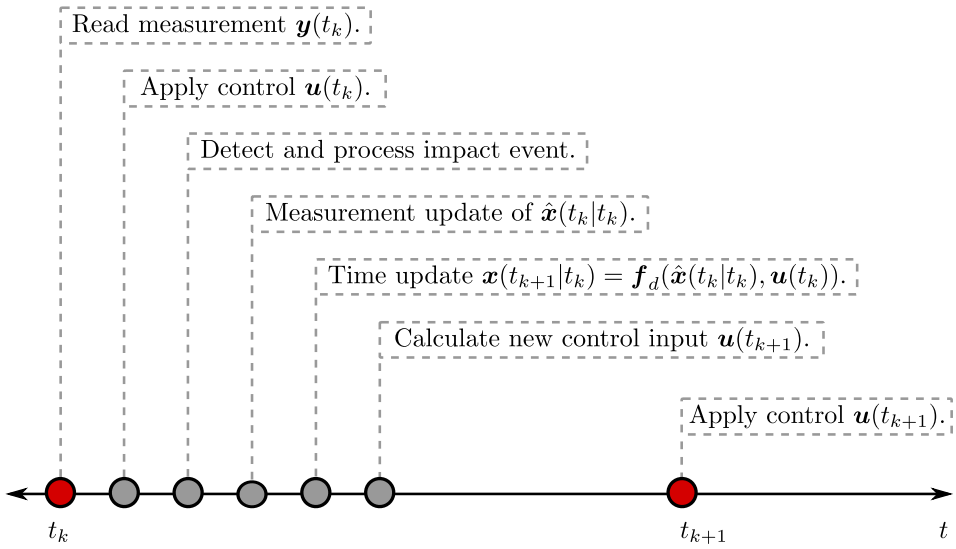


Figure 4.1 – Signal flow of closed-loop control with state estimator.

to hit the ground. Since this motion is usually encoded according to the absolute angle of the stance leg it is clear that an inaccurate estimate of the absolute angle leads to premature or delayed impact. This can cause a loss of robot's momentum and eventually lead to flipping the robot backwards and falling on the ground.

Two algorithms that are widely used for state estimation were studied — the well known Extended Kalman Filter and it's Jacobian-free alternative, the Unscented Kalman Filter.

The main goal of the estimator is to estimate the absolute orientation angle from available measurements. Basic signal flow of the closed-loop control is shown on the figure 4.1. From the figure, it is clear that the control algorithm is using predicted estimate calculated by the state estimator. The prediction of state and measurements is based on the discrete model of the robot dynamics and measurements. To detect the Impact Event (IE) several different sensors can be used. A reliable way how to detect that the leg is on the ground is to use a contact switches mounted on the bottom of the legs. An impact usually results is an abrupt change

of torso angular velocity, this can be measured using a gyroscope mounted directly on torso, as depicted on fig. 4.2. However, the gait of the robot can be designed so that the jumps in the velocities are minimized. In such an event the detection based on the gyroscope might be unreliable. For methods used to design such a trajectory see [Anderle and Čelikovský (2014)]. Another way of detection is based on the accelerometer measurements. The acceleration of the torso changes dramatically when the robot swing leg lands on the ground — this leg becomes the new stance leg and the weight of the robot is shifted to this new stance leg which acts as a new pivot, see the fig. 4.2. Mathematically this means that the model of the robot will fail to predict the measurements of the accelerometer. However, the prediction of the acceleration measurements should be radically improved by using the impact map, that is, by changing the coordinate system and reinitializing the robot's velocities. Mathematically this condition can be written as

$$e_{pa} = (\boldsymbol{y}_a - \tilde{\boldsymbol{a}}''_{ps})^T(\boldsymbol{y}_a - \tilde{\boldsymbol{a}}''_{ps}) < (\boldsymbol{y}_a - \boldsymbol{a}''_{ps})^T(\boldsymbol{y}_a - \boldsymbol{a}''_{ps}), \tag{4.1}$$

where $\boldsymbol{y}_a$ denotes the accelerometer measurements, $\boldsymbol{a}''_{ps}$ denotes proper accelerations prior to applying the impact map. Vector $\tilde{\boldsymbol{a}}''_{ps}$ denotes proper accelerations after applying the impact map. Regardless of the impact detection method, it is assumed that the impact will be successfully detected using sensors. Because the measurements are lagged one sampling time period, the information about the impact is always available T_s seconds after the impact has occurred. The model is not used to predict the impact, since this prediction is not reliable. Therefore if the impact occurs, the one step ahead prediction — calculated during the time update — is not valid and should be updated once the information about the impact is available. This update is done prior the measurement update, so that the measurement update is carried out using valid prediction. The robot estimator must store the information identifying the swing leg, so that it is possible to properly assign the encoder readings to the swing leg and the stance leg. Once the impact is detected, the encoders are reassigned accordingly.

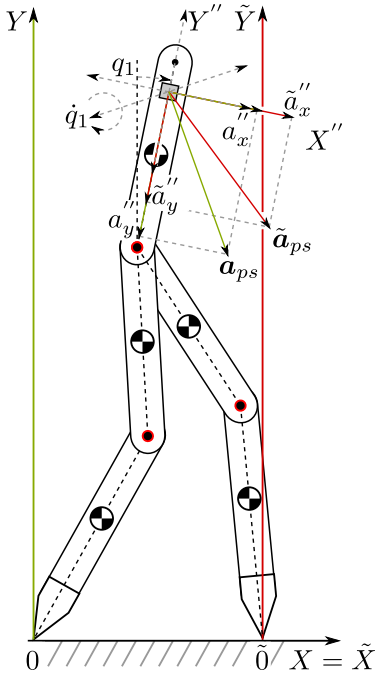


Figure 4.2 – Robot schematics with change of coordinate system.

4.1 Extended Kalman Filter

The algorithm of the Extended Kalman Filter (EKF) can be used to estimate the state of a nonlinear dynamical system [Simon (2006)]. For the application to the walking robot, the estimator must be extended with the model of impact. The impact map is assumed in following form

$$\boldsymbol{x}^+(t_k) = \Delta(\boldsymbol{x}^-(t_k), \boldsymbol{w}_I(t_k)), \tag{4.2}$$

where $\boldsymbol{w}_I(t_k) \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_I)$ are the random disturbances acting during the impact, uncorrelated with it's past samples, or with $\boldsymbol{w}(t_k)$ and $\boldsymbol{v}(t_k)$ or their past samples. The closed-loop control based on the EKF can be summarized in following steps.

1. The EKF filter is initialized with

$$\hat{\boldsymbol{x}}(t_0) = \mathbb{E}[\boldsymbol{x}(t_0)] \tag{4.3}$$
$$\mathbf{P}(t_0) = \mathbb{E}[(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))^T] \tag{4.4}$$

2. For $k = 1 \dots N$ carry out following steps.

   (a) Read the measurement $\boldsymbol{y}(t_k)$ and apply the control $\boldsymbol{u}(t_k)$.

   (b) Analyze if impact has occurred (using contact switches, gyroscope or accelerometer). In case impact has occurred, swap the encoder measurements for the swing leg and stance leg. Update the state and covariance prediction as

$$\hat{\boldsymbol{x}}(t_k|t_{k-1}) = \Delta(\hat{\boldsymbol{x}}(t_k|t_{k-1})) \tag{4.5}$$
$$\mathbf{P}(t_k|t_k) = \mathbf{F}_I(t_k)\mathbf{P}(t_k|t_{k-1})\mathbf{F}_I(t_k)^T + \mathbf{L}_I(t_k)\boldsymbol{\Sigma}_I\mathbf{L}_I(t_k)^T. \tag{4.6}$$

   where

$$\mathbf{F}_I(t_k) = \left.\frac{\partial \Delta}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_{k-1}) \\ \boldsymbol{w}=\mathbf{0}}}, \quad \mathbf{L}_I(t_k) = \left.\frac{\partial \Delta}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_{k-1}) \\ \boldsymbol{w}=\mathbf{0}}} \tag{4.7}$$

   (c) Calculate Jacobians associated with linearisation of the robot's measurement model.

$$\mathbf{H}(t_k) = \left.\frac{\partial \boldsymbol{h}_d}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_{k-1}) \\ \boldsymbol{w}=\mathbf{0}}}, \quad \mathbf{M}(t_k) = \left.\frac{\partial \boldsymbol{h}_d}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_{k-1}) \\ \boldsymbol{w}=\mathbf{0}}} \tag{4.8}$$

   (d) Measurement update: calculate the Kalman gain and the current measurement estimate

$$\mathbf{K}(t_k) = \mathbf{P}(t_k|t_{k-1})\mathbf{H}(t_k)^T(\mathbf{H}(t_k)\mathbf{P}(t_k|t_{k-1})\mathbf{H}(t_k)^T + \mathbf{M}(t_k)\boldsymbol{\Sigma}_R\mathbf{M}(t_k)^T)^{-1}, \tag{4.9}$$
$$\hat{\boldsymbol{y}}(t_k) = \boldsymbol{h}_d(\hat{\boldsymbol{x}}(t_k|t_{k-1}), \boldsymbol{u}(t_k), \mathbf{0}). \tag{4.10}$$

Perform the correction of the covariance and state estimates using new measurement.

$$\hat{\boldsymbol{x}}(t_k|t_k) = \hat{\boldsymbol{x}}(t_k|t_{k-1}) + \mathbf{K}_k(\boldsymbol{y}(t_k) - \hat{\boldsymbol{y}}(t_k)) \tag{4.11}$$

$$\mathbf{P}(t_k|t_k) = (\mathbb{1} - \mathbf{K}(t_k)\mathbf{H}(t_k))\mathbf{P}(t_k|t_{k-1}) \tag{4.12}$$

(e) Calculate Jacobians associated with linearisation of the dynamic equation of robot.

$$\mathbf{F}(t_k) = \left.\frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_k)\\ \boldsymbol{w}=\mathbf{0}}}, \quad \mathbf{L}(t_k) = \left.\frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}(t_k|t_k)\\ \boldsymbol{w}=\mathbf{0}}} \tag{4.13}$$

(f) Time update: perform the covariance update and state prediction.

$$\mathbf{P}(t_{k+1}|t_k) = \mathbf{F}(t_k)\mathbf{P}(t_k|t_k)\mathbf{F}(t_k)^T + \mathbf{L}(t_k)\boldsymbol{\Sigma}_Q\mathbf{L}(t_k)^T \tag{4.14}$$

$$\boldsymbol{x}(t_{k+1}|t_k) = \boldsymbol{f}_d(\hat{\boldsymbol{x}}(t_k|t_k), \boldsymbol{u}(t_k), \mathbf{0}) \tag{4.15}$$

(e) Calculate new control input $\boldsymbol{u}(t_{k+1})$.

During experiments it has been found that discretisation of the robot dynamics by the explicit Runge-Kutta method of fourth order leads to very complicated Jacobian matrices. And the Runge-Kutta algorithm is feasible only for the most simple case — a simple inverted pendulum — for more complex systems — even the two link robot — the Jacobians in explicit form required for the EKF algorithm are very complex. Also Jacobians $\frac{\partial \Delta}{\partial \boldsymbol{x}}$ and $\frac{\partial \Delta}{\partial \boldsymbol{w}}$ are of very high complexity, this renders the equation (4.6) very unpractical. A simple heuristic to circumvent the problem is to update the covariance $\mathbf{P}(t_k|t_{k-1})$ — in the event that the impact was detected — by a simple rule

$$\mathbf{P}(t_k|t_{k-1}) = \tilde{\mathbf{P}}(t_k|t_{k-1}) + c_I\mathbb{1}, \tag{4.16}$$

where $\tilde{\mathbf{P}}(t_k|t_{k-1})$ is obtained by swapping the entries of $\mathbf{P}(t_k|t_{k-1})$ accord-

ing to the impact change of coordinates, $\mathbb{1}$ denotes the identity matrix of appropriate dimensions and c_I is a correction factor. A drawback of the EKF is the inability to use Coulomb friction model due to its discontinuous nature. Nevertheless, the effect of the Coulomb friction is significant only during slow motion of the robot and is not important during agile walking.

4.2 Unscented Kalman Filter

The algorithm of UKF extended with the impact model can be summarized in the following steps.

1. The UKF is initialized by following estimates

$$\hat{\boldsymbol{x}}(t_0) = \mathbb{E}[\boldsymbol{x}(t_0)], \tag{4.17}$$
$$\mathbf{P}(t_0) = \mathbb{E}[(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))(\boldsymbol{x}(t_0) - \hat{\boldsymbol{x}}(t_0))^T]. \tag{4.18}$$

   For $k = 1 \dots N$ carry out following steps.
2. Read the measurement $\boldsymbol{y}(t_k)$ and apply the control $\boldsymbol{u}(t_k)$.
3. Process IE.

   (a) Analyze if impact has occurred (using contact switches, gyroscope or accelerometer). In case impact has occurred, swap the encoder measurements for the swing leg and stance leg.
   (b) Generate sigma points for the estimation of impact

$$\begin{aligned} \boldsymbol{x}^{(i)}(t_k|t_{k-1}) &= \hat{\boldsymbol{x}}(t_k|t_{k-1}) + \tilde{\boldsymbol{x}}^{(i)}(t_k) \qquad i = 1, \dots, 2n \\ \tilde{\boldsymbol{x}}^{(i)}(t_k) &= \left(\sqrt{n\mathbf{P}(t_k|t_{k-1})}\right)_i \qquad i = 1, \dots, n \\ \tilde{\boldsymbol{x}}^{(n+i)}(t_k) &= -\left(\sqrt{n\mathbf{P}(t_k|t_{k-1})}\right)_i \qquad i = 1, \dots, n \end{aligned} \tag{4.19}$$

   (c) Propagate sigma points using the robot impact model

$$\boldsymbol{x}_b^{+(i)}(t_k|t_{k-1}) = \Delta(\boldsymbol{x}^{(i)}(t_k|t_{k-1})), \qquad i = 1, \dots, 2n. \tag{4.20}$$

(d) Calculate estimate of the robot state after the impact

$$\hat{\boldsymbol{x}}^{+}(t_k|t_{k-1}) = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{x}_a^{+(i)}(t_k|t_{k-1}) + \boldsymbol{x}_b^{+(i)}(t_k|t_{k-1}). \tag{4.21}$$

(e) Estimate the covariance of the state estimate after the impact

$$\mathbf{P}_x^{+}(t_k|t_{k-1}) = \mathbf{P}_{x,a}^{+}(t_k|t_{k-1}) + \mathbf{P}_{x,b}^{+}(t_k|t_{k-1}) + \boldsymbol{\Sigma}_I, \tag{4.22}$$

$$\mathbf{P}_{x,a}^{+}(t_k|t_{k-1}) = \frac{1}{2n}\sum_{i=1}^{2n} \Bigg(\Big(\boldsymbol{x}_a^{+(i)}(t_k|t_{k-1}) - \hat{\boldsymbol{x}}^{+}(t_k|t_{k-1})\Big) \Big(\boldsymbol{x}_a^{+(i)}(t_k|t_{k-1}) - \hat{\boldsymbol{x}}^{+}(t_k|t_{k-1})\Big)^T \Bigg),$$

$$\mathbf{P}_{x,b}^{+}(t_k|t_{k-1}) = \frac{1}{2n}\sum_{i=1}^{2n} \Bigg(\Big(\boldsymbol{x}_b^{+(i)}(t_k|t_{k-1}) - \hat{\boldsymbol{x}}^{+}(t_k|t_{k-1})\Big) \Big(\boldsymbol{x}_b^{+(i)}(t_k|t_{k-1}) - \hat{\boldsymbol{x}}^{+}(t_k|t_{k-1})\Big)^T \Bigg).$$

4. Measurement update.

(a) Generate sigma points for the estimation of output:

$$\boldsymbol{x}^{(i)}(t_k|t_{k-1}) = \hat{\boldsymbol{x}}(t_k|t_{k-1}) + \tilde{\boldsymbol{x}}^{(i)}(t_k) \qquad i = 1, \ldots, 2n \tag{4.23}$$

$$\tilde{\boldsymbol{x}}^{(i)}(t_k) = \left(\sqrt{n\mathbf{P}(t_k|t_{k-1})}\right)_i \qquad i = 1, \ldots, n \tag{4.24}$$

$$\tilde{\boldsymbol{x}}^{(n+i)}(t_k) = -\left(\sqrt{n\mathbf{P}(t_k|t_{k-1})}\right)_i \qquad i = 1, \ldots, n \tag{4.25}$$

(b) Propagate sigma points using the robot measurement model

$$\boldsymbol{y}^{(i)}(t_k) = \boldsymbol{h}_d(\boldsymbol{x}^{(i)}(t_k), \boldsymbol{u}(t_k)). \tag{4.26}$$

(c) Calculate current estimate of the outputs as

$$\hat{\boldsymbol{y}}(t_k) = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{y}^{(i)}(t_k). \tag{4.27}$$

(d) Estimate the covariance of the predicted measurements

$$\mathbf{P}_y(t_k) = \frac{1}{2n}\sum_{i=1}^{2n}\Bigg(\Big(\boldsymbol{y}^{(i)}(t_k) - \hat{\boldsymbol{y}}(t_k)\Big) \Big(\boldsymbol{y}^{(i)}(t_k) - \hat{\boldsymbol{y}}(t_k)\Big)^T\Bigg) + \boldsymbol{\Sigma}_R. \tag{4.28}$$

(e) Estimate the cross-covariance between $\hat{\boldsymbol{x}}(t_k|t_{k-1})$ and $\hat{\boldsymbol{y}}(t_k)$

$$\mathbf{P}_{xy}(t_k) = \frac{1}{2n}\sum_{i=1}^{2n}\Bigg(\Big(\boldsymbol{x}^{(i)}(t_k) - \hat{\boldsymbol{x}}(t_k|t_{k-1})\Big) \Big(\boldsymbol{y}^{(i)}(t_k) - \hat{\boldsymbol{y}}(t_k)\Big)^T\Bigg). \tag{4.29}$$

(e) Perform the measurement update

$$\mathbf{K}(t_k) = \mathbf{P}_{xy}(t_k)\mathbf{P}_y^{-1}(t_k) \tag{4.30}$$
$$\hat{\boldsymbol{x}}(t_k|t_k) = \hat{\boldsymbol{x}}(t_k|t_{k-1}) + \mathbf{K}(t_k)(\boldsymbol{y}(t_k) - \hat{\boldsymbol{y}}(t_k)) \tag{4.31}$$
$$\mathbf{P}(t_k|t_k) = \mathbf{P}(t_k|t_{k-1}) - \mathbf{K}(t_k)\mathbf{P}_y(t_k)\mathbf{K}(t_k)^T \tag{4.32}$$

3. Time update.

(a) Generate sigma points for one-step-ahead state prediction:

$$\boldsymbol{x}^{(i)}(t_k) = \hat{\boldsymbol{x}}(t_k|t_k) + \tilde{\boldsymbol{x}}^{(i)}(t_k) \qquad i = 1, \ldots, 2n \tag{4.33}$$
$$\tilde{\boldsymbol{x}}^{(i)}(t_k) = \left(\sqrt{n\mathbf{P}(t_k|t_k)}\right)_i^T \qquad i = 1, \ldots, n \tag{4.34}$$
$$\tilde{\boldsymbol{x}}^{(n+i)}(t_k) = -\left(\sqrt{n\mathbf{P}(t_k|t_k)}\right)_i^T \qquad i = n+1, \ldots, 2n \tag{4.35}$$

(b) Propagate sigma points using the robot model (2.29), that is

$$\boldsymbol{x}^{(i)}(t_{k+1}) = \boldsymbol{f}_d(\hat{\boldsymbol{x}}^{(i)}(t_k), \boldsymbol{u}(t_k)). \tag{4.36}$$

(c) Calculate one-step-ahead prediction of the state as

$$\hat{\boldsymbol{x}}(t_{k+1}|t_k) = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{x}^{(i)}(t_{k+1}). \tag{4.37}$$

(d) Estimate the covariance of the one-step-ahead state estimate

$$\mathbf{P}(t_{k+1}|t_k) = \frac{1}{2n}\sum_{i=1}^{2n} \left(\left(\boldsymbol{x}^{(i)}(t_{k+1}) - \hat{\boldsymbol{x}}(t_{k+1}|t_k)\right) \left(\boldsymbol{x}^{(i)}(t_{k+1}) - \hat{\boldsymbol{x}}(t_{k+1}|t_k)\right)^T \right) + \boldsymbol{\Sigma}_Q \tag{4.38}$$

(d) Reuse the sigma points for the potential estimation of impact and propagate sigma points using the robot impact model

$$\boldsymbol{x}^{(i+)}(t_k) = \Delta(\hat{\boldsymbol{x}}^{(i-)}(t_k)). \tag{4.39}$$

(b) Propagate sigma points using the robot model (2.29) for potential estimation of impact

$$\boldsymbol{x}_a^{(i)}(t_{k+1}) = \boldsymbol{f}_d(\hat{\boldsymbol{x}}^{(i)}(t_k), \boldsymbol{u}(t_k)). \tag{4.40}$$

4.3 Application of EKF and UKF to absolute angle estimation for simulated three-link bipedal walking robot

This subsection is devoted to a simulation study and comparison of the EKF and UKF estimation algorithms. Both algorithms were used to estimate the absolute angle of the three-link walking robot depicted on the fig. 4.3. The estimation algorithms were tested in closed-loop with a nonlinear controller. The controller was originally designed in [Grizzle

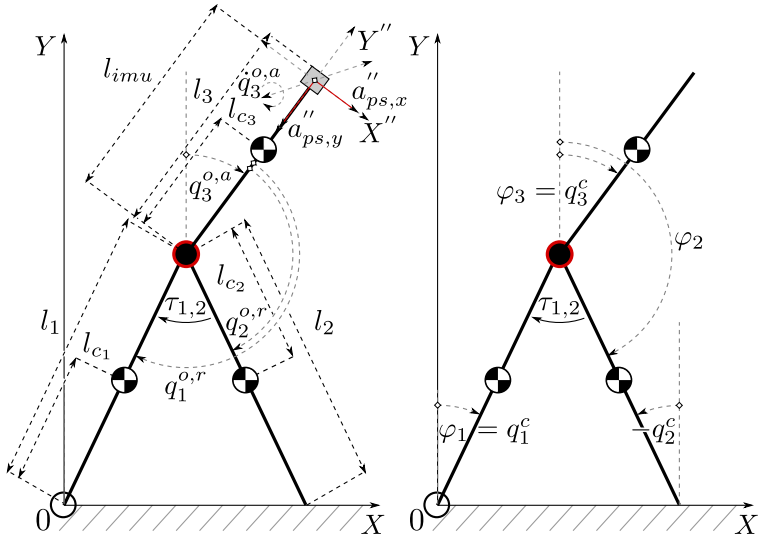


Figure 4.3 – Definition of configuration angles: (Left) EKF/UKF design. (Right) Controller design.

et al. (2001)]. The controller was adopted here for a three-link model of walking robot. The parameters of the three-link model are based on five-link laboratory prototype described in [Anderte et al. (2015)]. The three-link model is composed of two legs and a torso. Due to absence of knees the robot model is unable to flex the legs during the walking cycle. This results in premature contact of the swing leg with the ground. A solution to this problem was proposed in [McGeer (1990)], where the author suggests to mount small flaps on the end of each leg, and fold the flap of the swing leg to avoid the contact and unfold it when the contact should be initiated. In the following simulation example the solution based on foldable flaps is simulated. The three-link model is a simplification of more complicated model which describes also the movement of knees. However, three-link model comprises all essential challenges which needs to be addressed during the design of estimator of the absolute-orientation angle for a bipedal underactuated walking robot. The key problems that needs to be addressed are: complicated nonlinear dynamics with fast and unstable behavior and a hybrid model resulting from contact with the ground. The estimation of the absolute-orientation angle must be of high

Table 4.1 – Parameters of the 3-link robot

l_1, l_2, l_3	length of 1st, 2nd and 3rd link	$[m]$
l_{c_1}, l_{c_2}, l_{c_3}	center of gravity of 1st, 2nd and 3rd link	$[m]$
m_1, m_2, m_3	mass of 1st, 2nd and 3rd link	$[kg]$
I_1, I_2, I_3	inertia of 1st, 2nd and 3rd link	$[kg.m^2]$
τ_1, τ_2	torque generated by 1st and 2nd motor	$[N.m]$
g	gravitational acceleration	$[m.s^{-2}]$

quality since whole motion of the robot is encoded using this information. An imprecise estimate leads to loss of robot's momentum and eventually destabilizes the feedback control loop. The closed-loop works with two sets of configuration angles. One set — denoted as $\boldsymbol{q}^o$ — is defined with respect to sensors and this set is used for the design of EKF and UKF observers, since the dynamical model based on this configuration directly predicts the measurements taken by the sensors. The nonlinear controller is based on a different set of configuration angles denoted as $\boldsymbol{q}^c$. This set results in less complicated matrices of the robot model and the use of the particular set of configuration angles results in more natural limit cycle. Physical parameters of the robot are listed in tab. 4.1. Values of physical parameters of the robot are given in the following table

$$\begin{aligned} l_1 &= 0.535, & l_2 &= 0.535, & l_3 &= 0.25, \\ l_{c_1} &= 0.2675, & l_{c_2} &= 0.2675, & l_{c_3} &= 0.1189, \\ m_1 &= 0.4192, & m_2 &= 0.4192, & m_3 &= 1.1646, \\ I_1 &= 0.0034, & I_2 &= 0.0034, & I_3 &= 0.0182\,. \end{aligned} \tag{4.41}$$

4.3.1 Measurement and dynamical models used for observer design

Configuration angles are defined by two relative angles $q_1^{o,r}$, $q_2^{o,r}$ and one absolute angle $q_3^{o,a}$. These angles are defined in accordance with the sen-

sors used for measurements. In this simulation example relative angles $q_1^{o,r}$, $q_2^{o,r}$ are assumed to be measured using optical incremental sensors. Angular velocity $\dot{q}_3^{o,a}$ of the torso can be directly measured using a gyroscope — which is a part of an IMU mounted on the torso, as depicted on fig. 4.3. Where l_{imu} denotes the length from the hips to the IMU. Finally the IMU also includes a planar two axis accelerometer which provides the measurements of the proper acceleration of the torso in the coordinates of the IMU. Measurement model is given as

$$\boldsymbol{y}_{q^{o,r}} = \boldsymbol{q}^{o,r} + \boldsymbol{e}_{q^{o,r}} = [q_1^{o,r}, q_2^{o,r}]^T + [e_{q_1^{o,r}}, e_{q_2^{o,r}}]^T \tag{4.42}$$

$$y_{\dot{q}_3^{o,a}} = \dot{q}_3^{o,a} + e_{\dot{q}_3^{o,a}} \tag{4.43}$$

$$\boldsymbol{y}_a = \boldsymbol{a}_{ps}^{''} + \boldsymbol{e}_{a_{ps}^{''}} = [a_{ps,x}^{''}, a_{ps,y}^{''}]^T + [e_{a_{ps,x}^{''}}, e_{a_{ps,y}^{''}}]^T. \tag{4.44}$$

Errors of the measurement are all normally distributed, uncorrelated across time, $\boldsymbol{e}_{q^{o,r}} \sim \mathcal{N}(0, \mathbf{R}_r)$, $e_{\dot{q}_3^{o,a}} \sim \mathcal{N}(0, \mathbf{R}_{gyro})$, $\boldsymbol{e}_{a_{ps}^{''}} \sim \mathcal{N}(0, \mathbf{R}_{acc})$. The proper acceleration measured by the accelerometer is obtained using

$$\boldsymbol{a}_{ps}^{''} = \mathbf{R}^{-}(\varphi_3) \left(l_1 \mathbf{R}^{+}(\varphi_1) \begin{bmatrix} \ddot{\varphi}_1 \\ \dot{\varphi}_1^2 \end{bmatrix} + l_{imu} \mathbf{R}^{+}(\varphi_3) \begin{bmatrix} \ddot{\varphi}_3 \\ \dot{\varphi}_3^2 \end{bmatrix} - \begin{bmatrix} 0 \\ g \end{bmatrix} \right). \tag{4.45}$$

where φ_1 and φ_3 are absolute angles defined as on the right part of the fig. 4.3. These angles are defined as

$$\varphi_1 = q_1^{o,r} + q_3^{o,a} - \pi, \tag{4.46}$$

$$\varphi_2 = q_2^{o,r} + q_3^{o,a}, \tag{4.47}$$

$$\varphi_3 = q_3^{o,a}. \tag{4.48}$$

Note that the matrix $\mathbf{R}^{+}(\varphi_1)$ can be written as

$$\mathbf{R}^{+}(\varphi_1) = \begin{bmatrix} \cos(\varphi_1) & \sin(\varphi_1) \\ -\sin(\varphi_1) & \cos(\varphi_1) \end{bmatrix} =$$

$$= - \begin{bmatrix} \cos(q_1^{o,r}) & \sin(q_1^{o,r}) \\ -\sin(q_1^{o,r}) & \cos(q_1^{o,r}) \end{bmatrix} \begin{bmatrix} \cos(q_3^{o,a}) & \sin(q_3^{o,a}) \\ -\sin(q_3^{o,a}) & \cos(q_3^{o,a}) \end{bmatrix}. \tag{4.49}$$

Matrices $\mathbf{R}^+(\varphi_3)$ and $\mathbf{R}^-(\varphi_3)$ are given as (3.25). The components of the proper acceleration obtained in the accelerometer coordinate system X'', Y'' are obtained by substituting (4.49) and (3.25) into (4.45), resulting in

$$\begin{bmatrix} a''_{ps,x} \\ a''_{ps,y} \end{bmatrix} = -l_1 \begin{bmatrix} \cos(q_1^{o,r}) & \sin(q_1^{o,r}) \\ -\sin(q_1^{o,r}) & \cos(q_1^{o,r}) \end{bmatrix} \begin{bmatrix} (\ddot{q}_1^{o,r} + \ddot{q}_3^{o,a}) \\ -(\dot{q}_1^{o,r} + \dot{q}_3^{o,a})^2 \end{bmatrix} + \\ + l_{imu} \begin{bmatrix} \ddot{q}_3^{o,a} \\ -(\dot{q}_3^{o,a})^2 \end{bmatrix} - \begin{bmatrix} \cos(q_3^{o,a}) & -\sin(q_3^{o,a}) \\ \sin(q_3^{o,a}) & \cos(q_3^{o,a}) \end{bmatrix} \begin{bmatrix} 0 \\ g \end{bmatrix}. \tag{4.50}$$

The model of the robot defined by configuration angles $\boldsymbol{q}^o$ is given as

$$\boldsymbol{D}_o(\boldsymbol{q}^o)\ddot{\boldsymbol{q}}^o + \boldsymbol{C}_o(\boldsymbol{q}^o, \dot{\boldsymbol{q}}^o)\dot{\boldsymbol{q}}^o + \boldsymbol{G}_o(\boldsymbol{q}^o) = \boldsymbol{B}_o\boldsymbol{u} \tag{4.51}$$

where the entries of the matrix $\mathbf{D}_o$ are given as

$$\begin{aligned} D_{o,1,1} &= \beta_1, \\ D_{o,1,2} &= -\beta_4 \cos(q_1^{o,r} - q_2^{o,r}), \\ D_{o,1,3} &= \beta_1 - \beta_5 \cos(q_1^{o,r}) - \beta_4 \cos(q_1^{o,r} - q_2^{o,r}), \\ D_{o,2,2} &= \beta_2, \\ D_{o,2,3} &= \beta_2 - \beta_4 \cos(q_1^{o,r} - q_2^{o,r}), \\ D_{o,3,3} &= \beta_1 + \beta_2 + \beta_3 - 2\beta_5 \cos(q_1^{o,r}) - 2\beta_4 \cos(q_1^{o,r} - q_2^{o,r}) \end{aligned} \tag{4.52}$$

and the remaining entries are completed by symmetry. The elements of matrix $\mathbf{C}_o$ are as follows

$$\begin{aligned} C_{o,1,1} &= 0, \\ C_{o,1,2} &= -\beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_2^{o,r} + \dot{q}_3^{o,a}), \\ C_{o,1,3} &= -\beta_5 \sin(q_1^{o,r})\dot{q}_3^{o,a} - \beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_2^{o,r} + \dot{q}_3^{o,a}) \\ C_{o,2,1} &= \beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_1^{o,r} + \dot{q}_3^{o,a}), \\ C_{o,2,2} &= 0, \end{aligned}$$

$$
\begin{aligned}
C_{o,2,3} &= \beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_1^{o,r} + \dot{q}_3^{o,a}) \\
C_{o,3,1} &= (\beta_5 \sin(q_1^{o,r}) + \beta_4 \sin(q_1^{o,r} - q_2^{o,r}))(\dot{q}_1^{o,r} + \dot{q}_3^{o,a}), \\
C_{o,3,2} &= -\beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_2^{o,r} + \dot{q}_3^{o,a}), \\
C_{o,3,1} &= \beta_5 \sin(q_1^{o,r})\dot{q}_1^{o,r} + \beta_4 \sin(q_1^{o,r} - q_2^{o,r})(\dot{q}_1^{o,r} - \dot{q}_2^{o,r})
\end{aligned} \tag{4.53}
$$

Vector describing gravity effects $\boldsymbol{G}_o$ and matrix mapping motor torques to joint torques $\mathbf{B}_o$ are given as follows

$$
\boldsymbol{G}_o = \begin{bmatrix} g\beta_6 \sin(q_1^{o,r} + q_3^{o,a}) \\ -g\beta_7 \sin(q_2^{o,r} + q_3^{o,a}) \\ g(\beta_6 \sin(q_1^{o,r} + q_3^{o,a}) - \beta_7 \sin(q_2^{o,r} + q_3^{o,a}) - \beta_8 \sin(q_3^{o,a})) \end{bmatrix}, \tag{4.54}
$$

$$
\boldsymbol{B}_o = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}. \tag{4.55}
$$

The parameter substitution is identical with (3.81) and is given as

$$
\begin{aligned}
\beta_1 &= I_1 + l_1^2 m_2 + l_1^2 m_3 + l_{c_1}^2 m_1, & \beta_2 &= I_2 + m_2 l_{c_2}^2, \\
\beta_3 &= I_3 + m_3 l_{c_3}^2, & \beta_4 &= l_1 l_{c_2} m_2, \\
\beta_5 &= l_1 l_{c_3} m_3, & \beta_6 &= (l_1 m_2 + l_1 m_3 + l_{c_1} m_1), \\
\beta_7 &= l_{c_2} m_2, & \beta_8 &= l_{c_3} m_3, \\
\beta_9 &= m_1 + m_2 + m_3. &&
\end{aligned} \tag{4.56}
$$

Note that parameter β_9 is used only in the unpinned model of the robot. The vector field corresponding to the continuous state-space model is

$$
\boldsymbol{f}_o = \begin{bmatrix} \dot{\boldsymbol{q}}^o \\ \mathbf{D}_o^{-1}(-\mathbf{C}_o \dot{\boldsymbol{q}}^o - \boldsymbol{G}_o + \mathbf{B}_o \boldsymbol{u}) \end{bmatrix}. \tag{4.57}
$$

The associated EKF and UKF estimators are reinitialized each time an impact event is detected. The detection itself can be done using several methods as was written at the beginning of the chapter 4. The impact

map is calculated as described in section 2.2. The additional entries for the extended matrix associated with the unpinned 5DoF model of the 3-link robot required for the calculation of the impact map are given as

$$
\begin{aligned}
D_{oe,4,1} &= -\beta_6 \cos(q_1^{o,r} + q_3^{o,a}) \\
D_{oe,4,2} &= \beta_7 \cos(q_2^{o,r} + q_3^{o,a}) \\
D_{oe,4,3} &= \beta_8 \cos(q_3^{o,a}) - \beta_6 \cos(q_1^{o,r} + q_3^{o,a}) + \beta_7 \cos(q_2^{o,r} + q_3^{o,a}) \\
D_{oe,4,4} &= \beta_9 \\
D_{oe,4,5} &= 0 \\
D_{oe,5,1} &= \beta_6 \sin(q_1^{o,r} + q_3^{o,a}) \\
D_{oe,5,2} &= -\beta_7 \sin(q_2^{o,r} + q_3^{o,a}) \\
D_{oe,5,3} &= \beta_6 \sin(q_1^{o,r} + q_3^{o,a}) - \beta_7 \sin(q_2^{o,r} + q_3^{o,a}) - \beta_8 \sin(q_3^{o,a}) \\
D_{oe,5,5} &= \beta_9
\end{aligned} \tag{4.58}
$$

and the remaining entries are completed by symmetry. The impact results in discontinuous change in angular velocities. That is, the state vector $\boldsymbol{x}^{o,-} = [q_1^{o,r}, q_2^{o,r}, q_3^{o,a}, \dot{q}_1^{o,r,-}, \dot{q}_2^{o,r,-}, \dot{q}_3^{o,a,-}]^T$ just before the impact will be mapped to a new state $\boldsymbol{x}^{o,+} = [q_1^{o,r}, q_2^{o,r}, q_3^{o,a}, \dot{q}_1^{o,r,+}, \dot{q}_2^{o,r,+}, \dot{q}_3^{o,a,+}]^T$, which represents the state of the robot just after the impact. Finally, the leg swapping is accomplished by the following transformation

$$
\begin{aligned}
\tilde{q}_1^{o,r} &= q_2^{o,r}, & \dot{\tilde{q}}_1^{o,r} &= \dot{q}_2^{o,r+}, \\
\tilde{q}_2^{o,r} &= q_1^{o,r}, & \dot{\tilde{q}}_2^{o,r} &= \dot{q}_1^{o,r+}, \\
\tilde{q}_3^{o,a} &= q_3^{o,a}, & \dot{\tilde{q}}_3^{o,a} &= \dot{q}_3^{o,a+},
\end{aligned} \tag{4.59}
$$

which essentially states that the relative angles are swapped and the absolute angle remains the same. The relative velocities are swapped accordingly. The new state vector $\tilde{\boldsymbol{x}}^o = [\tilde{q}_1^{o,r}, \tilde{q}_2^{o,r}, \tilde{q}_3^{o,a}, \dot{\tilde{q}}_1^{o,r,+}, \dot{\tilde{q}}_2^{o,r,+}, \dot{\tilde{q}}_3^{o,a,+}]^T$ will be used to reinitialize the model for the EKF or the UKF estimator. Finally, to obtain the state vector in the controller configuration angles $\boldsymbol{q}^c$, following transformation is required

$$
\begin{aligned}
q_1^c &= q_1^{o,r} + q_3^{o,a} - \pi, & \dot{q}_1^c &= \dot{q}_1^{o,r} + \dot{q}_3^{o,a}, \\
q_2^c &= q_2^{o,r} + q_3^{o,a} - \pi, & \dot{q}_2^c &= \dot{q}_2^{o,r} + \dot{q}_3^{o,a},
\end{aligned}
$$

$$q_3^c = q_3^{o,a}, \qquad \dot{q}_3^c = \dot{q}_3^{o,a}. \tag{4.60}$$

4.3.2 Dynamical model for controller design

Using the Lagrangian formalism and configuration angles $\boldsymbol{q}_c$ leads to the following model

$$\mathbf{D}_c(\boldsymbol{q}^c)\ddot{\boldsymbol{q}}^c + \mathbf{C}_c(\boldsymbol{q}^c, \dot{\boldsymbol{q}}^c)\dot{\boldsymbol{q}}^c + \mathbf{G}_c(\boldsymbol{q}^c) = \mathbf{B}_c\boldsymbol{u}, \tag{4.61}$$

where $\boldsymbol{u}$ are the torques generated by actuators and the matrices $\mathbf{D}_c$, $\mathbf{C}_c$, $\mathbf{G}_c$ and $\mathbf{B}_c$ are defined as follows

$$\begin{aligned}
\mathbf{D}_c &= \begin{bmatrix} \beta_1, & \beta_4 \mathrm{c}_{1,2}^{c-}, & \beta_5 \mathrm{c}_{1,3}^{c-} \\ \beta_4 \mathrm{c}_{1,2}^{c-}, & \beta_2, & 0 \\ \beta_5 \mathrm{c}_{1,3}^{c-}, & 0, & \beta_3 \end{bmatrix} \\
\mathbf{C}_c &= \begin{bmatrix} 0, & -\beta_4 \mathrm{s}_{1,2}^{c-}\dot{q}_2^c, & \beta_5 \mathrm{s}_{1,3}^{c-}\dot{q}_3^c \\ \beta_4 \mathrm{s}_{1,2}^{c-}\dot{q}_1^c, & 0, & 0 \\ -\beta_5 \mathrm{s}_{1,3}^{c-}\dot{q}_1^c, & 0, & 0 \end{bmatrix} \\
\mathbf{G}_c &= \begin{bmatrix} -g\beta_6 \sin(q_1^c) \\ g\beta_7 \sin(q_2^c) \\ -g\beta_8 \sin(q_3^c) \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix},
\end{aligned} \tag{4.62}$$

with abbreviations $\mathrm{s}_{1,2}^{c-} = \sin(q_1^c - q_2^c)$, $\mathrm{s}_{1,3}^{c-} = \sin(q_1^c - q_3^c)$, $\mathrm{c}_{1,2}^{c-} = \cos(q_1^c - q_2^c)$ and $\mathrm{c}_{1,3}^{c-} = \cos(q_1^c - q_3^c)$. The vector fields corresponding to the continuous state-space model used for controller design is

$$\boldsymbol{f}_c = \begin{bmatrix} \dot{\boldsymbol{q}}^c \\ \mathbf{D}_c^{-1}(-\mathbf{C}_c\dot{\boldsymbol{q}}^c - \boldsymbol{G}_c) \end{bmatrix}, \qquad \boldsymbol{g}_c = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_c^{-1}(\mathbf{B}_c\boldsymbol{u}) \end{bmatrix}. \tag{4.63}$$

4.3.3 Generating a stable walking gait

A steady robotic walking can be viewed as a periodic motion. A controller should ensure that the robot enters a stable limit cycle after the transient dies out. Unfortunately, the analysis of stability of a limit cycles for such a

complex nonlinear system as a walking robot is not trivial. To simplify the analysis, the system is linearized using output feedback linearization. This method requires a special set of outputs to be specified. These outputs — in robotics also called virtual constraints [Westervelt et al. (2007)] — are asymptotically zeroed via the feedback resulting in a reduction of degrees of freedom, which usually also amounts to achieving the desired control task. The reduced system — called zero dynamics — can be used to asses the stability of the overall system if certain conditions are met, for an authoritative account on this method see [A. Isidori (1999)]. The number of virtual constrains that can be asymptotically satisfied is equal to the number of independent actuators. A three-link robot has three DoF and two actuators, therefore the reduced uncontrollable system has only one degree of freedom. Based on the zero dynamics and the method of Poincaré [Khalil (2002)] one can design a feedback controller that forces the robot to enter a stable limit cycle if the robot has sufficient initial momentum [Westervelt et al. (2007)], [Grizzle et al. (2001)].

Controller will be designed using $\boldsymbol{q}^c$ configuration angles due to following reasons. The matrices of the robot model are less complicated and the design procedure of the controller is described in detail in detail in [Grizzle et al. (2001)] for the model defined using $\boldsymbol{q}^c$. Moreover, the resulting limit cycle is more natural than the limit cycle generated by the controller designed in the coordinates of the observer. Also, note that the parametrisation of the robot model does not change with different sets of configuration angles. This shows that identification methods from section 3 can be used for redesign of the controller.

Virtual constraints encoding a simple walking pattern can be defined as

$$\boldsymbol{w}^c := \begin{bmatrix} w_1^c \\ w_2^c \end{bmatrix} := \begin{bmatrix} h_1^c(\boldsymbol{q}^c) \\ h_2^c(\boldsymbol{q}^c) \end{bmatrix} := \begin{bmatrix} q_3^c - q_3^{c,d} \\ q_2^c + q_1^c \end{bmatrix}, \tag{4.64}$$

where the virtual constraints $\boldsymbol{w}^c$ are functions of configuration angles. By zeroing the constraint w_1^c robot will maintain the desired torso orientation denoted as $q_3^{a,d}$. Zeroing the second constraint ensures mirroring of the

legs orientation. Direct computation yields

$$\ddot{\boldsymbol{w}}^c = L_{f_c}^2 \boldsymbol{h}^c + L_{g_c} L_{f_c} \boldsymbol{h}^c \boldsymbol{u}, \tag{4.65}$$

where L_{f_c} and L_{g_c} denote Lie derivatives along the vector fields $\boldsymbol{f}_c$ and $\boldsymbol{g}_c$ respectively. By defining a decoupling control

$$\boldsymbol{v}^c := L_{f_c}^2 \boldsymbol{h}^c + L_{g_c} L_{f_c} \boldsymbol{h}^c \boldsymbol{u} \tag{4.66}$$

and a local co-ordinate transformation

$$\Phi(\boldsymbol{q}^c) := \begin{bmatrix} w_1^c \\ w_2^c \\ q_1^c \end{bmatrix} := \begin{bmatrix} q_3^c - q_3^{c,d} \\ q_2^c + q_1^c \\ q_1^c \end{bmatrix}, \tag{4.67}$$

the system can be written in a decoupled form [Grizzle et al. (2001)]

$$\begin{bmatrix} \ddot{\boldsymbol{w}}^c \\ \ddot{q}_1^c \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^c \\ \zeta_0(\boldsymbol{w}^c, \dot{\boldsymbol{w}}^c, q_1^c, \dot{q}_1^c) + \boldsymbol{\zeta}_1(\boldsymbol{w}^c, \dot{\boldsymbol{w}}^c, q_1^c, \dot{q}_1^c)^T \boldsymbol{v}^c \end{bmatrix}. \tag{4.68}$$

Where $\boldsymbol{v}^c$ denotes the virtual inputs. With the application of following feedback control

$$v_i^c = \frac{1}{\epsilon} \left(-\text{sign}(\epsilon \dot{q}_i^c) |\epsilon \dot{q}_i^c|^\alpha - \text{sign}(\phi_i) |\phi_i|^{\alpha/2-\alpha} \right), \tag{4.69}$$

where $0 < \alpha < 1$ and

$$\phi_i = q_i^c + (\frac{1}{2} - \alpha) \text{sign}(\epsilon \dot{q}_i^c) |\epsilon \dot{q}_i^c|^{2-\alpha} \tag{4.70}$$

it can be shown that for the design parameters $\epsilon = 0.1$, $\alpha = 0.9$, $q_3^{c,d} = \pi/6$ and robot model parameters (4.41), the generated limit cycle is stable, provided robot starts with sufficient momentum.

4.3.4 Simulation results

This subsection contains the simulation results based on the three-link model described in the section 4.3. The simulation is focused on evaluating

the performance of the absolute angle estimation using the EKF and UKF estimation algorithms, described in sections 4.1 and 4.2. The data are collected during the closed-loop control using the controller described in subsection 4.3.3. The initial conditions of the robot are set so that it has sufficient momentum to advance in the forward direction. The initial state of the EKF and UKF estimators is set so that there is a 10% error in the estimates of angles and all the estimates of velocities are set to 0. These initial conditions are set in this way because in order to get the closed-loop working the robot must be pushed forward and it is usually not known in advance what the velocities will be. The initial state and its initial estimate are set to

$$\boldsymbol{x}(t_0) = \begin{bmatrix} 2.65, & 3.43, & 0.1, & 1.5, & -2.5, & 0.5 \end{bmatrix}^T, \tag{4.71}$$

$$\hat{\boldsymbol{x}}(t_0) = \begin{bmatrix} 2.38, & 3.1, & 0, & 0, & 0, & 0 \end{bmatrix}^T. \tag{4.72}$$

The initial covariance $\mathbf{P}(t_0)$ of the estimate is set to identity matrix of appropriate dimensions. The duration of the simulation T is set to 5 $[s]$. During this time the robot makes several steps. The sampling time T_s of measurements was 0.01, resulting in $N = 500$ samples per one simulation run. Integration time T_i for the Euler's method was set to $T_i = T_s/8$ and for the Runge-Kutta method to $T_i = T_s/2$. Four different setups of sensors are studied. In the first setup, only relative angles are available, this corresponds to use of IRC sensors only. The second setup uses IRC sensors and additionally also a gyroscope to measure the angular velocity of the torso. This setup will be denoted as IRC & gyroscope. The third setup uses IRC sensors and additionally also an accelerometer mounted on the torso to measure proper accelerations of the torso in it's body coordinates. This setup will be referred to as IRC & accelerometer. Finally the fourth setup assumes the usage of all of the previous sensors in conjunction, that is IRC sensors, the gyroscope and the accelerometer. The setup is denoted as IRC & IMU. The covariance matrices of measurement errors for particular sensors were set to

$$\boldsymbol{\Sigma}_{irc} = 10^{-10}\mathbb{1}_{2\times 2}, \qquad \Sigma_{gyro} = 10^{-2}, \qquad \boldsymbol{\Sigma}_{acc} = 10^{-2}\mathbb{1}_{2\times 2}, \tag{4.73}$$

where $\mathbf{\Sigma}_{irc}$ denotes the variance of the error in the measurements of the IRC sensors. Variance of the gyroscope measurement errors is denoted as Σ_{gyro}. And the variance of the accelerometer measurement errors is denoted as $\mathbf{\Sigma}_{acc}$. The measurements are generated as described is section 4.3.1. The covariance matrix of the random disturbances acting on the robot is given by diagonal matrix

$$\mathbf{\Sigma}_Q = \text{diag}[10^{-3}, 10^{-3}, 10^{-3}, 10^{-2}, 10^{-2}, 10^{-2}]. \tag{4.74}$$

The covariance of the impact for the case of the EKF algorithm was calculated using the heuristic (4.16) with $c_I = 0.0075$ tuned manually. The performance of the EKF and UKF algorithms was evaluated using 50 MC simulations for each setup of sensors. The following performance statistics were used for comparison. The vector $\bar{\bar{\boldsymbol{s}}}$ denotes the sample mean of total error for each state variable. Vector $\bar{\bar{\boldsymbol{s}}} = [\bar{\bar{s}}_1, \ldots, \bar{\bar{s}}_{n_x}]$, where n_x is the number of states. Variable $\bar{\bar{\boldsymbol{s}}}$ is the average of the summed squared errors of all simulation runs for a particular sensor setup and is calculated as

$$\bar{\bar{\boldsymbol{s}}} = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \bar{\boldsymbol{s}}^i, \tag{4.75}$$

where $N_{mc} = 50$ denotes the number of MC simulation runs, $\bar{\boldsymbol{s}}^i$ denotes the mean squared error calculated for i^{th} simulation run as $\bar{\boldsymbol{s}}^i = \frac{1}{N} \sum_{k=1}^{k=N} (\hat{\boldsymbol{x}}^{o,i}(t_k) - \boldsymbol{x}^{o,i}(t_k))^2$. The sample covariance corresponding to $\bar{\bar{\boldsymbol{s}}}$ is calculated as

$$\mathbf{\Sigma}_s = \frac{1}{N_{mc} - 1} \sum_{i=1}^{N_{mc}} (\bar{\boldsymbol{s}}^i - \bar{\bar{\boldsymbol{s}}})(\bar{\boldsymbol{s}}^i - \bar{\bar{\boldsymbol{s}}})^T. \tag{4.76}$$

Also note that $\bar{\bar{s}}_3$ denotes the third entry — corresponding to the error in the AO angle $q_3^{o,a}$ — from the vector $\bar{\bar{\boldsymbol{s}}}$. Further, $\sigma_{s,3}^2$ corresponds to the third entry from the diagonal of the matrix $\mathbf{\Sigma}_s$ and denotes the sample variance corresponding to $\bar{\bar{s}}_3$. The tables 4.2 and 4.3 show the performance statistics of the MC simulations. Table 4.2 contains simulation results when the estimators had perfect knowledge of parameters.

Table 4.3 contains simulation results corresponding to situation when the estimators had erroneous parameters of the robot. The errors in the parameters used by EKF and UKF are used to test the robustness of the estimation algorithms. Table 4.2 shows that when the model used for estimation is known perfectly adding gyroscope or accelerometer has a little effect on the quality of the AO estimate. However, when the model of the system is not perfectly known, the resulting estimators benefit from additional knowledge provided by the gyroscope or the accelerometer, as can be seen in the table 4.3. It is also observed that the UKF showed very stable behavior, always providing estimates of the AO. On the other hand the EKF failed to provide reasonable AO estimates and the closed-loop became unstable. This behavior was observed only when the parameters of the system used for the time update of the EKF were not known perfectly. It can be seen, however, that adding additional sensors was beneficial for the estimation performance. This shows that by using the hybrid model in the state estimators improves the estimates significantly. It also shows that the algorithms of EKF and UKF extended with the impact model can be used in the closed-loop control of the underactuated walking robots. Typical plots of the UKF estimator — estimating the vector $\boldsymbol{x}^o$ — are given on the fig. 4.4. The abrupt changes that can be seen in the plots are due to the impact and relabeling. It can be seen that the estimation performance is excellent even in the presence of impact. It should be noted here that without the incorporating the impact to the estimator, the estimator fails to provide useful estimates.

Table 4.2 – Estimation with perfect knowledge of model parameters.

EKF				
Error stat.	IRC	IRC & Gyro.	IRC & Accel.	IRC & IMU
$\bar{\bar{s}}_3$	5.31×10^{-4}	1.92×10^{-2}	1.43×10^{-3}	1.22×10^{-5}
$\sigma^2_{s,3}$	3.82×10^{-10}	2.99×10^{-3}	9.70×10^{-5}	2.75×10^{-12}
$\sum_{i=1}^{n_x} \bar{\bar{s}}_i$	2.82×10^{-2}	7.04×10^{-2}	5.77×10^{-1}	1.93×10^{-2}
$\mathrm{tr}(\boldsymbol{\Sigma}_s)$	1.74×10^{-7}	7.23×10^{-3}	7.40×10^{0}	1.14×10^{-8}
UKF				
Error stat.	IRC	IRC & Gyro.	IRC & Accel.	IRC & IMU
$\bar{\bar{s}}_3$	3.70×10^{-5}	6.85×10^{-5}	2.13×10^{-5}	2.42×10^{-5}
$\sigma^2_{s,3}$	6.34×10^{-11}	3.06×10^{-9}	4.24×10^{-11}	4.08×10^{-11}
$\sum_{i=1}^{n_x} \bar{\bar{s}}_i$	2.20×10^{-2}	1.73×10^{-2}	1.94×10^{-2}	1.72×10^{-2}
$\mathrm{tr}(\boldsymbol{\Sigma}_s)$	8.45×10^{-10}	7.50×10^{-9}	3.31×10^{-5}	1.29×10^{-9}

Table 4.3 – Estimation with 10% errors in model parameters.

EKF				
Error stat.	IRC	IRC & Gyro.	IRC & Accel.	IRC & IMU
$\bar{\bar{s}}_3$	NaN	NaN	1.95×10^{1}	1.29×10^{1}
$\sigma^2_{s,3}$	NaN	NaN	1.75×10^{2}	1.48×10^{2}
$\sum_{i=1}^{n_x} \bar{\bar{s}}_i$	NaN	NaN	1.38×10^{2}	7.31×10^{1}
$\mathrm{tr}(\boldsymbol{\Sigma}_s)$	NaN	NaN	5.38×10^{3}	1.01×10^{3}
UKF				
Error stat.	IRC	IRC & Gyro.	IRC & Accel.	IRC & IMU
$\bar{\bar{s}}_3$	3.72×10^{1}	1.67×10^{0}	5.67×10^{-1}	9.55×10^{-1}
$\sigma^2_{s,3}$	1.70×10^{3}	1.52×10^{1}	2.06×10^{0}	2.17×10^{1}
$\sum_{i=1}^{n_x} \bar{\bar{s}}_i$	3.33×10^{2}	3.86×10^{1}	5.25×10^{1}	4.02×10^{1}
$\mathrm{tr}(\boldsymbol{\Sigma}_s)$	1.52×10^{5}	4.24×10^{2}	7.47×10^{2}	2.78×10^{2}

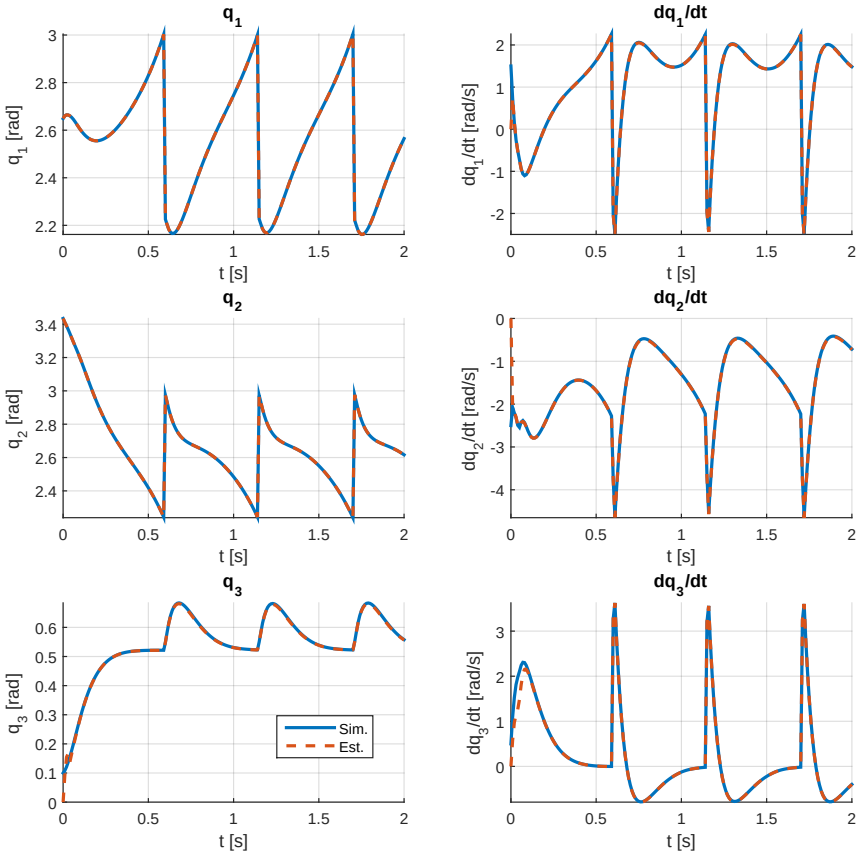


Figure 4.4 – Typical plots of the UKF estimator with perfect knowledge of parameters, using only IRC measurements of relative angles $q_1^{o,r}$ and $q_2^{o,r}$.

5 Conclusions

This work studies two problems, both closely related to the control of underactuated walking robots. First studied problem is the offline identification of underactuated walking robot. The second studied problem is the online estimation of the absolute orientation of the robot.

Offline experimental identification of underactuated walking robots

A walking robot is a highly nonlinear dynamical system. However, this model has a special property that it is linear in parameters — after proper reparametrization. This property would potentially allow to solve the parameter estimation problem very easily by transforming it to the problem of linear regression. However, there are two fundamental problems that restrict the application of the linear regression methods. The first problem is unavailability of the measurement of the absolute orientation angle in most bipedal robots. However, some works present a special measurement apparatus which allows such a measurement. Nevertheless, this apparatus restricts the practical usability of such a robot outside the laboratory [El Yaaqoubi and Abba (2009)], [Park et al. (2011)]. When the absolute orientation angle of the robot is unknown, the linear regression cannot be used, since the design matrix required in the algorithm can not be formed. This problem leads to using measurement instruments like gyroscopes and accelerometers or laser distance sensors. The usage of such sensors is, however, problematic as well since these instruments are likely to provide noisy measurements. The noisy measurements are again representing a problem. The usage of the design matrix containing errors in

regressor variables leads to biased estimates of the robot parameters.

Online state estimation for underactuated walking robots

The problem of the absolute angle estimation during the walking was studied and solved using nonlinear observers converging in finite time to avoid the problem of uncertainty of the impact. However this solution seems to be suboptimal. The reasons are following, the estimate is based on the robot model, which is likely to be only an approximation to a real robot. Because of this, the estimate is not perfect and will be affected by the impact. A wrong estimate of the state after the impact is easily achieved. This error can have a serious effect on the performance of the estimator.

5.1 Contributions of the author

Offline experimental identification of underactuated walking robots

The problems restricting the application of the methods of linear regression to the problem of identification of underactuated walking robots were solved in the chapter 3 of this work and the results were presented in [Dolinsky and Celikovsky (2017)]. The exploiting of the linear regression methods involve solution of the two problems. The first problem is how to include the measurements from the inertial sensors, e.g. gyroscopes and acceleroemters or the laser distance sensor in the regression matrix. Otherwise this matrix can not be formed. The required mathematical relationships were described in the chapter 3 for a complex model of 5-link planar robot. Based on this model a generalization to other models is straight forward. The second restriction in the application of the linear regression methods are the measurement errors of the measurement instruments, i.e. the gyroscopes, acceleroemters or the laser distance sensor. The measurement errors introduce errors in the regression matrix and thus results in the biased estimates of the parameters. The errors

enter the regression matrix in a nonlinear way and therefore the solution is not straight forward. These errors can be recursively minimized using the method of ML tailored for the walking robots. This method was formulated and verified using both simulations and laboratory experiments. The simulations shows that the ML method overperforms the benchmark algorithms classically used in identification of nonlinear systems. The laboratory experiments show that the method is very promising and results in very accurate models with excellent prediction capability. Further, the models can be subsequently used in control, since the key robot parameters required for the controller design can be estimated.

Online state estimation for underactuated walking robots

A solution to more robust estimation of the post-impact state is presented in the chapter 4 of this work together with the simulation study that shows that this approach results in increased performance. Two well known estimation algorithms — EKF and UKF — that are capable of incorporating the measurement noise statistics were extended to be applicable on the hybrid model of a walking robot. The extension in the EKF is computationally very inefficient and a heuristic alternative was proposed. The algorithm of UKF based on the so-called unscented transformation provides a better framework to handle the problem of the hybrid nature of the robot and the extension is much more computationally efficient than in the case of EKF. The estimation algorithms were tested using Monte Carlo simulations. The results show a considerable improvement in the estimators' performance.

5.2 Open problems

Among the interesting open problems in identification is the efficient estimation from closed-loop data containing noisy measurements of gyroscope, accelerometers or distance measuring sensors such as laser sensor or camera. The estimation from the closed-loop noisy data is a challenging problem due to correlation between the torque measurements and the

absolute orientation measurement errors. Another open problem is to estimate parameters of the robot during walking on the uneven terrain, in particular, walking on a slope. This task requires the estimation of both absolute orientation and the estimation of the slope. Finally, extending and validating the methods of the offline identification utilizing the linear relationship in parameters during 3D walking is a matter of ongoing research.

6 Appendix

6.1 Models of robots

6.1.1 5-link — AO referenced to the stance leg

In this subsection a model for the five link planar bipedal robot depicted on the left part of the figure 2.1. The AO of the robot is given with respect to the vertical and the stance leg. The absolute angles $\boldsymbol{\phi}$ defined with respect to horizontal and beginning of each link are given as

$$\phi_1 = q_1^a, \tag{6.1}$$
$$\phi_2 = q_1^a + q_2^r, \tag{6.2}$$
$$\phi_3 = q_1^a + q_2^r + q_3^r, \tag{6.3}$$
$$\phi_4 = q_1^a + q_2^r + q_3^r + q_4^r, \tag{6.4}$$
$$\phi_5 = q_1^a + q_2^r + q_3^r + q_4^r + q_5^r. \tag{6.5}$$

The entries of the matrices $\mathbf{D}$, $\mathbf{C}$ and vector $\boldsymbol{G}$ for the 5DoF 5-link robot can be obtained using symbolic software. Matrix $\mathbf{D}$ is given for the extended — unpinned — model to allow for the calculation of the impact map.

$$\begin{aligned} D_{1,1} = {} & I_1 + I_2 + I_3 + I_4 + I_5 + (2l_2 l_{c_5} m_5)\cos(q_3^r + q_4^r + q_5^r) + (2l_1 l_4 m_5 + \\ & 2l_1 l_{c_4} m_4)\cos(q_2^r + q_3^r + q_4^r) + (2l_1 l_2 m_3 + 2l_1 l_2 m_4 + 2l_1 l_2 m_5 + \\ & 2l_1 l_{c_2} m_2)\cos(q_2^r) + (2l_1 l_3 m_4 + 2l_1 l_3 m_5 + 2l_1 l_{c_3} m_3)\cos(q_2^r + q_3^r) + \\ & (2l_2 l_3 m_4 + 2l_2 l_3 m_5 + 2l_2 l_{c_3} m_3)\cos(q_3^r) + \\ & (2l_1 l_{c_5} m_5)\cos(q_2^r + q_3^r + q_4^r + q_5^r) + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4 + l_2^2 m_3 + \end{aligned}$$

$$l_1^2 m_5 + l_2^2 m_4 + l_2^2 m_5 + l_3^2 m_4 + l_3^2 m_5 + l_4^2 m_5 + l_{c_1}^2 m_1 + l_{c_2}^2 m_2 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + (2 l_3 l_{c_5} m_5) \cos(q_4^r + q_5^r) + (2 l_2 l_4 m_5 + 2 l_2 l_{c_4} m_4) \cos(q_3^r + q_4^r) + (2 l_4 l_{c_5} m_5) \cos(q_5^r) + (2 l_3 l_4 m_5 + 2 l_3 l_{c_4} m_4) \cos(q_4^r), \tag{6.6}$$

$$D_{1,2} = I_2 + I_3 + I_4 + I_5 + (2 l_2 l_{c_5} m_5) \cos(q_3^r + q_4^r + q_5^r) + (l_1 l_4 m_5 + l_1 l_{c_4} m_4) \cos(q_2^r + q_3^r + q_4^r) + (l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 l_2 m_5 + l_1 l_{c_2} m_2) \cos(q_2^r) + (l_1 l_3 m_4 + l_1 l_3 m_5 + l_1 l_{c_3} m_3) \cos(q_2^r + q_3^r) + (2 l_2 l_3 m_4 + 2 l_2 l_3 m_5 + 2 l_2 l_{c_3} m_3) \cos(q_3^r) + (l_1 l_{c_5} m_5) \cos(q_2^r + q_3^r + q_4^r + q_5^r) + l_2^2 m_3 + l_2^2 m_4 + l_2^2 m_5 + l_3^2 m_4 + l_3^2 m_5 + l_4^2 m_5 + l_{c_2}^2 m_2 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + (2 l_3 l_{c_5} m_5) \cos(q_4^r + q_5^r) + (2 l_2 l_4 m_5 + 2 l_2 l_{c_4} m_4) \cos(q_3^r + q_4^r) + (2 l_4 l_{c_5} m_5) \cos(q_5^r) + (2 l_3 l_4 m_5 + 2 l_3 l_{c_4} m_4) \cos(q_4^r), \tag{6.7}$$

$$D_{1,3} = I_3 + I_4 + I_5 + (l_2 l_{c_5} m_5) \cos(q_3^r + q_4^r + q_5^r) + (l_1 l_4 m_5 + l_1 l_{c_4} m_4) \cos(q_2^r + q_3^r + q_4^r) + (l_1 l_3 m_4 + l_1 l_3 m_5 + l_1 l_{c_3} m_3) \cos(q_2^r + q_3^r) + (l_2 l_3 m_4 + l_2 l_3 m_5 + l_2 l_{c_3} m_3) \cos(q_3^r) + (l_1 l_{c_5} m_5) \cos(q_2^r + q_3^r + q_4^r + q_5^r) + l_3^2 m_4 + l_3^2 m_5 + l_4^2 m_5 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + (2 l_3 l_{c_5} m_5) \cos(q_4^r + q_5^r) + (l_2 l_4 m_5 + l_2 l_{c_4} m_4) \cos(q_3^r + q_4^r) + (2 l_4 l_{c_5} m_5) \cos(q_5^r) + (2 l_3 l_4 m_5 + 2 l_3 l_{c_4} m_4) \cos(q_4^r), \tag{6.8}$$

$$D_{1,4} = I_4 + I_5 + (l_2 l_{c_5} m_5) \cos(q_3^r + q_4^r + q_5^r) + (l_1 l_4 m_5 + l_1 l_{c_4} m_4) \cos(q_2^r + q_3^r + q_4^r) + (l_1 l_{c_5} m_5) \cos(q_2^r + q_3^r + q_4^r + q_5^r) + l_4^2 m_5 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + (l_3 l_{c_5} m_5) \cos(q_4^r + q_5^r) + (l_2 l_4 m_5 + l_2 l_{c_4} m_4) \cos(q_3^r + q_4^r) + (2 l_4 l_{c_5} m_5) \cos(q_5^r) + (l_3 l_4 m_5 + l_3 l_{c_4} m_4) \cos(q_4^r), \tag{6.9}$$

$$D_{1,5} = I_5 + (l_2 l_{c_5} m_5) \cos(q_3^r + q_4^r + q_5^r) + (l_1 l_{c_5} m_5) \cos(q_2^r + q_3^r + q_4^r + q_5^r) + l_{c_5}^2 m_5 + (l_3 l_{c_5} m_5) \cos(q_4^r + q_5^r) +$$

$$(l_4 l_{c_5} m_5)\cos(q_5^r), \tag{6.10}$$

$$\begin{aligned} D_{1,6} = {} & (l_2 m_3 + l_2 m_4 + l_2 m_5 + l_{c_2} m_2)\cos(q_1^a + q_2^r) + (l_3 m_4 + l_3 m_5 + \\ & l_{c_3} m_3)\cos(q_1^a + q_2^r + q_3^r) + (l_{c_5} m_5)\cos(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + \\ & (l_1 m_2 + l_1 m_3 + l_1 m_4 + l_1 m_5 + l_{c_1} m_1)\cos(q_1^a) + (l_4 m_5 + \\ & l_{c_4} m_4)\cos(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.11}$$

$$\begin{aligned} D_{1,7} = {} & (-l_2 m_3 - l_2 m_4 - l_2 m_5 - l_{c_2} m_2)\sin(q_1^a + q_2^r) + (-l_3 m_4 - l_3 m_5 - \\ & l_{c_3} m_3)\sin(q_1^a + q_2^r + q_3^r) + (-l_{c_5} m_5)\sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + \\ & (-l_1 m_2 - l_1 m_3 - l_1 m_4 - l_1 m_5 - l_{c_1} m_1)\sin(q_1^a) + (-l_4 m_5 - \\ & l_{c_4} m_4)\sin(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.12}$$

$$\begin{aligned} D_{2,2} = {} & I_2 + I_3 + I_4 + I_5 + (2 l_2 l_{c_5} m_5)\cos(q_3^r + q_4^r + q_5^r) + (2 l_2 l_3 m_4 + \\ & 2 l_2 l_3 m_5 + 2 l_2 l_{c_3} m_3)\cos(q_3^r) + l_2^2 m_3 + l_2^2 m_4 + l_2^2 m_5 + l_3^2 m_4 + l_3^2 m_5 + \\ & l_4^2 m_5 + l_{c_2}^2 m_2 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + (2 l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + \\ & (2 l_2 l_4 m_5 + 2 l_2 l_{c_4} m_4)\cos(q_3^r + q_4^r) + (2 l_4 l_{c_5} m_5)\cos(q_5^r) + (2 l_3 l_4 m_5 + \\ & 2 l_3 l_{c_4} m_4)\cos(q_4^r), \end{aligned} \tag{6.13}$$

$$\begin{aligned} D_{2,3} = {} & I_3 + I_4 + I_5 + (l_2 l_{c_5} m_5)\cos(q_3^r + q_4^r + q_5^r) + (l_2 l_3 m_4 + l_2 l_3 m_5 + \\ & l_2 l_{c_3} m_3)\cos(q_3^r) + l_3^2 m_4 + l_3^2 m_5 + l_4^2 m_5 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + \\ & (2 l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + (l_2 l_4 m_5 + l_2 l_{c_4} m_4)\cos(q_3^r + q_4^r) + \\ & (2 l_4 l_{c_5} m_5)\cos(q_5^r) + (2 l_3 l_4 m_5 + 2 l_3 l_{c_4} m_4)\cos(q_4^r), \end{aligned} \tag{6.14}$$

$$\begin{aligned} D_{2,4} = {} & I_4 + I_5 + (l_2 l_{c_5} m_5)\cos(q_3^r + q_4^r + q_5^r) + l_4^2 m_5 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + \\ & (l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + (l_2 l_4 m_5 + l_2 l_{c_4} m_4)\cos(q_3^r + q_4^r) + \\ & (2 l_4 l_{c_5} m_5)\cos(q_5^r) + (l_3 l_4 m_5 + l_3 l_{c_4} m_4)\cos(q_4^r), \end{aligned} \tag{6.15}$$

$$\begin{aligned} D_{2,5} = {} & I_5 + (l_2 l_{c_5} m_5)\cos(q_3^r + q_4^r + q_5^r) + l_{c_5}^2 m_5 + (l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + \\ & (l_4 l_{c_5} m_5)\cos(q_5^r), \end{aligned} \tag{6.16}$$

$$\begin{aligned} D_{2,6} = {} & (l_2 m_3 + l_2 m_4 + l_2 m_5 + l_{c_2} m_2)\cos(q_1^a + q_2^r) + (l_3 m_4 + l_3 m_5 + \\ & l_{c_3} m_3)\cos(q_1^a + q_2^r + q_3^r) + (l_{c_5} m_5)\cos(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + \end{aligned}$$

$$(l_4 m_5 + l_{c_4} m_4)\cos(q_1^a + q_2^r + q_3^r + q_4^r), \tag{6.17}$$

$$\begin{aligned} D_{2,7} = {} & (-l_2 m_3 - l_2 m_4 - l_2 m_5 - l_{c_2} m_2)\sin(q_1^a + q_2^r) + (-l_3 m_4 - l_3 m_5 - \\ & l_{c_3} m_3)\sin(q_1^a + q_2^r + q_3^r) + (-l_{c_5} m_5)\sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + \\ & (-l_4 m_5 - l_{c_4} m_4)\sin(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.18}$$

$$\begin{aligned} D_{3,3} = {} & I_3 + I_4 + I_5 + l_3^2 m_4 + l_3^2 m_5 + l_4^2 m_5 + l_{c_3}^2 m_3 + l_{c_4}^2 m_4 + l_{c_5}^2 m_5 + \\ & (2 l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + (2 l_4 l_{c_5} m_5)\cos(q_5^r) + (2 l_3 l_4 m_5 + \\ & 2 l_3 l_{c_4} m_4)\cos(q_4^r), \end{aligned} \tag{6.19}$$

$$\begin{aligned} D_{3,4} = {} & I_4 + I_5 + m_5 l_4^2 + m_4 l_{c_4}^2 + m_5 l_{c_5}^2 + (l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + \\ & (2 l_4 l_{c_5} m_5)\cos(q_5^r) + (l_3 l_4 m_5 + l_3 l_{c_4} m_4)\cos(q_4^r), \end{aligned} \tag{6.20}$$

$$D_{3,5} = I_5 + m_5 l_{c_5}^2 + (l_3 l_{c_5} m_5)\cos(q_4^r + q_5^r) + (l_4 l_{c_5} m_5)\cos(q_5^r), \tag{6.21}$$

$$\begin{aligned} D_{3,6} = {} & (l_3 m_4 + l_3 m_5 + l_{c_3} m_3)\cos(q_1^a + q_2^r + q_3^r) + \\ & (l_{c_5} m_5)\cos(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (l_4 m_5 + \\ & l_{c_4} m_4)\cos(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.22}$$

$$\begin{aligned} D_{3,7} = {} & (-l_3 m_4 - l_3 m_5 - l_{c_3} m_3)\sin(q_1^a + q_2^r + q_3^r) + \\ & (-l_{c_5} m_5)\sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-l_4 m_5 - \\ & l_{c_4} m_4)\sin(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.23}$$

$$D_{4,4} = I_4 + I_5 + m_5 l_4^2 + m_4 l_{c_4}^2 + m_5 l_{c_5}^2 + (2 l_4 l_{c_5} m_5)\cos(q_5^r), \tag{6.24}$$

$$D_{4,5} = I_5 + m_5 l_{c_5}^2 + (l_4 l_{c_5} m_5)\cos(q_5^r), \tag{6.25}$$

$$\begin{aligned} D_{4,6} = {} & (l_{c_5} m_5)\cos(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (l_4 m_5 + \\ & l_{c_4} m_4)\cos(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.26}$$

$$\begin{aligned} D_{4,7} = {} & (-l_{c_5} m_5)\sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-l_4 m_5 - \\ & l_{c_4} m_4)\sin(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.27}$$

$$D_{5,5} = I_5 + m_5 l_{c_5}^2, \tag{6.28}$$

$$D_{5,6} = l_{c_5} m_5 \cos(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r), \tag{6.29}$$

$$D_{5,7} = -l_{c_5} m_5 \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r), \tag{6.30}$$

$$D_{6,6} = D_{7,7} = m_1 + m_2 + m_3 + m_4 + m_5, \quad D_{6,7} = 0. \tag{6.31}$$

The remaining entries of the matrix $\mathbf{D}$ are completed by symmetry. Entries for the matrix $\mathbf{C}$ are

$$\begin{aligned} C_{1,1} = {} & (-\dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + \\ & (-\dot{q}_2^r l_1 l_4 m_5 - \dot{q}_3^r l_1 l_4 m_5 - \dot{q}_4^r l_1 l_4 m_5 - \dot{q}_2^r l_1 l_{c_4} m_4 - \dot{q}_3^r l_1 l_{c_4} m_4 - \\ & \dot{q}_4^r l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_2^r l_1 l_{c_5} m_5 - \\ & \dot{q}_3^r l_1 l_{c_5} m_5 - \dot{q}_4^r l_1 l_{c_5} m_5 - \dot{q}_5^r l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + \\ & (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \\ & \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_2^r l_1 l_3 m_4 - \dot{q}_2^r l_1 l_3 m_5 - \\ & \dot{q}_3^r l_1 l_3 m_4 - \dot{q}_3^r l_1 l_3 m_5 - \dot{q}_2^r l_1 l_{c_3} m_3 - \dot{q}_3^r l_1 l_{c_3} m_3) \sin(q_2^r + q_3^r) + \\ & (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r) + (-\dot{q}_3^r l_2 l_3 m_4 - \dot{q}_3^r l_2 l_3 m_5 - \\ & \dot{q}_3^r l_2 l_{c_3} m_3) \sin(q_3^r) + (-\dot{q}_2^r l_1 l_2 m_3 - \dot{q}_2^r l_1 l_2 m_4 - \dot{q}_2^r l_1 l_2 m_5 - \\ & \dot{q}_2^r l_1 l_{c_2} m_2) \sin(q_2^r), \end{aligned} \tag{6.32}$$

$$\begin{aligned} C_{1,2} = {} & (-\dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + \\ & (-\dot{q}_1^a l_1 l_4 m_5 - \dot{q}_2^r l_1 l_4 m_5 - \dot{q}_3^r l_1 l_4 m_5 - \dot{q}_4^r l_1 l_4 m_5 - \dot{q}_1^a l_1 l_{c_4} m_4 - \\ & \dot{q}_2^r l_1 l_{c_4} m_4 - \dot{q}_3^r l_1 l_{c_4} m_4 - \dot{q}_4^r l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + \\ & (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_1^a l_1 l_{c_5} m_5 - \dot{q}_2^r l_1 l_{c_5} m_5 - \dot{q}_3^r l_1 l_{c_5} m_5 - \\ & \dot{q}_4^r l_1 l_{c_5} m_5 - \dot{q}_5^r l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \\ & \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_{c_4} m_4 - \\ & \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r) + \\ & (-\dot{q}_3^r l_2 l_3 m_4 - \dot{q}_3^r l_2 l_3 m_5 - \dot{q}_3^r l_2 l_{c_3} m_3) \sin(q_3^r) + (-\dot{q}_1^a l_1 l_3 m_4 - \\ & \dot{q}_1^a l_1 l_3 m_5 - \dot{q}_2^r l_1 l_3 m_4 - \dot{q}_2^r l_1 l_3 m_5 - \dot{q}_3^r l_1 l_3 m_4 - \dot{q}_3^r l_1 l_3 m_5 - \dot{q}_1^a l_1 l_{c_3} m_3 - \\ & \dot{q}_2^r l_1 l_{c_3} m_3 - \dot{q}_3^r l_1 l_{c_3} m_3) \sin(q_2^r + q_3^r) + (-\dot{q}_1^a l_1 l_2 m_3 - \dot{q}_1^a l_1 l_2 m_4 - \end{aligned}$$

$$\dot{q}_2^r l_1 l_2 m_3 - \dot{q}_1^a l_1 l_2 m_5 - \dot{q}_2^r l_1 l_2 m_4 - \dot{q}_2^r l_1 l_2 m_5 - \dot{q}_1^a l_1 l_{c_2} m_2 - \dot{q}_2^r l_1 l_{c_2} m_2) \sin(q_2^r), \tag{6.33}$$

$$\begin{aligned} C_{1,3} = {} & (-\dot{q}_1^a l_2 l_{c_5} m_5 - \dot{q}_2^r l_2 l_{c_5} m_5 - \dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \\ & \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_1^a l_1 l_4 m_5 - \dot{q}_2^r l_1 l_4 m_5 - \dot{q}_3^r l_1 l_4 m_5 - \\ & \dot{q}_4^r l_1 l_4 m_5 - \dot{q}_1^a l_1 l_{c_4} m_4 - \dot{q}_2^r l_1 l_{c_4} m_4 - \dot{q}_3^r l_1 l_{c_4} m_4 - \\ & \dot{q}_4^r l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_1^a l_1 l_{c_5} m_5 - \\ & \dot{q}_2^r l_1 l_{c_5} m_5 - \dot{q}_3^r l_1 l_{c_5} m_5 - \dot{q}_4^r l_1 l_{c_5} m_5 - \\ & \dot{q}_5^r l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \\ & \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r) + \\ & (-\dot{q}_1^a l_2 l_4 m_5 - \dot{q}_2^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_1^a l_2 l_{c_4} m_4 - \\ & \dot{q}_2^r l_2 l_{c_4} m_4 - \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_1^a l_1 l_3 m_4 - \\ & \dot{q}_1^a l_1 l_3 m_5 - \dot{q}_2^r l_1 l_3 m_4 - \dot{q}_2^r l_1 l_3 m_5 - \dot{q}_3^r l_1 l_3 m_4 - \dot{q}_3^r l_1 l_3 m_5 - \dot{q}_1^a l_1 l_{c_3} m_3 - \\ & \dot{q}_2^r l_1 l_{c_3} m_3 - \dot{q}_3^r l_1 l_{c_3} m_3) \sin(q_2^r + q_3^r) + (-\dot{q}_1^a l_2 l_3 m_4 - \dot{q}_1^a l_2 l_3 m_5 - \\ & \dot{q}_2^r l_2 l_3 m_4 - \dot{q}_2^r l_2 l_3 m_5 - \dot{q}_3^r l_2 l_3 m_4 - \dot{q}_3^r l_2 l_3 m_5 - \dot{q}_1^a l_2 l_{c_3} m_3 - \dot{q}_2^r l_2 l_{c_3} m_3 - \\ & \dot{q}_3^r l_2 l_{c_3} m_3) \sin(q_3^r), \end{aligned} \tag{6.34}$$

$$\begin{aligned} C_{1,4} = {} & (-\dot{q}_1^a l_2 l_{c_5} m_5 - \dot{q}_2^r l_2 l_{c_5} m_5 - \dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \\ & \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_1^a l_1 l_4 m_5 - \dot{q}_2^r l_1 l_4 m_5 - \dot{q}_3^r l_1 l_4 m_5 - \\ & \dot{q}_4^r l_1 l_4 m_5 - \dot{q}_1^a l_1 l_{c_4} m_4 - \dot{q}_2^r l_1 l_{c_4} m_4 - \dot{q}_3^r l_1 l_{c_4} m_4 - \\ & \dot{q}_4^r l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_1^a l_1 l_{c_5} m_5 - \\ & \dot{q}_2^r l_1 l_{c_5} m_5 - \dot{q}_3^r l_1 l_{c_5} m_5 - \dot{q}_4^r l_1 l_{c_5} m_5 - \\ & \dot{q}_5^r l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + (-\dot{q}_1^a l_3 l_{c_5} m_5 - \dot{q}_2^r l_3 l_{c_5} m_5 - \\ & \dot{q}_3^r l_3 l_{c_5} m_5 - \dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_1^a l_2 l_4 m_5 - \\ & \dot{q}_2^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_1^a l_2 l_{c_4} m_4 - \dot{q}_2^r l_2 l_{c_4} m_4 - \\ & \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_1^a l_3 l_4 m_5 - \dot{q}_2^r l_3 l_4 m_5 - \\ & \dot{q}_3^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_4 m_5 - \dot{q}_1^a l_3 l_{c_4} m_4 - \dot{q}_2^r l_3 l_{c_4} m_4 - \dot{q}_3^r l_3 l_{c_4} m_4 - \\ & \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r), \end{aligned} \tag{6.35}$$

$$C_{1,5} = (-l_3 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_4^r + q_5^r) + (-l_4 l_{c_5} m_5 (\dot{q}_1^a +$$

$$\dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_5^r) + (-l_2 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_3^r + q_4^r + q_5^r) + (-l_1 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_2^r + q_3^r + q_4^r + q_5^r), \tag{6.36}$$

$$C_{2,1} = (\dot{q}_1^a l_1 l_3 m_4 + \dot{q}_1^a l_1 l_3 m_5 + \dot{q}_1^a l_1 l_{c_3} m_3) \sin(q_2^r + q_3^r) + (\dot{q}_1^a l_1 l_2 m_3 + \dot{q}_1^a l_1 l_2 m_4 + \dot{q}_1^a l_1 l_2 m_5 + \dot{q}_1^a l_1 l_{c_2} m_2) \sin(q_2^r) + (\dot{q}_1^a l_1 l_4 m_5 + \dot{q}_1^a l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (\dot{q}_1^a l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + (-\dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r) + (-\dot{q}_3^r l_2 l_3 m_4 - \dot{q}_3^r l_2 l_3 m_5 - \dot{q}_3^r l_2 l_{c_3} m_3) \sin(q_3^r), \tag{6.37}$$

$$C_{2,2} = (-\dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_3^r l_2 (l_3 m_4 + l_3 m_5 + l_{c_3} m_3)) \sin(q_3^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r (l_3 l_4 m_5 + l_3 l_{c_4} m_4)) \sin(q_4^r) + (-\dot{q}_4^r (l_2 l_4 m_5 + l_2 l_{c_4} m_4) - \dot{q}_3^r l_2 (l_4 m_5 + l_{c_4} m_4)) \sin(q_3^r + q_4^r), \tag{6.38}$$

$$C_{2,3} = (-\dot{q}_1^a l_2 l_{c_5} m_5 - \dot{q}_2^r l_2 l_{c_5} m_5 - \dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r) + (-\dot{q}_1^a l_2 l_4 m_5 - \dot{q}_2^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_1^a l_2 l_{c_4} m_4 - \dot{q}_2^r l_2 l_{c_4} m_4 - \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_1^a l_2 l_3 m_4 - \dot{q}_1^a l_2 l_3 m_5 - \dot{q}_2^r l_2 l_3 m_4 - \dot{q}_2^r l_2 l_3 m_5 - \dot{q}_3^r l_2 l_3 m_4 - \dot{q}_3^r l_2 l_3 m_5 - \dot{q}_1^a l_2 l_{c_3} m_3 - \dot{q}_2^r l_2 l_{c_3} m_3 - \dot{q}_3^r l_2 l_{c_3} m_3) \sin(q_3^r), \tag{6.39}$$

$$C_{2,4} = (-\dot{q}_1^a l_2 l_{c_5} m_5 - \dot{q}_2^r l_2 l_{c_5} m_5 - \dot{q}_3^r l_2 l_{c_5} m_5 - \dot{q}_4^r l_2 l_{c_5} m_5 - \dot{q}_5^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_1^a l_3 l_{c_5} m_5 - \dot{q}_2^r l_3 l_{c_5} m_5 - \dot{q}_3^r l_3 l_{c_5} m_5 - \dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_1^a l_2 l_4 m_5 - \dot{q}_2^r l_2 l_4 m_5 - \dot{q}_3^r l_2 l_4 m_5 - \dot{q}_4^r l_2 l_4 m_5 - \dot{q}_1^a l_2 l_{c_4} m_4 -$$

$$\dot{q}_2^r l_2 l_{c_4} m_4 - \dot{q}_3^r l_2 l_{c_4} m_4 - \dot{q}_4^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (-\dot{q}_1^a l_3 l_4 m_5 - \dot{q}_2^r l_3 l_4 m_5 - \dot{q}_3^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_4 m_5 - \dot{q}_1^a l_3 l_{c_4} m_4 - \dot{q}_2^r l_3 l_{c_4} m_4 - \dot{q}_3^r l_3 l_{c_4} m_4 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r), \tag{6.40}$$

$$C_{2,5} = (-l_3 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_4^r + q_5^r) + (-l_4 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_5^r) + (-l_2 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_3^r + q_4^r + q_5^r), \tag{6.41}$$

$$C_{3,1} = (\dot{q}_1^a l_1 l_3 m_4 + \dot{q}_1^a l_1 l_3 m_5 + \dot{q}_1^a l_1 l_{c_3} m_3) \sin(q_2^r + q_3^r) + (\dot{q}_1^a l_2 l_4 m_5 + \dot{q}_2^r l_2 l_4 m_5 + \dot{q}_1^a l_2 l_{c_4} m_4 + \dot{q}_2^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (\dot{q}_1^a l_2 l_3 m_4 + \dot{q}_1^a l_2 l_3 m_5 + \dot{q}_2^r l_2 l_3 m_4 + \dot{q}_2^r l_2 l_3 m_5 + \dot{q}_1^a l_2 l_{c_3} m_3 + \dot{q}_2^r l_2 l_{c_3} m_3) \sin(q_3^r) + (\dot{q}_1^a l_1 l_4 m_5 + \dot{q}_1^a l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (\dot{q}_1^a l_2 l_{c_5} m_5 + \dot{q}_2^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (\dot{q}_1^a l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r l_3 l_4 m_5 - \dot{q}_4^r l_3 l_{c_4} m_4) \sin(q_4^r), \tag{6.42}$$

$$C_{3,2} = (\dot{q}_1^a (l_2 l_4 m_5 + l_2 l_{c_4} m_4) + \dot{q}_2^r (l_2 l_4 m_5 + l_2 l_{c_4} m_4)) \sin(q_3^r + q_4^r) + (\dot{q}_1^a (l_2 l_3 m_4 + l_2 l_3 m_5 + l_2 l_{c_3} m_3) + \dot{q}_2^r (l_2 l_3 m_4 + l_2 l_3 m_5 + l_2 l_{c_3} m_3)) \sin(q_3^r) + (\dot{q}_1^a l_2 l_{c_5} m_5 + \dot{q}_2^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r (l_3 l_4 m_5 + l_3 l_{c_4} m_4)) \sin(q_4^r), \tag{6.43}$$

$$C_{3,3} = (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_4^r (l_3 l_4 m_5 + l_3 l_{c_4} m_4)) \sin(q_4^r), \tag{6.44}$$

$$C_{3,4} = (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r) + (-\dot{q}_1^a l_3 l_{c_5} m_5 - \dot{q}_2^r l_3 l_{c_5} m_5 - \dot{q}_3^r l_3 l_{c_5} m_5 - \dot{q}_4^r l_3 l_{c_5} m_5 - \dot{q}_5^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (-\dot{q}_1^a l_3 (l_4 m_5 + l_{c_4} m_4) - \dot{q}_2^r l_3 (l_4 m_5 + l_{c_4} m_4) - \dot{q}_3^r l_3 (l_4 m_5 + l_{c_4} m_4) - \dot{q}_4^r l_3 (l_4 m_5 + l_{c_4} m_4)) \sin(q_4^r), \tag{6.45}$$

$$C_{3,5} = (-l_3 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_4^r + q_5^r) + (-l_4 l_{c_5} m_5 (\dot{q}_1^a +$$

$$\dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_5^r), \tag{6.46}$$

$$\begin{aligned} C_{4,1} = {} & (\dot{q}_1^a l_3 l_{c_5} m_5 + \dot{q}_2^r l_3 l_{c_5} m_5 + \dot{q}_3^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (\dot{q}_1^a l_2 l_4 m_5 + \\ & \dot{q}_2^r l_2 l_4 m_5 + \dot{q}_1^a l_2 l_{c_4} m_4 + \dot{q}_2^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (\dot{q}_1^a l_3 l_4 m_5 + \\ & \dot{q}_2^r l_3 l_4 m_5 + \dot{q}_3^r l_3 l_4 m_5 + \dot{q}_1^a l_3 l_{c_4} m_4 + \dot{q}_2^r l_3 l_{c_4} m_4 + \dot{q}_3^r l_3 l_{c_4} m_4) \sin(q_4^r) + \\ & (\dot{q}_1^a l_1 l_4 m_5 + \dot{q}_1^a l_1 l_{c_4} m_4) \sin(q_2^r + q_3^r + q_4^r) + (\dot{q}_1^a l_2 l_{c_5} m_5 + \\ & \dot{q}_2^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (\dot{q}_1^a l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r) + \\ & (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r), \end{aligned} \tag{6.47}$$

$$\begin{aligned} C_{4,2} = {} & (\dot{q}_1^a l_3 l_{c_5} m_5 + \dot{q}_2^r l_3 l_{c_5} m_5 + \dot{q}_3^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (\dot{q}_1^a l_2 l_4 m_5 + \\ & \dot{q}_2^r l_2 l_4 m_5 + \dot{q}_1^a l_2 l_{c_4} m_4 + \dot{q}_2^r l_2 l_{c_4} m_4) \sin(q_3^r + q_4^r) + (\dot{q}_1^a l_3 l_4 m_5 + \\ & \dot{q}_2^r l_3 l_4 m_5 + \dot{q}_3^r l_3 l_4 m_5 + \dot{q}_1^a l_3 l_{c_4} m_4 + \dot{q}_2^r l_3 l_{c_4} m_4 + \dot{q}_3^r l_3 l_{c_4} m_4) \sin(q_4^r) + \\ & (\dot{q}_1^a l_2 l_{c_5} m_5 + \dot{q}_2^r l_2 l_{c_5} m_5) \sin(q_3^r + q_4^r + q_5^r) + (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r), \end{aligned} \tag{6.48}$$

$$\begin{aligned} C_{4,3} = {} & (\dot{q}_1^a l_3 l_{c_5} m_5 + \dot{q}_2^r l_3 l_{c_5} m_5 + \dot{q}_3^r l_3 l_{c_5} m_5) \sin(q_4^r + q_5^r) + (\dot{q}_1^a l_3 (l_4 m_5 + \\ & l_{c_4} m_4) + \dot{q}_2^r l_3 (l_4 m_5 + l_{c_4} m_4) + \dot{q}_3^r l_3 (l_4 m_5 + l_{c_4} m_4)) \sin(q_4^r) + \\ & (-\dot{q}_5^r l_4 l_{c_5} m_5) \sin(q_5^r), \end{aligned} \tag{6.49}$$

$$C_{4,4} = -\dot{q}_5^r l_4 l_{c_5} m_5 \sin(q_5^r), \tag{6.50}$$

$$C_{4,5} = (-l_4 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r + \dot{q}_5^r)) \sin(q_5^r), \tag{6.51}$$

$$\begin{aligned} C_{5,1} = {} & (l_{c_5} m_5 (\dot{q}_1^a l_3 + \dot{q}_2^r l_3 + \dot{q}_3^r l_3)) \sin(q_4^r + q_5^r) + (l_{c_5} m_5 (\dot{q}_1^a l_2 + \\ & \dot{q}_2^r l_2)) \sin(q_3^r + q_4^r + q_5^r) + (l_{c_5} m_5 (\dot{q}_1^a l_4 + \dot{q}_2^r l_4 + \dot{q}_3^r l_4 + \dot{q}_4^r l_4)) \sin(q_5^r) + \\ & (\dot{q}_1^a l_1 l_{c_5} m_5) \sin(q_2^r + q_3^r + q_4^r + q_5^r), \end{aligned} \tag{6.52}$$

$$\begin{aligned} C_{5,2} = {} & (l_{c_5} m_5 (\dot{q}_1^a l_3 + \dot{q}_2^r l_3 + \dot{q}_3^r l_3)) \sin(q_4^r + q_5^r) + (l_{c_5} m_5 (\dot{q}_1^a l_2 + \\ & \dot{q}_2^r l_2)) \sin(q_3^r + q_4^r + q_5^r) + (l_{c_5} m_5 (\dot{q}_1^a l_4 + \dot{q}_2^r l_4 + \dot{q}_3^r l_4 + \dot{q}_4^r l_4)) \sin(q_5^r), \end{aligned} \tag{6.53}$$

$$C_{5,3} = (l_{c_5} m_5 (\dot{q}_1^a l_3 + \dot{q}_2^r l_3 + \dot{q}_3^r l_3)) \sin(q_4^r + q_5^r) + (l_{c_5} m_5 (\dot{q}_1^a l_4 + \dot{q}_2^r l_4 +$$

$$\dot{q}_3^r l_4 + \dot{q}_4^r l_4)) \sin(q_5^r), \tag{6.54}$$

$$C_{5,4} = (l_4 l_{c_5} m_5 (\dot{q}_1^a + \dot{q}_2^r + \dot{q}_3^r + \dot{q}_4^r)) \sin(q_5^r), \tag{6.55}$$

$$C_{5,5} = 0. \tag{6.56}$$

Vector $\boldsymbol{G}$ describing the gravity effects is given as follows

$$\begin{aligned} G_{1,1} = {} & (-g l_{c_5} m_5) \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-g l_4 m_5 - \\ & g l_{c_4} m_4) \sin(q_1^a + q_2^r + q_3^r + q_4^r) + (-g l_1 m_2 - g l_1 m_3 - g l_1 m_4 - \\ & g l_1 m_5 - g l_{c_1} m_1) \sin(q_1^a) + (-g l_3 m_4 - g l_3 m_5 - \\ & g l_{c_3} m_3) \sin(q_1^a + q_2^r + q_3^r) + (-g l_2 m_3 - g l_2 m_4 - g l_2 m_5 - \\ & g l_{c_2} m_2) \sin(q_1^a + q_2^r), \end{aligned} \tag{6.57}$$

$$\begin{aligned} G_{2,1} = {} & (-g l_{c_5} m_5) \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-g l_4 m_5 - \\ & g l_{c_4} m_4) \sin(q_1^a + q_2^r + q_3^r + q_4^r) + (-g l_3 m_4 - g l_3 m_5 - \\ & g l_{c_3} m_3) \sin(q_1^a + q_2^r + q_3^r) + (-g l_2 m_3 - g l_2 m_4 - g l_2 m_5 - \\ & g l_{c_2} m_2) \sin(q_1^a + q_2^r), \end{aligned} \tag{6.58}$$

$$\begin{aligned} G_{3,1} = {} & (-g l_{c_5} m_5) \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-g l_4 m_5 - \\ & g l_{c_4} m_4) \sin(q_1^a + q_2^r + q_3^r + q_4^r) + (-g l_3 m_4 - g l_3 m_5 - \\ & g l_{c_3} m_3) \sin(q_1^a + q_2^r + q_3^r), \end{aligned} \tag{6.59}$$

$$\begin{aligned} G_{4,1} = {} & (-g l_{c_5} m_5) \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r) + (-g l_4 m_5 - \\ & g l_{c_4} m_4) \sin(q_1^a + q_2^r + q_3^r + q_4^r), \end{aligned} \tag{6.60}$$

$$G_{5,1} = -g l_{c_5} m_5 \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r). \tag{6.61}$$

The position of the end point of the swing leg is

$$\begin{aligned} \Upsilon_1 = {} & z_1 + l_1 \sin(q_1^a) + l_2 \sin(q_1^a + q_2^r) + \\ & + l_4 \sin(q_1^a + q_2^r + q_3^r + q_4^r) + l_5 \sin(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r), \end{aligned} \tag{6.62}$$

$$\Upsilon_2 = z_2 + l_1 \cos\left(q_1^a\right) + l_2 \cos\left(q_1^a + q_2^r\right) + \\ + l_4 \cos\left(q_1^a + q_2^r + q_3^r + q_4^r\right) + l_5 \cos\left(q_1^a + q_2^r + q_3^r + q_4^r + q_5^r\right). \tag{6.63}$$

The relabeling map can be obtained as

$$\tilde{q}_1^{a,+} = q_1^{a,-} + q_2^{r,-} + q_3^{r,-} + q_4^{r,-} + q_5^{r,-} - \pi, \tag{6.64}$$
$$\tilde{q}_2^{r,+} = 2\pi - q_5^{r,-}, \tag{6.65}$$
$$\tilde{q}_3^{r,+} = \pi - q_4^{r,-}, \tag{6.66}$$
$$\tilde{q}_4^{r,+} = \pi - q_3^{r,-}, \tag{6.67}$$
$$\tilde{q}_5^{r,+} = 2\pi - q_2^{r,-}, \tag{6.68}$$
$$\dot{\tilde{q}}_1^{a,+} = \dot{q}_1^{a,+} + \dot{q}_2^{r,+} + \dot{q}_3^{r,+} + \dot{q}_4^{r,+} + \dot{q}_5^{r,+}, \tag{6.69}$$
$$\dot{\tilde{q}}_2^{r,+} = -\dot{q}_5^{r,+}, \tag{6.70}$$
$$\dot{\tilde{q}}_3^{r,+} = -\dot{q}_4^{r,+}, \tag{6.71}$$
$$\dot{\tilde{q}}_4^{r,+} = -\dot{q}_3^{r,+}, \tag{6.72}$$
$$\dot{\tilde{q}}_5^{r,+} = -\dot{q}_2^{r,+}. \tag{6.73}$$

6.1.2 5-link with absolute orientation referenced to the torso

This subsection describes further the model for the five link planar bipedal robot depicted on the right part of the figure 2.1. The AO of the robot is given with respect to the vertical and the torso. The absolute angles $\boldsymbol{\phi}$ defined with respect to horizontal and beginning of each link are given as

$$\phi_1 = q_3^a + q_2^r + q_1^r - \pi, \tag{6.74}$$
$$\phi_2 = q_3^a + q_2^r - \pi, \tag{6.75}$$
$$\phi_3 = q_3^a, \tag{6.76}$$
$$\phi_4 = q_3^a + q_4^r, \tag{6.77}$$
$$\phi_5 = q_3^a + q_4^r + q_5^r. \tag{6.78}$$

The entries of the matrices $\mathbf{D}$, $\mathbf{C}$ and vector $\boldsymbol{G}$ for the 5DoF 5-link robot can be obtained using symbolic software by rederiving the matrices or by using the canonical change of coordinates. Let $\bar{\boldsymbol{q}} = \boldsymbol{F}$ be a local change of coordinates. If the velocities are expressed as $\dot{\bar{\boldsymbol{q}}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}$ then the kinetic energy of the robot is given — according to [Westervelt et al. (2007)] — as

$$\bar{K}(\bar{\boldsymbol{q}}, \dot{\bar{\boldsymbol{q}}}) = \frac{1}{2}\dot{\bar{\boldsymbol{q}}}^T \bar{\mathbf{D}}(\bar{\boldsymbol{q}})\dot{\bar{\boldsymbol{q}}}, \tag{6.79}$$

where

$$\bar{\mathbf{D}}(\bar{\boldsymbol{q}}) = \frac{\partial \boldsymbol{F}(\boldsymbol{q})^T}{\partial \boldsymbol{q}}\mathbf{D}(\boldsymbol{q})\frac{\partial \boldsymbol{F}(\boldsymbol{q})}{\partial \boldsymbol{q}}\bigg|_{\boldsymbol{q}=\boldsymbol{F}^{-1}(\bar{\boldsymbol{q}})}. \tag{6.80}$$

The potential energy is

$$\bar{V}(\bar{\boldsymbol{q}}) = V(\boldsymbol{q})\Big|_{\boldsymbol{q}=\boldsymbol{F}^{-1}(\bar{\boldsymbol{q}})}. \tag{6.81}$$

The transformation

$$\begin{bmatrix} \bar{\boldsymbol{q}} \\ \dot{\bar{\boldsymbol{q}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}(\boldsymbol{q}) \\ \dfrac{\partial \boldsymbol{F}(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} \end{bmatrix} \tag{6.82}$$

is called a canonical change of coordinates. Using (6.80) – (6.82) and (2.17) one can obtain the matrices $\mathbf{D}$, $\mathbf{C}$ and $\boldsymbol{G}$ from matrices derived in previous subsection for the case of 5-link with AO defined with respect to the horizontal and the stance leg. The relabeling map can be obtained as

$$\tilde{q}_1^{r,+} = q_5^{r,-}, \qquad \dot{\tilde{q}}_1^{r,+} = \dot{q}_5^{r,+}, \tag{6.83}$$
$$\tilde{q}_2^{r,+} = q_4^{r,-}, \qquad \dot{\tilde{q}}_2^{r,+} = \dot{q}_4^{r,+}, \tag{6.84}$$
$$\tilde{q}_3^{a,+} = q_3^{a,-}, \qquad \dot{\tilde{q}}_3^{a,+} = \dot{q}_3^{a,+}, \tag{6.85}$$
$$\tilde{q}_4^{r,+} = q_2^{r,-}, \qquad \dot{\tilde{q}}_4^{r,+} = \dot{q}_2^{r,+}, \tag{6.86}$$
$$\tilde{q}_5^{r,+} = q_1^{r,-}, \qquad \dot{\tilde{q}}_5^{r,+} = \dot{q}_1^{r,+}. \tag{6.87}$$

Bibliography

A. Isidori (1999). *Nonlinear control systems*, volume 242 of *Lecture Notes in Control and Information Sciences*. Springer London, London.

An, C., Atkeson, C.G., and Hollerbach, J. (1985). Estimation of Inertial Parameters of Rigid Body Links of Manipulators. *Proceedings of Conference on Decision and Control*, 24, 990–995.

Anderle, M. and Čelikovský, S. (2014). Cyclic walking-like trajectory design and tracking in mechanical chain with impacts. In *Proceedings of the XXIst Congreso ACCA 2014*, p. 341–346. Santiago de Chile.

Anderte, M., Celikovsky, S., and Dolinskc, K. (2015). Simple model of underactuated walking robot. In *2015 10th Asian Control Conference (ASCC)*, 1–6. IEEE.

Armstrong, B. (1987). On finding 'exciting' trajectories for identification experiments involving systems with non-linear dynamics. *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 4, 1131–1139.

Babitsky, V.I. (1998). *Theory of Vibro-Impact Systems and Applications*. Foundations of Engineering Mechanics. Springer Berlin Heidelberg, Berlin, Heidelberg.

Brogliato, B. (1999). *Nonsmooth Mechanics*. Communications and Control Engineering. Springer London, London.

Buss, B.G., Hamed, K.A., Griffin, B.A., and Grizzle, J.W. (2016). Experimental results for 3D bipedal robot walking based on systematic

optimization of virtual constraints. In *2016 American Control Conference (ACC)*, volume 2016-July, 4785–4792. IEEE.

Capisani, L.M., Ferrara, A., and Magnani, L. (2007). MIMO identification with optimal experiment design for rigid robot manipulators. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*.

Chevallereau, B.C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E.R., Canudas-de wit, C., and Grizzle, J.W. (2003). Designing, building, and controlling an experimental platform for the study of walking. *IEEE Control Systems Magazine*, 57–79.

Dolinsky, K. and Celikovsky, S. (2017). Application of the Method of Maximum Likelihood to Identification of Bipedal Walking Robots. *IEEE Transactions on Control Systems Technology*, DOI:10.1109/TCST.2017.2709277.

El Yaaqoubi, N.L. and Abba, G. (2009). Identification of physical parameters including ground model parameters of walking robot rabbit. In *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, 269–276.

Gautier, M., Janot, A., and Vandanjon, P.O. (2008). DIDIM: A new method for the dynamic identification of robots from only torque data. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2122–2127.

Gautier, M. and Khalil, W. (1991). Exciting trajectories for the identification of base inertial parameters of robots. *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, (4), 2–7.

Gautier, M. and Poignet, P. (2001). Extended Kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12), 1361–1372.

Gautier, M., Janot, A., and Vandanjon, P.O. (2013a). A new closed-loop output error method for parameter identification of robot dynamics. *IEEE Transactions on Control Systems Technology*, 21(2), 428–444.

Gautier, M., Jubien, A., and Janot, A. (2013b). A new iterative online dynamic identification method of robots from only force/torque data. In *2013 9th Asian Control Conference, ASCC 2013*.

Grizzle, J.W., Hurst, J., Morris, B., Park, H.w., and Sreenath, K. (2009). MABEL , A New Robotic Bipedal Walker and Runner. 2030–2036.

Grizzle, J., Abba, G., and Plestan, F. (2001). Correction to "Asymptotically stable walking for biped robots: analysis via systems with impulse effects". *IEEE Transactions on Automatic Control*, 46(3), 513–513.

Hurmuzlu, Y. and Marghitu, D.B. (1994). Rigid Body Collisions of Planar Kinematic Chains With Multiple Contact Points. *The International Journal of Robotics Research*, 13(1), 82–92.

Janot, A., Gautier, M., Jubien, A., and Vandanjon, P.O. (2014a). Comparison between the CLOE method and the DIDIM method for robots identification. *IEEE Transactions on Control Systems Technology*, 22(5), 1935–1941.

Janot, A., Olivier Vandanjon, P., and Gautier, M. (2014b). An instrumental variable approach for rigid industrial robots identification. *Control Engineering Practice*, 25(1), 85–101.

Janot, A., Vandanjon, P.O., and Gautier, M. (2014c). A Generic Instrumental Variable Approach for Industrial Robot Identification. *IEEE Trans. Control Syst. Technol.*, 22(1), 132–145.

Khalil, H.K. (2002). *Nonlinear Systems*. Prentice-Hall, Upper Saddle River.

Landau, L.D. and Lifshitz, E.M. (1976). *Mechanics, Third Edition: Volume 1*.

Lebastard, V., Aoustin, Y., and Plestan, F. (2006). Observer-based control for absolute orientation estimation of a five-link walking biped robot. *14th Mediterranean Conference on Control and Automation, MED'06*, 2522–2527.

McGeer, T. (1990). Passive Dynamic Walking. *The International Journal of Robotics Research*, 9(2), 62–82.

Olsen, M.M. and Petersen, H. (2001). A new method for estimating parameters of a dynamic robot model. *IEEE Transactions on Robotics and Automation*, 17(1), 95–100.

Olsen, M.M., Swevers, J., and Verdonck, W. (2002). Maximum Likelihood Identification of a Dynamic Robot Model: Implementation Issues. *The International Journal of Robotics Research*, 21(2), 89–96.

Park, H.W., Sreenath, K., Hurst, J.W., and Grizzle, J.W. (2011). Identification of a Bipedal Robot with a Compliant Drivetrain: Parameter estimation for control design. *IEEE Control Systems*, 31(2), 63–88.

Poignet, P. and Gautier, M. (2000). Comparison of weighted least squares and extended Kalman filtering methods for dynamic identification of robots. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 4(April), 3622–3627.

Simon, D. (2006). *Optimal State Estimation*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Swevers, J., Ganseman, C., De Schutter, J., and Van Brussel, H. (1996). Experimental Robot Identification Using Optimised Periodic Trajectories.

Swevers, J., Ganseman, C., Bilgin Tükel, D., De Schutter, J., and Van Brüssel, H. (1997). Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5), 730–740.

Westervelt, E., Chevallereau, C., Morris, B., Grizzle, J., and Ho Choi, J. (2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*, volume 26 of *Automation and Control Engineering*. CRC Press.

Publications of the author

Publications related to the thesis

Journal publications with impact factor

I. K. Dolinský, S. Čelikovský. Application of the Method of Maximum Likelihood to Identification of Bipedal Walking Robots. *IEEE Transactions on Control Systems Technology (In press, availabe in Early Access)*, DOI: 10.1109/TCST.2017.2709277, Print ISSN: 1063-6536, Online ISSN: 1558-0865, 2017. Co-authorship: 70%.

Publications indexed in Web of Science

III. M. Anderle, S. Čelikovský, K. Dolinský. Simple model of underactuated walking robot. *Proceedings of the 10th Asian Control Conference (ASCC)*, 2015. Co-authorship: 33%.

IV. K. Dolinský, S. Čelikovský. Polynomial Regression Aided Identification Method for a Class of Mechanical Systems. *Mediterranean Conference on Control and Automation (MED)*, 924–929, 2014. Co-authorship: 50%.

V. K. Dolinský, S. Čelikovský. Kalman Filter under Nonlinear System Transformations. *American Control Conference (ACC)*, 4789–4794, 2012. Co-authorship: 50%.

Other publications

VI. K. Dolinský, S. Čelikovský. Splines smoothing assisted least-squares identification of robotic manipulators. *Proc. of the Memorias del XVI Congreso Latinoamericano de Control Automático (CLCA)*, 702–707, 2014. Co-authorship: 50%.

VII. K. Dolinský, S. Čelikovský. Adaptive Nonlinear Tracking for Robotic Walking. *Cybernetics and Physics*, 1:1:28-35, 2012. Co-authorship: 50%.

VIII. K. Dolinský, S. Čelikovský. Estimation of Viscous Friction Parameters in Acrobot. *Proc. of the 5th International Scientific Conference on Physics and Control.* 2011. Co-authorship: 50%.

Publications unrelated to the thesis

Peer-reviewed journals with no impact factor

IX. K. Dolinský, A. Jadlovská. Application of Results of Experimental Identification in Control of Laboratory Helicopter Model, *Advances in Electrical and Electronic Engineering*, Ostrava, CZ, vol. 9, no. 4, p.p. 157-166, 2011, ISSN 1804-3119. Co-authorship: 50%.

Other publications

X. K. Dolinský, A. Jadlovská. Implementácia výsledkov experimentálnej identifikácie v riadení výukového modelu helikoptéry, *Electrical Engineering and Informatics: Proc. of the FEI TU of Košice*, Slovakia, 2010, pp. 546-550, ISBN 978-80-553-0460-1. Co-authorship: 50%.

XI. A. Jadlovská, K. Dolinský, R. Lonščák. Application of Designed Program Modules in C# Language for Simulation of Models of Dynamic Systems. *Proc. of the 17th International Conference on Process Control '09*, Slovakia, 2009, pp. 534–547. Co-authorship: 33%.