Report on R&D 2

Dynamic Robot Model Parameter Identification via Domain Specific Optimization

Jeyaprakash Rajagopal*

B-IT Master Studies in Autonomous Systems

University of Applied Sciences Bonn-Rhein-Sieg

Advisors: Prof. Dr Paul G. Plöger†
Sven Schneider‡

Dec 20, 2017

*jeyaprakash.rajagopal@smail.inf.h-brs.de
†paul.ploeger@h-brs.de
‡sven.schneider@h-brs.de

Contents

Abstract

An accurate trajectory tracking in the end-effector is the fundamental requirement for most of the manipulation related tasks. The standard industrial robot controller's execution of a trajectory results in a lot of deviations, and the solution for this particular problem can be found by adapting to the advanced model based controllers where the dynamic effects are also accounted through the control laws. Another important perspective that needs to be considered is the parameter estimation of the manipulator links. This work presents the domain specific parameter estimation algorithm for the robot manipulator and the aim of this project is to estimate the dynamic model parameter of the manipulator links with the use of the joint motion and the torque data. Another important objective is to estimate the static friction in all the individual joints of the robot. This work uses the modified Newton-Euler formulation that is simplified in a matrix form that linearly relates the forces or torques through the acceleration matrix to the mass inertia of the rigid body and this linearity is useful when using the model fitting method called ridge regression for the identification. The estimation algorithm is investigated and the findings are presented in this work. The results of this estimation are not plausible when comparing it with the specifications provided by the manufacturers. Another important contribution of this work is the augmentation of the differences between the use of the direct differentiation of the joint acceleration and the other method is based on signal processing concepts. The static friction is estimated from all the youBot joints and those estimates can be used in the advanced model based controller in order to resolve one of the external effects that affects the manipulator movements. This work can be used as a basis to find the accurate model parameters which is the basic need of a computed torque-control and predictive control.

Keywords: *Domain-specific parameter estimation, Rigid-body dynamics, Manipulator kinematics, Friction, Trajectory parameterization and optimization.*

Notations and Acronyms

Notations

$\mathbf{g}$ Acceleration due to gravity.

$\mathbf{M}$, $\mathbf{C}$ Manipulator inertia matrix, Coriolis/Centrifugal forces respectively

$\mathbf{N}$ Vector of external forces acting on the manipulator

$\boldsymbol{\tau}$, n Joint torque and torque vector respectively

$\mathbf{L}$ Angular momentum

$\boldsymbol{\omega}$ Angular velocity

$\dot{\boldsymbol{\omega}}$ Angular acceleration

$\boldsymbol{\Phi}, \boldsymbol{\psi}$ Dynamic model parameters such as mass, center of mass, moment of inertia before and after the projection to the joint's internal axis respectively

$\mathbf{w}$ Wrench includes force and torque vectors

$\mathbf{I}$ Moment of inertia

$\boldsymbol{\theta}$, $\mathbf{p}$ represents joint position

$\dot{\boldsymbol{\theta}}$, $\dot{\boldsymbol{p}}$ represents joint velocity

$\ddot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{p}}$ represents joint acceleration

$\mathbf{a}$, $\ddot{\boldsymbol{q}}$ Body acceleration

$[\mathbf{t}\times]$ Skew symmetric matrix

$\mathbf{R}$ Rotation matrix

$\mathbf{T}$ Transformation between the frames

$\boldsymbol{T_w}$ Transmission matrix

$\mathbf{v}$ Spatial velocity vector or a twist

$\mathbf{a}$ Spatial acceleration vector

$\mathbf{F}$ Spatial force vector

$\mathbf{M}$ Motion transform

$\mathbf{X}$ Force transform

$\boldsymbol{P_e}, \boldsymbol{I_e}$ Position and integral error respectively

Acronyms

DoF Degrees-of-Freedom

OROCOS Open RObot COntrol Software

KDL Kinematics Dynamics Library

COBYLA Constrained Optimization BY Linear Approximations

3D 3 Dimensional

COM Center Of Mass

Nm Newton meter

RMS Root Mean Square

DSP Digital Signal Processing

Hz Hertz

List of Figures

List of Tables

Acknowledgements

I would like to thank Prof. Dr Paul G. Plöger and Sven Schneider for supervising this project and I am really grateful for the motivation and support that I have got throughout this project. My special thanks to Sven Schneider for assisting me in the experimentation of the static friction estimates.

1 Introduction

An accurate dynamic model is the basic need of an advanced model based controller to execute the motions that are programmed. A model represents the physical properties of the robot which comprises kinematic, geometrical and dynamic properties. The ultimate goal of the robot manipulator is to interact/move in the environment and the motion of the rigid bodies depend on their own interactions with the environment. The motion of the joints are described by the position, velocity and acceleration inputs. The resulting torque from the joint motion can be applied directly in the joints of the manipulator and the manipulator dynamics describe the movement of the joints with respect to the applied torques/forces [23]. The manipulator is represented as a serial chain of rigid bodies that are connected together through joints. The dynamic equations are generated based on the robot model that has been constructed as a kinematic chain. The mass distribution that comprises mass, and moment of inertia have a significant relation on the motion of the rigid body. The dynamic robot models define the relationship between the manipulator motion and the torques generated by it's actuators. The existing model parameters provided by the manufacturers are basically extracted from the CAD model or the computer generated models and the accurate model parameters are mandatory for most of the control problems. The main purpose of estimating the model parameter is to obtain an accurate model that generates joint torques that are close-enough to the real robot. In this work, recursive Newton-Euler formulation [1] is used to derive the dynamic equations of the chain of rigid bodies. This work also focuses on modeling the static friction torque and the compensation techniques are discussed. The basic definitions of this section are referred from [1] [12].

Friction opposes the motion of a rigid body which is in contact with the other rigid bodies through the joints. There are three friction problems such as impending, status unknown and relative motion. Impending motion describes the motion that is yet to happen and status unknown uses the equilibrium equations to determine the friction force direction on the surface and finally the relative motion deals with the rigid bodies that are in contact. The friction estimates are not provided by the manufacturers and the prediction of these parameters when the robot is in motion is rather complex. So, it is necessary to model the friction parameters to avoid the effects on the manipulator movements due to the friction phenomenon. The control the frictional effect [8] have gained a lot of importance in the field of robotics over the years and it is necessary to provide the modeling and compensation of the frictional effects which act on the rigid body. Since friction can cause undesirable effects in a mechanical motion systems such as steady-state errors and limit cycling. This work aims to model the frictional forces on the youBot joints in order to minimize the external effect due to friction. Friction can be classified into static and kinetic/dynamic friction and the same is depicted in the Fig. 1.1 where the applied force is getting higher than the frictional force causes motion. Static friction is going to be modeled in this work since modeling the kinetic/dynamic friction that occurs when the robot is in motion. The static friction describes that the rigid body that is in contact with the other body which doesn't involve any motion. Secondly, when a rigid body is in motion with some velocity then it is considered as kinetic friction. This work considers the relative motion between the rigid bodies and the model is obtained for the same purpose. An important aspect to consider in this concept is the friction coefficient and it's value depends mainly on the kind of contact surface between the rigid bodies. This work focuses on the rigid bodies that can't change it's shape.

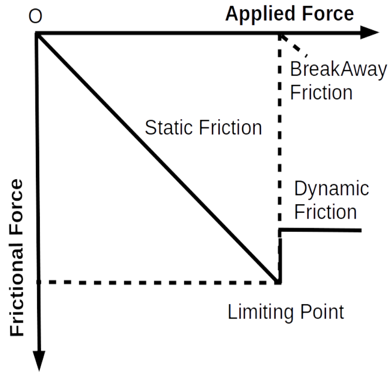

Figure 1.1: Friction mechanism.

Trajectory parameterization and optimization techniques have been investigated that helps generating the periodic and band-limited inputs that can cover the complete workspace of the robot. This can be used to achieve the better estimation of the dynamic robot model parameters. There are many optimization techniques available in the field of robotics and the method discussed by Swevers et. al. [23] provides a finite Fourier series based parameterization and the optimization is achieved through the d-optimality criterion. This work does the augmentation on the possible improvements in the estimation of dynamic robot model parameters with the help of the trajectory parameterization and optimization techniques which uses the digital signal processing methods. The modules discussed and implemented in this project would improve the control of the manipulator movements considerably.

1.1 Contributions

The contributions of this research and development project are:

- Identification of the dynamic model parameters such as mass, inertial parameters and the COM extracted from the moments
- Static friction estimation based on experimentation
- The basic computed PI control for youBot base and the manipulator
- Augmentation of the trajectory parameterization and optimization
- The logic in order to validate the identified model parameters

1.2 Report structure

The report outline is

- Section 2 state of the art analysis on the parameter estimation, static friction and augmentation of special motions.
- Section 3 describes the hardware, software platform and the method that are used to estimate the dynamic robot model parameters. This section discusses the static friction model, compensation as well and the augmentation about using the trajectory optimization techniques for the better estimation.
- Section 4 the estimation of the dynamic model parameter results and the static friction estimates.
- Section 5 presents the conclusions and possible future work.

2 Related work

This section contains the brief explanation about the state-of-the-art analysis of this work.

2.1 Identification of the dynamic robot model parameters

This section provides the state-of-the-art analysis of the dynamic model parameter estimation/identification procedure and the following context answer two important questions such as why the existing model parameters are not sufficient? and what is the need of estimating the dynamic model parameters of the manipulator links based on the domain-specific optimization technique?. The existing model parameter are extracted from the computer based models or CAD design based methods which produces significant deviation from the expected movements, so there is a need to estimate the model parameters in order to minimize the dynamic effects on the model based controllers and to generate the joint torques which are close-enough to the actuator torques [1]. Dwiputra et al. [6] presents the result based on the modelica model produces deviation in the trajectory tracking when comparing the real robot's trajectory due to the discrepancies in the model. The results of the model based controllers suffer in accuracy due to the fact that the model discrepancies and the dynamic effects have a huge impact on the movements of the manipulator, this work does not address the problem of kinematic model discrepancies. Basically the models are highly parameterized hence it brings the challenge along when there is a need of the precise control. There are many general optimization techniques used for the estimation and there are many libraries available to do the optimization such as nlopt[1], Scilab[4] which uses many optimization algorithms [17] in order to optimize the model parameters based on linear approximations. The precise control of the manipulator movements in the computed torque-control [10] required an accurate model and the model used in this work is not justified whether estimation happened through the general optimization[1] techniques or the domain-specific technique.

Whereas An et. al. [1] made use of the recursive Newton-Euler formulation which is a domain-specific estimation technique to estimate/identify the dynamic model parameters such as mass, center of mass from the first moments and the moment of inertia. The similar technique with some changes is used in [11] to estimate the link parameters in an effective manner based on the linear least squares technique. The domain specific optimization technique allows us to do the estimation procedure based on the kinematic properties of the manipulator, motion of the joints and the resulting torque/force of the robot manipulator. Whereas the general estimation/optimization techniques use different algorithms for this purpose which has a challenge in bringing the close enough accuracy on the final result and it is mandatory to guide the general optimizers in order to bring the closeness with the properties of the parameter identification.
Swevers et al. [23] proposed the identification procedure that estimates the dynamic model parameters based on the optimized robot excitation trajectories. The finite Fourier series is used

[1] `http://ab-initio.mit.edu/wiki/index.php/NLopt_Reference`

in this approach to generate the periodic excitation trajectories. The velocity and acceleration is computed in the frequency domain by differentiating the measured joint positions to improve the efficiency of the estimation process. The high frequency noises in the frequency domain are removed hence the differentiation would be free of noise that improves the system identification. The measured joint positions are used in the differentiation since the measured data differs significantly from the given joint position due to the accumulation of errors in the Cartesian space. Olsen et. al. [14] proposed an identification algorithm that uses no special test motions that is based on a single axis of the rotary joints, this work is the proceedings by the same author from the concept where the special test motions were mandatory. Olsen et al. [15] proposed the maximum likelihood dynamic model identification procedure in the proceedings that allowing the estimation procedure use the noisy joint positions and the derived velocity, acceleration which still allows the efficient estimation of the parameters. Gautier et. al. [9] proposes a method that estimates both the dynamic, friction parameters by sampling the torque data in a lower sample rate. It can be seen that the periodic excitation method replaces numerical differentiation which improves the result of the estimation procedure. This work uses the estimation procedure that is same as the reference [1].

2.2 Friction modeling and compensation

The identification results of the computer torque-control [10] can be improved by modeling and compensating the external effects such as friction and damping. Dwiputra et. al. [6] discusses that the external effects would have a significant improvement in the control of the manipulator. Friction is a non-linear phenomenon which introduces the control problems in a robot manipulator such as static position errors, limit-cycles and stick-slip error [22] and it can be classified into static and kinetic friction. Friction force is a function of load and the direction of the manipulator's velocity is called Coulomb friction and the relation with the lubricants is called Viscous friction [23]. The compensation technique that can be applied to the actuators with high pre-sliding stiffness discussed gives a compensation of the frictional torques [2] is represented in the Fig. 2.1. The adaptive friction compensation technique [27] proposes an adaptive control algorithm for the robot dynamics with both the static and dynamic friction.

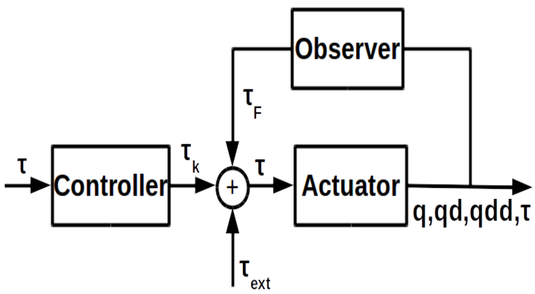

Figure 2.1: Friction compensation technique based on the article [2].

The Fig. 2.1 explains the observer based on-line compensation technique where the torque is given as an input to the controller and the control law has the feedback loop that resolves the error after the actuator's execution of a trajectory.

2.3 Trajectory parameterization and optimization

The random trajectories [18] were generated through Moveit [21] planning interface and it influences the accuracy of the estimation of the dynamic robot model parameters considerably. So, it is important to parameterize the trajectories in a periodic and it should be within the limit of the joints. This technique covers the robot's workspace with the lower frequency and the resulting parameterized and optimized trajectories are used in the identification of the system improves the accuracy of the estimation itself. Trajectory parameterization is an important method that uses finite Fourier series to create the periodic, band-limited samples[23]. The optimization is based on the d-optimality identifies the relevant values for the trajectory parameters.

3 Approach

3.1 Description of the hardware platform

The robot used in this project is KUKA youBot. It consists of a base and the manipulator that is mounted on the base as shown in Fig. 3.1. The youBot base has 4 omni-directional wheels and each of these wheels can be controlled individually or collectively for the purpose of navigation. The communication with both the base and the manipulator can be established via ether-CAT connection. The youBot manipulator has five rotary joints which are attached between the links introduces the joint constraints which makes the manipulator as a 5 DoF arm. If the rigid bodies have no joint constraints in between it leaves the rigid body with 6 DoF which is floating in the open space. Each joint has an encoder attached to it which measures the relative position of the joints which then can be used in the transformations between the coordinate frames. The encoder data is not accurate due to many factors e.g. aging of the sensors and there will be position errors in the measured joint positions. So, the measured data is used as the joint positions instead of using the given joint positions. The youBot base and the manipulator can be controlled via position, velocity or torque control modes. In this project, all the three modes are made use of and the youBot base has been used to verify the logic of measuring the static friction that will also be tested in the manipulator's individual joints. The direct testing of this logic in the robot manipulator could cause damage to the motors or to the manipulator itself.

Figure 3.1: The KUKA youBot with the 5DoF manipulator. Image courtesy[1]

[1] `https://static.generation-robots.com/5045/kuka-youbot-robot-mobile-omni-directionel-avec-bras.jpg`

3.1.1 Kinematics of the manipulator and the conventions

Kinematics is one of the important branches of study in classical mechanics which enables the possibility of studying the motion of the rigid bodies without taking the forces, moments into consideration which actually induces the motion [5]. In contrast, the relationship between the motion of the rigid bodies and the forces/torques is studied in rigid body dynamics. This work considers the n number of joints with the n-1 number of links and the joints considered in this work are revolute. The robotic manipulator is composed of a set of rigid bodies that can't reform and it is connected through the joints with the gripper attached in the end-effector. It is quite normal that the base of the manipulator is fixed in most of the cases but it is in use for a few cases and the motion occurs by commanding the joint information such as joint angle, velocity and acceleration to the joints then the links would also move from a particular pose to another. This project uses the kinematic and geometrical information from the manufacturer [26] and the specifications are assumed to be right. The serial chain of rigid bodies connected through joints are represented in a kinematic chain and each rigid body's frame is mapped to another frame with the help of a compact matrix form which is a 4×4 homogeneous transformation matrix and the transform can be given as follows

$$ {}^{i}_{i+1}\boldsymbol{T} = \begin{bmatrix} & {}^{i}\boldsymbol{R}_{i+1} & & {}^{i}_{i+1}\boldsymbol{t} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3.1} $$

where R represents the rotation matrix which is a 3×3 matrix that describes how the orientation of a frame i+1 is represented with respect to the frame i. The translation between two frames are represented with the term t which is a 3×1 vector. Basically, the frame description is formed by combining the translation vector and the rotation matrix and the combination represents the origin of a frame with respect to another embedding frame. The frame of reference in a body fixed point is mandatory in computing the transforms between all the links. The transformations between the frames can be computed in two different ways such as forward and backward recursion. The forward recursion is used in this project and it is briefed in the following sections. The schematic representation of the youBot manipulator based on the youBot-store model when it is in the candle configuration is pictorially given in Fig. 3.2. The kinematics of the robot manipulator can be analyzed in three different ways such as position, velocity and acceleration kinematics. These three methods can be used to compute the recursion through the kinematic chain with two different techniques such as forward and inverse kinematics but this approach uses the forward position, velocity and acceleration kinematics. The position kinematics consider the relative references from the frame i+1 to i whereas velocity and acceleration kinematic analysis use the forward recursion by composing the velocity and acceleration of the previous link with the contribution of the current link.

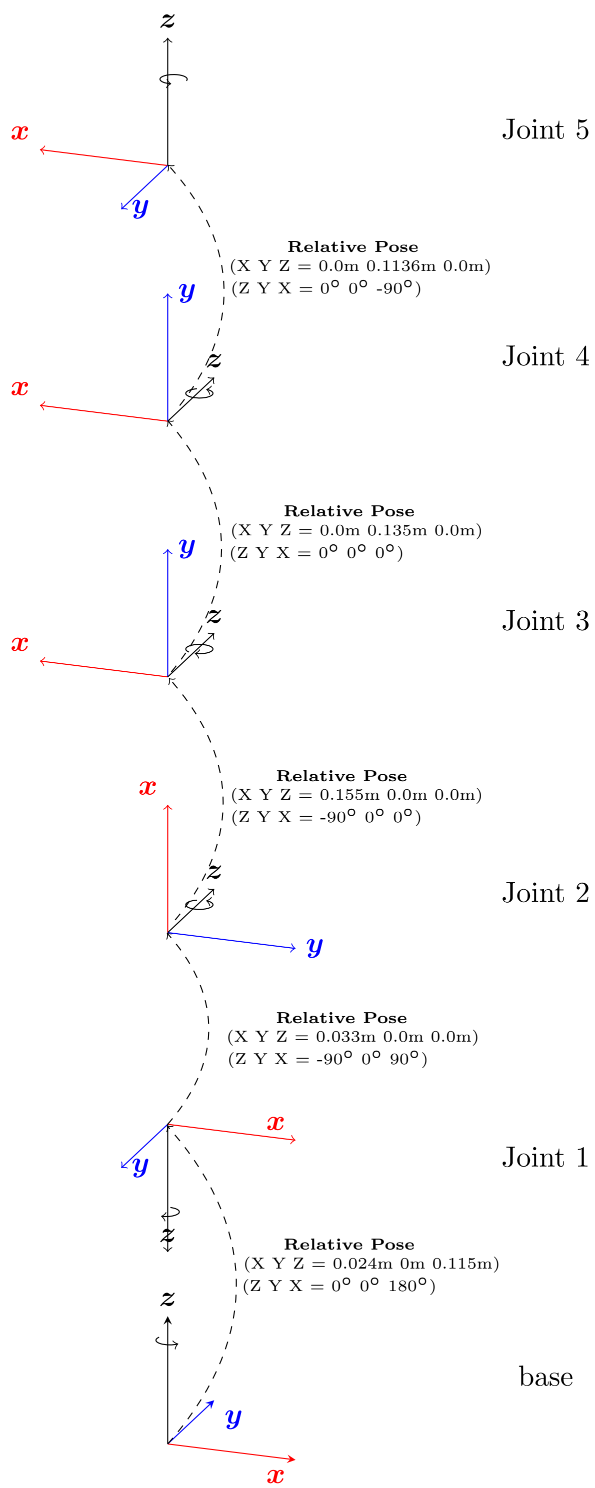

Figure 3.2: Frame representation of the youBot-store dynamic robot model where the youBot manipulator is in candle configuration which is in fact the default in the given kinematic specifications [26]. This diagram is referred from the article [18].

The coordinate system of the youBot manipulator considers the zero positions to be the home position of the manipulator. It means that the absolute joint angles which are measured from the manipulator joints have to go through the offset computations in the kinematic analysis. In this work, youBot-store model is used and the kinematic analysis differs from the coordinate frames used by the real robot and both these coordinate frames are given in the table 3.1.

Table 3.1: The robot manipulator's(youBot manipulator) coordinate frame and the kinematic coordinate frame for the youBot-store model

CF	Joint	Minimum	Straight up	Maximum
youBot	1	0°	169°	338°
	2	0°	65°	155°
	3	0°	-146°	-297°
	4	0°	102.5°	205°
	5	0°	167.5°	335°
youBot-store kinematic	1	-169°	0°	169°
	2	-65°	0°	90°
	3	146°	0°	-151°
	4	-102.5°	0°	102.5°
	5	167.5°	0°	167.5°

3.1.2 Extrinsic to intrinsic conversion

The equations used in this section are referred based on the article [5]. The intrinsic convention is the most widely used convention since the visualization of the frames are simpler in a geometrical system. The youBot model specification uses extrinsic Tait-Bryan angles with the ZYX order. The extrinsic rotation performs the rotation around a fixed axis and the intrinsic rotation performs the rotation around the moving axis. The extrinsic rotation around the Cartesian axes in some order is equal to the intrinsic rotation around the axes in the opposite order. The rotation about a particular co-ordinate axis can be represented for the individual axis(elementary rotations) as

$$\boldsymbol{R_x(\alpha)} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{cos(\alpha)} & \boldsymbol{-sin(\alpha)} \\ \mathbf{0} & \boldsymbol{sin(\alpha)} & \boldsymbol{cos(\alpha)} \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{R_y(\beta)} = \begin{bmatrix} \boldsymbol{cos(\beta)} & \mathbf{0} & \boldsymbol{sin(\beta)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{-sin(\beta)} & \mathbf{0} & \boldsymbol{cos(\beta)} \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{R_z(\gamma)} = \begin{bmatrix} \boldsymbol{cos(\gamma)} & \boldsymbol{-sin(\gamma)} & \mathbf{0} \\ \boldsymbol{sin(\gamma)} & \boldsymbol{cos(\gamma)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3.4}$$

Euler angles used to specify the orientation of an object that performs rotation around two axes only i.e. Z-Y-Z, whereas the Tait-Bryan angles rotate about three distinct Cartesian axis i.e. the first angle specify the rotation around z-axis, the second rotation around y-axis, and the final rotation about x-axis. An orientation of the frame can be represented as a rotation of that frame with respect to it's frame of reference. The youbot-store model used in this work is represented with the proper Euler angles convention(extrinsic rotation). The rotations are extrinsic when the elemental rotation is performed for a Cartesian axis and the preceding rotation is relative to the world frame which doesn't change. Whereas, the intrinsic(Tait-Bryan angles) rotations perform the elemental rotation about a Cartesian axis at first and the next elemental rotation is based on the coordinate system that is the result of the previous rotation. The following matrix is used for the conversion of the extrinsic convention to the intrinsic convention

$$\boldsymbol{R_{XYZ}(\alpha,\beta,\gamma)} = \begin{bmatrix} c\beta * c\gamma & -c\beta * s\gamma & s\beta \\ s\alpha * s\beta * c\gamma + c\alpha * s\gamma & -s\alpha * s\beta * s\gamma + c\alpha * c\gamma & -s\alpha * c\beta \\ -c\alpha * s\beta * c\gamma + s\alpha * s\gamma & c\alpha * s\beta * s\gamma + s\alpha * c\gamma & c\alpha * c\beta \end{bmatrix} \tag{3.5}$$

The conversion from the extrinsic rotation to the intrinsic is achieved by inverting the rotation order as explained above. In equation (3.5), the extrinsic rotation(z, y, x) is equivalent to intrinsic (x, y, z) as it is shown below

An example ZYX =[-90°, 0°, 90°] is used to check the property that has mentioned above

When applied ZYX in extrinsic based rotation results in the matrix form after multiplying the matrices (3.4), (3.3), (3.2) which is based on a fixed axis in the following order

$$\boldsymbol{{}^{A}_{B}R_{ZYX}(\gamma,\beta,\alpha) = Rx(\gamma) \cdot Ry(\beta) \cdot Rz(\alpha)} \tag{3.6}$$

The order of rotation axis is really important when representing the rotation in a particular way. In the above equation (3.6) the first rotation happens around z-axis by $\boldsymbol{\alpha}$, then rotation around y-axis by $\boldsymbol{\beta}$ and finally x-axis around by $\boldsymbol{\gamma}$. The order of rotation happens in the equation (3.6) from right side to the left and it results with the rotation matrix

$$\boldsymbol{Extrinsic(ZYX)} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.7}$$

Intrinsic rotation results in

$$\boldsymbol{Intrinsic(XYZ)} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.8}$$

3.2 Description of the software platform

This project is developed on Ubuntu 14.04 operating system and this section briefly explains programming languages and the simulation tools used in this project.

- **c++** is used to identify the break-away friction parameters of the youBot-base and the manipulator
- **Python** [25] programming language is used for the dynamic model parameters estimation. There is an extension in python called numpy is used in this work for the simplified matrix operations and it is possible to use the linear algebra concepts. The matrix is created as an array in numpy which makes the computations simpler. There is a tool named pandas is used in this project in order to read the measured joint information from a file
- **Simbody** [19] is a simulator which can be used to simulate and verify the forward dynamics. This library is a open-source toolkit used to simulate the robots which describes the serial chain of rigid bodies. It is possible to perform forward, inverse and hybrid dynamics for the robot manipulators. The youbot-store model parameters, kinematic and geometrical parameters are used in this project to create the simulation of the youBot manipulator. This simulation can be effectively used to study the dynamics operations of the youBot manipulator. The youBot store model uses the extrinsic(fixed) frame conventions and it has to be converted to the intrinsic(Tait-Bryan angle) frame conventions in order to simulate the robot according to it's mechanism
- **Orocos KDL** [20] is used mainly for the inverse dynamics computation. This tool is useful in generating joint torques based on the joint information such as joint angle, velocity and acceleration. The results from this computation are used in validating the estimated inertial parameters
- **Difference between Simbody and KDL** are presented in this section. Simbody is a library with the powerful physics engine which facilitates the creation and the visualization of the manipulator that makes user to verify the results based on the manipulator movements. Whereas KDL library provides functionalities that build the robot by using the kinematic and geometrical specifications and the results can't be visualized. A rigid body in KDL is represented as a segment which is a combination of the joint and the link specifications such as a frame and a rigid body inertia. Wherein, Simbody provides a class called Mobilizer to attach the body and the joint together.

3.3 Modeling of the kinematics

The basic definitions of this section are referred from the articles [7] [3] [5].

3.3.1 Forward position kinematics

The position analysis of the robot manipulator can be classified into forward and inverse position kinematics. Forward position kinematics compute the pose of the end-effector in robot's workspace with the given kinematic information and the joint positions. Inverse kinematics computes the joint configuration with the given pose of the end-effector in the backward recursion. This work uses the forward position kinematics in order to find the position of the end-effector based on the given joint data. It is possible to find the kinematic equations of the rigid body using the forward kinematics in a quite straight forward and in a simple way. In contrast, it is complex to compute the inverse kinematics of the manipulator due to the possibility of the multiple solutions and the singularities. A simple matrix form that is widely used in the field of robotics that combines both the translation vector and the orientation matrix of the rigid body is called the homogeneous transformation. The homogeneous transformation is used in this work to describe the relationship between different coordinate frames.

$$\boldsymbol{H_i} = \begin{bmatrix} & \boldsymbol{R_i} & & \boldsymbol{P_i} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \boldsymbol{D_i} = \begin{bmatrix} & \boldsymbol{R_i} & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3.9}$$

where H represents the transform between the reference frame of the link and the respective tip frame, D represents the Homogeneous transformation matrix which is based on the rotation axis of the joint(revolute joint). The relative rotation R and translation t can be represented in different homogeneous transformations and these two matrices can be multiplied to get the H matrix as given in the following equation

$$\boldsymbol{H} = \boldsymbol{T_{R(x/y/z)}(R) \cdot T_t(t)} \tag{3.10}$$

where R represents the rotation matrix, t represents the translation vector. The matrix creation $\boldsymbol{T_R}$, $\boldsymbol{T_t}$ represents the relative rotation matrix and translation vector represented in the homogeneous transformation form respectively. The parameter in the $\boldsymbol{T_R}$ function in the above equation (3.10) represents the relative orientation of the link frame based on the embedding frame of the previous link. $\boldsymbol{T_t}$ function in the above equation represents the relative position of the link frame with respect to the reference frame that is attached with the previous link. The equation computes the transform based on the joint constraints can be given as follows

$$\boldsymbol{D} = \boldsymbol{T_{R(x/y/z)}(\theta) \cdot (T_t = 0)} \tag{3.11}$$

where $\boldsymbol{\theta}$ represents the joint angle and the parameter in the $\boldsymbol{T_R}$ operator in the equation (3.11) is based on the joint constraints and the possible rotations are given in the equations from (3.2) (3.3) (3.4). By multiplying the equations (3.10), (3.11), it is possible to find the transform between the pose of the successor's body frame with respect to the predecessor's coordinate frame. The transformations from the base till the final link's body frames can be multiplied together to obtain the end-effector's pose as

$$^{n}\boldsymbol{T_0} = (\boldsymbol{H_0 D_0})(\boldsymbol{H_1 D_1}) \cdots (\boldsymbol{H_n D_n}) \tag{3.12}$$

If there is no rotation change between the body frames results in the following form since the translation is accounted in the pose displacement matrix(H) computation and the rotation is an identity matrix

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3.13}$$

The pure translation can be specified in the homogeneous transformation matrix by keeping the rotation as an identity matrix as follows

$$\boldsymbol{Translation} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{t_x} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \boldsymbol{t_y} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \boldsymbol{t_z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3.14}$$

where $\boldsymbol{t_x}, \boldsymbol{t_y}, \boldsymbol{t_z}$ represents the translation vector.

3.3.2 Forward velocity kinematics

The velocity analysis of the robot manipulator can also be classified into forward and inverse velocity kinematics. In this work, the forward velocity kinematics is used in the computations. Every segment or link of the manipulator has an angular velocity $\boldsymbol{\omega}$ and the linear/translational velocity v is observed when mapping of the neighbor link's velocity. The velocity kinematics mapping occurs generally in forward recursions and body's fixed point has been mapped to the world reference frame using the motion vectors defined based on the convention used in this work.

The Plücker convention based velocity twist that is specified in the article [7] has to be changed according to the convention used in this work

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\omega_x} \\ \boldsymbol{\omega_y} \\ \boldsymbol{\omega_z} \\ \boldsymbol{v_x} \\ \boldsymbol{v_y} \\ \boldsymbol{v_z} \end{bmatrix} \tag{3.15}$$

The spatial velocity twist has been modified from the Plücker transform to the linear-before-angular convention as given in the following equation (3.16)

$$\boldsymbol{v'} = \begin{bmatrix} \boldsymbol{v_x} \\ \boldsymbol{v_y} \\ \boldsymbol{v_z} \\ \boldsymbol{\omega_x} \\ \boldsymbol{\omega_y} \\ \boldsymbol{\omega_z} \end{bmatrix} \tag{3.16}$$

The selection of the order is basically author's convenience. Where the first 3×1 coordinate vector $[\boldsymbol{v_x v_y v_z}]^{\boldsymbol{T}}$ in the equation (3.16) represents the translational velocity in the Cartesian space. The second part of the spatial vector VelocityTwist represents the angular velocity $[\boldsymbol{\omega_x \omega_y \omega_z}]^{\boldsymbol{T}}$ in the Cartesian space. The sub-space matrix that converts the joint space velocity to the Cartesian space velocity can be given based on the rotation axis of the joint and the translational part is zero due to the reason that the revolute joint is the consideration of this project

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \tag{3.17}$$

As it is known in the equation (3.17) that the revolute joint has only the angular(rotation part) velocity part in the Cartesian space and the translational velocity part is $[0\ 0\ 0]^{\boldsymbol{T}}$ in the subspace matrix representation. If the joint in prismatic then the translational vector will be the active part in the sub-space matrix. If the joint is revolute and the rotation is around the z-axis the 6×1 spatial vector can be given as

$$\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \dot{\boldsymbol{q}} \end{bmatrix} \tag{3.18}$$

The representation of a twist is different from the article [7] that represents the motion vector $\boldsymbol{^A M_B}$ from the distal links to the proximal links as

$$\boldsymbol{^A M_B} = \begin{bmatrix} \boldsymbol{R_i^T} & \mathbf{0} \\ [\boldsymbol{t}\times]\boldsymbol{R_i^T} & \boldsymbol{R_i^T} \end{bmatrix} \tag{3.19}$$

In this work, the motion vector transform is modified based on the twist convention as

$$\boldsymbol{^A M'_B} = \begin{bmatrix} \boldsymbol{R_i^T} & [\boldsymbol{t}\times]\boldsymbol{R_i^T} \\ \mathbf{0} & \boldsymbol{R_i^T} \end{bmatrix} \tag{3.20}$$

$\boldsymbol{^A M_B}$ represents the motion transform from the Cartesian frame B to the frame A. $\mathrm{R}^{\boldsymbol{-1}}$ is the rotation matrix that transforms 3D vectors from the B to A coordinates. The contribution of a particular joint can be computed by multiplying the subspace matrix with it's twist as given in the equation (3.21), for example the rotation around the z axis is ($[0\ 0\ 0\ 0\ 0\ 1]^{\boldsymbol{T}}$) and it can be multiplied with the joint velocity $\dot{\boldsymbol{q}}_{\boldsymbol{i}}$ to get the contribution of the joint which is transformed into the Cartesian space as given below

$$\boldsymbol{v_{Ji}} = \boldsymbol{z_i} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{i}} \tag{3.21}$$

In order to achieve forward velocity kinematics, the twists of the base joint must be mapped to the end-effector.

$$\boldsymbol{v'_i} = \boldsymbol{v'_{i-1}} + \boldsymbol{^0 M'_i} \cdot \boldsymbol{v_{Ji}} \tag{3.22}$$

The forward recursion is used in this work and it starts with the twist of the base and it is added to the next body of the manipulator. This operation is continued till the last joint of the manipulator. The twist of the arm fixed joint is $\mathrm{v}_{\boldsymbol{J0}} = [\mathbf{0}\ \mathbf{0}]^{\boldsymbol{T}}$. The individual joint contributions in the Cartesian frame which has been mapped into joint's internal axis after the multiplication of the sub-space matrix is represented in the form $\mathrm{v}_{\boldsymbol{Ji}}$. Then the joint contribution is mapped to a fixed point where the base coordinate system is attached. The spatial velocity of the rigid body is observed based on this fixed point when the body is at motion. The rotation matrix and translation vector used in the motion transform (3.20) are extracted after mapping the rigid body's coordinate frame to the world coordinate frame. The rotation R is inversed in the motion transform due to the fact that the recursion is forward and the transformations of the coordinate frames are happening from 0 to i in the kinematic chain but this work maps the twist velocity of the frame i back to the world coordinate frame 0.

3.3.3 Forward acceleration kinematics

The similar mapping is applied to the accelerations where the acceleration due to the joint contribution is transformed with respect to the world coordinates using the motion vector mentioned above and the spatial acceleration vector in the article [7] can be given in the following form

$$\boldsymbol{a} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_x \\ \dot{\boldsymbol{\omega}}_y \\ \dot{\boldsymbol{\omega}}_z \\ \dot{\boldsymbol{v}}_x \\ \dot{\boldsymbol{v}}_y \\ \dot{\boldsymbol{v}}_z \end{bmatrix} \tag{3.23}$$

As discussed in the velocity analysis, the linear-before-angular convention of the spatial acceleration matrix can be given as

$$\boldsymbol{a}' = \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \ddot{\boldsymbol{q}} \end{bmatrix} \tag{3.24}$$

By taking the derivative of the equation (3.21), it is possible to obtain the acceleration of the body

$$\frac{\partial \boldsymbol{\omega}_i}{\partial t} = \frac{\partial \boldsymbol{\omega}_{i-1}}{\partial t} + \boldsymbol{z}\frac{\partial \dot{\boldsymbol{q}}_i}{\partial t} + \dot{\boldsymbol{q}}_i \frac{\partial \boldsymbol{z}}{\partial t}$$

The above equation can be simplified as

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_{i-1} + \boldsymbol{z}\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{z}}\dot{\boldsymbol{q}}_i \tag{3.25}$$

where

- $\ddot{\boldsymbol{q}}$ is the joint angular acceleration
- $\dot{\boldsymbol{q}}$ is the joint angular velocity
- z is the sub-space matrix
- $\dot{\boldsymbol{\omega}}_{\boldsymbol{i-1}}$ is the acceleration of the predecessor link
- $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}}$ is the angular velocity, acceleration vector of the body respectively

The equation (3.25) is modified based on the convention used in this work as given below

$$\boldsymbol{a_i' = a_{i-1}' + {}^0M'_i \cdot (z\ddot{q}_i) + v_i' \times ({}^0M'_i \cdot v_{Ji})} \tag{3.26}$$

where

- $\boldsymbol{a_i'}$ is the spatial acceleration of the body
- $\boldsymbol{{}^0M'_i}$ is the motion transform from the point in the body i to the point attached with the world coordinate frame
- z$\ddot{\boldsymbol{q}}_{\boldsymbol{i}}$ represents the mapping of joint acceleration into the Cartesian space acceleration which is in body coordinates

In the above equation (3.26). The differentiation of the twist v$_{\boldsymbol{i}}$ is represented in the skew-symmetric matrix form as follows

$$\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} \times = \begin{bmatrix} \boldsymbol{v}\times & \boldsymbol{\omega}\times \\ \boldsymbol{\omega}\times & \boldsymbol{0} \end{bmatrix} \tag{3.27}$$

The v× acts as a differentiation operator which maps v to $\dot{\boldsymbol{v}}$. Featherstone [7] uses the convention angular-before linear hence the above equation is the modified version of the spatial vector in a skew-symmetric form that can be found in the article.

3.4 Modeling of the dynamics

It is important to brief the problem of interest in this section before analyzing the robot dynamics. The manipulator is composed of n joints and n-1 links where link 0 is the base of the manipulator, link n-1 is the last link in the kinematic chain. The accurate dynamic model parameters are the basic need in control algorithms i.e. computed torque-control in tracking the trajectories precisely and this work doesn't put any efforts on checking the accuracy of the kinematic model. The ten unknown dynamic model parameters represented in a matrix that decides the mass distribution of the rigid body are

- m - mass of the link
- m · c - the moments of the link about the COM($c_{\boldsymbol{x}}$, $c_{\boldsymbol{y}}$, $c_{\boldsymbol{z}}$). The center of mass vector has to be computed from the moments of the link
- $I_{\boldsymbol{xx}}$, $I_{\boldsymbol{yy}}$, $I_{\boldsymbol{zz}}$ - principle axis of inertia which is present in the diagonal elements of the inertial matrix
- $I_{\boldsymbol{xy}}$, $I_{\boldsymbol{xz}}$, $I_{\boldsymbol{yz}}$ - the products of inertia

This module takes the measured joint angle, velocity, acceleration and the torque as inputs in order to estimate the dynamic robot model parameters of the manipulator links. In this project, there are 5 links that go through the estimation procedure and there are 50 unknowns to be estimated based on model fitting method called ridge regression which is explained in the later part of this section. After the model parameters are estimated with the help of the joint and torque data, it is important to check the plausibility of the estimated parameters manually and it is mandatory to compare the torque prediction accuracy between the estimated model parameters and the existing model parameters. The validation procedure is explained in the experimentation section. The dynamics of the manipulator is described with the set of non-linear differential equations which depend on the kinematic and the inertial properties of the robot. The general equation of motion that combines both the translational and rotational dynamics of a rigid body can be given as

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{3.28}$$

where the equation (3.28) is based on the article [13]. $\boldsymbol{\tau}$ is the vector of net torques acting on the rigid body and M is the manipulator inertia matrix that decides the mass distribution of the rigid body. C represents the external forces such as Coriolis and centrifugal forces acting on the manipulator links and the tracking error is caused by such forces. The gravitational force vector also includes the frictional effect which depends on the velocity given in N($\boldsymbol{\theta}$,$\dot{\boldsymbol{\theta}}$). $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}$ are the joint position, velocity and acceleration respectively. A rigid body has a local coordinate system attached to it which can be mapped to the other rigid body through the coordinate system attached. Every rigid body has a COM point where the inertial frames are attached and it can be located with the c($[\boldsymbol{c_x}, \boldsymbol{c_y}, \boldsymbol{c_z}]$) vector from the reference point of the rigid body. The representation is pictorially given in the Fig. 3.3. Every rigid body has a reference and the tip frame attached in order to compute the transformation within the link itself

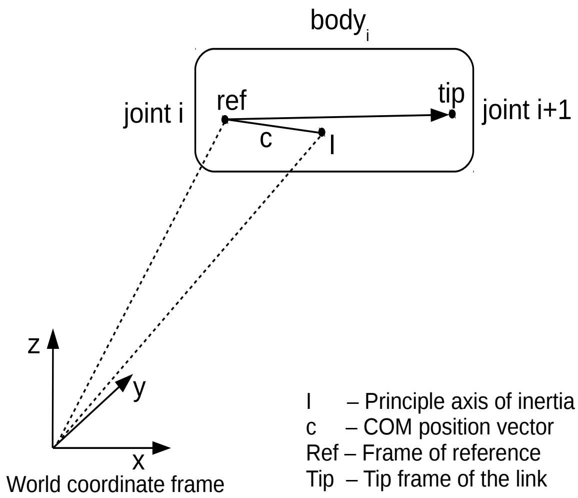

Figure 3.3: Schematic representation of a Rigid body with the coordinate frames attached to it. The reference coordinate frame attached to the body is defined with respect to the world reference frame. The COM can be located with the vector c. Inertial frame is attached in the COM represented as **I**. The **tip** reference frame is useful in computing the link transformations.

The general Newton-Euler formulation is specified in the equation (3.28) uses the joint data $(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ to compute the moments(forces in some cases) that are required to achieve the desired end-effector motion and this operation is called inverse dynamics. The equation (3.28) is formulated by combining the Newton's second law which accounts the translational motions and Euler's equation represents the rotational motion of the rigid body. The complete context of this section is referred from [1] and the implicit conventions are explained briefly. The net force acting on the rigid body can be written as

$$\boldsymbol{F_{net}} = \boldsymbol{f_{link}} + \boldsymbol{m} \cdot \boldsymbol{g} = \boldsymbol{m} \cdot \ddot{\boldsymbol{q}} \tag{3.29}$$

where, $\mathrm{F}_{\boldsymbol{net}}$ represents the net force acting on the rigid body. The gravity vector is given as **g** and **m** is mass of the body. The net force act on the rigid body comprises the gravitational force m·g and the contribution of the body itself is represented as $\boldsymbol{f_{link}}$. $\ddot{\boldsymbol{q}}$ represents the Cartesian space acceleration.

Newton's second law states that the net force acting on a rigid body is proportional to the product of the mass of the body which is a constant and the acceleration is measured about the center of mass point. The rigid body under consideration also accounts the external forces exerted by other rigid bodies.

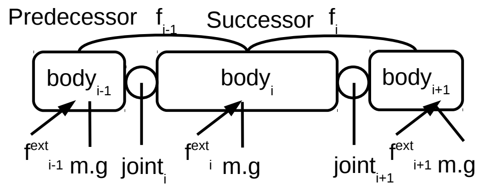

Figure 3.4: Forces acting on the rigid body

$$\sum \boldsymbol{F_i} = \boldsymbol{m} \cdot \boldsymbol{\ddot{q}} \tag{3.30}$$

is equivalent to

$$\boldsymbol{F_{net}} = \boldsymbol{f^{ext}_{i-1}} + \boldsymbol{f_{i-1}} + \boldsymbol{m} \cdot \boldsymbol{g} = \boldsymbol{m} \cdot \boldsymbol{\ddot{q}} \tag{3.31}$$

where $\boldsymbol{f^{ext}_{i-1}}$ represents the external forces acting on the rigid body. f$_{\boldsymbol{i-1}}$ represents the force exerted by the predecessor body and the gravitational forces are also accounted.

Derivation of the Newton's equation

The linear momentum of a particle can be given as

$$\boldsymbol{P} = \boldsymbol{m} \cdot \boldsymbol{\dot{q}} \tag{3.32}$$

where P represents the momentum of a particle with the unit $\boldsymbol{kg \cdot m/s}$, **m** is mass and $\boldsymbol{\dot{q}}$ represents velocity. The rate of change of linear momentum of a particle yields us the equation (3.33), since the net force is directly proportional to the rate of change in linear momentum$(\boldsymbol{m \cdot \dot{q}})$.

$$\frac{\boldsymbol{\partial P}}{\boldsymbol{\partial t}} = \boldsymbol{m} \cdot \frac{\boldsymbol{\partial \dot{q}}}{\boldsymbol{\partial t}} + \boldsymbol{\dot{q}} \cdot \frac{\boldsymbol{\partial m}}{\boldsymbol{\partial t}} \tag{3.33}$$

As, it is known that the derivation of a constant is zero. So the above equation (3.33) can be simplified to represent the force of the rigid body as

$$\boldsymbol{F} = \boldsymbol{m} \cdot \boldsymbol{\ddot{q}} \tag{3.34}$$

The similar approach is used to derive the angular motion/moments of the rigid body from the rate of change in angular momentum which is given in the following section.

Derivation of the Euler's equation

The derivation of the Euler's equation is referred from the article [13] [3]. The moment is a result of the angular motion of the rigid body about a point i.e. center of mass. The rotational motion of a rigid body based on the Euler's equation can be given as

$$\boldsymbol{\tau = I\dot{\omega} + \omega \times (I\omega)} \tag{3.35}$$

The applied net torque can be given as the difference between the angular momentum of that particular rigid body and the angular acceleration of the rigid body based on it's inertial frame of reference and

$$\boldsymbol{\tau_{net} = \tau_{link} - c_i \times f_{link} = I\dot{\omega} + \omega \times (I\omega)} \tag{3.36}$$

where I $\boldsymbol{\dot{\omega}}$ represents the change in angular momentum due to the angular acceleration and $\boldsymbol{\omega \times (I\omega)}$ represents the change in angular momentum due to the inertia tensor rotation. The angular momentum for an object can be given as

$$\boldsymbol{L = I \cdot \omega} \tag{3.37}$$

where the moment of inertia(I) can be defined as the ratio of the angular momentum(L) with it's angular velocity $\boldsymbol{\omega}$.

$$\boldsymbol{I = L/\omega} \tag{3.38}$$

The Euler's equation can be derived by equating the rate of change of angular momentum to the applied torque [13]. The derivation of the equation of motion (3.37) can be written as

$$\boldsymbol{\frac{\partial L}{\partial t} = I \cdot \frac{\partial \omega}{\partial t} + \omega \cdot \frac{\partial I}{\partial t}} \tag{3.39}$$

In order to represent the angular momentum of the rigid body with respect to the inertial frame of reference can be done by

$$\boldsymbol{I = R(t) \cdot I_{body} \cdot R(t)^T} \tag{3.40}$$

this is an important transformation to local coordinates which is multiplied with the inertia tensor of the link and transform back to the world coordinates. The time derivative of the rotation matrix equals to the change in direction, as the magnitude is constant.

$$\frac{\partial \boldsymbol{R}(t)}{\partial t} = \boldsymbol{\omega} \times \boldsymbol{R}(t) \tag{3.41}$$

The $\frac{\partial \boldsymbol{I}(t)}{\partial t}$ with respect to the inertial frame of reference can be derived as

$$\begin{aligned}
\frac{\partial \boldsymbol{I}(t)}{\partial t} &= \frac{\partial}{\partial t}(\boldsymbol{R}(t) \cdot \boldsymbol{I} \cdot \boldsymbol{R}(t)^{\boldsymbol{T}}) \\
&= \frac{\partial \boldsymbol{R}(t)}{\partial t}\boldsymbol{I}\boldsymbol{R}(t)^{\boldsymbol{T}} + \boldsymbol{R}(t)\boldsymbol{I}\frac{\partial \boldsymbol{R}(t)^{\boldsymbol{T}}}{\partial t} \\
&= (\boldsymbol{\omega} \times \boldsymbol{R}(t))\boldsymbol{I}\boldsymbol{R}(t)^{\boldsymbol{T}} + \boldsymbol{R}(t)\boldsymbol{I}((\boldsymbol{w_t}\boldsymbol{R}(t))^{\boldsymbol{T}}) \\
&= \boldsymbol{\omega} \times \boldsymbol{I}(t) - \boldsymbol{I}(t)\boldsymbol{\omega}\times
\end{aligned}$$

the result of this derivation can be applied in the equation (3.39)

$$\boldsymbol{\tau} = \boldsymbol{I}(t) \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \cdot (\boldsymbol{\omega} \times \boldsymbol{I}(t) - \boldsymbol{I}(t)\boldsymbol{\omega}\times) \tag{3.42}$$

the term $(\boldsymbol{\omega} \cdot \boldsymbol{I}(t) \cdot \boldsymbol{\omega}\times)$ gets canceled and the result of this derivation is the angular momentum vector which is the equation (3.35).

Derivation of the Newton-Euler's formulation in a matrix form

The equations and derivations of this section are based on the references [1] [7] [11]. The resultant force vector F and torque $\boldsymbol{\tau}$ vectors are represented in a single matrix which is called a wrench. Wrench is a generalized force that combines the force and the torque and it is represented in a 6×1 coordinate vector.

$$\begin{bmatrix} \boldsymbol{f_{link}} \\ \boldsymbol{\tau_{link}} \end{bmatrix} = \begin{bmatrix} \ddot{\boldsymbol{p}} - \boldsymbol{g} & [\dot{\boldsymbol{\omega}}\times] + [\boldsymbol{\omega}\times][\boldsymbol{\omega}\times] & \boldsymbol{0} \\ \boldsymbol{0} & [(\boldsymbol{g} - \ddot{\boldsymbol{p}})\times] & [\cdot\dot{\boldsymbol{\omega}}] + [\dot{\boldsymbol{\omega}}\times][\cdot\boldsymbol{\omega}] \end{bmatrix} \times \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{m} \cdot \boldsymbol{c} \\ \boldsymbol{I_{xx}} \\ \boldsymbol{I_{xy}} \\ \boldsymbol{I_{xz}} \\ \boldsymbol{I_{yy}} \\ \boldsymbol{I_{yz}} \\ \boldsymbol{I_{zz}} \end{bmatrix} \tag{3.43}$$

The above equation (3.43) can be re-written in a simplified form as given in the equation (3.44). The resulting torque is basically represented in the joint space of the manipulator and the conversion has to happen in order to represent the joint torque in the Cartesian space and the final form that is obtained is nothing but a wrench.

$$\boldsymbol{w_{link}} = \boldsymbol{A} \cdot \boldsymbol{\psi} \tag{3.44}$$

where

- $\boldsymbol{w_{link}}$ represents the 6×1 force-torque vector is also called as wrench of the link, the force-torque vector is computed based on the movement of the link
- $\boldsymbol{A}$ is the 6×10 acceleration matrix
- $\boldsymbol{\psi}$ is a matrix of interest with the unknown dynamic model parameters of the links

A matrix is considered to be skew-symmetric when the property $\boldsymbol{f_{ij}(v)} = -\boldsymbol{f_{ji}(v)}$ is satisfied. A vector(v = [x, y, z]) can be represented as a matrix in 3 dimensions with the use of skew symmetric matrix (3.45) and it can be written as $\mathbf{a} \times \mathbf{b} = \mathbf{f(a)} \cdot \mathbf{b}$ which is computed from the first vector $\mathbf{a}$. The linear mapping of a 3D vector to the corresponding skew symmetric matrix can be represented as follows

$$\boldsymbol{f(v)} = \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{z} & \boldsymbol{y} \\ \boldsymbol{z} & \boldsymbol{0} & -\boldsymbol{x} \\ -\boldsymbol{y} & \boldsymbol{x} & \boldsymbol{0} \end{bmatrix} \tag{3.45}$$

The joint acceleration $\ddot{\boldsymbol{p}}$ is mapped [24] to the body coordinates $\ddot{\boldsymbol{q}}$ by

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{p}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{c} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{c}) \tag{3.46}$$

The mapping the joint acceleration to the Cartesian space is achieved by simply combining the efforts such as joint acceleration $\ddot{\boldsymbol{p}}$ with the angular acceleration of the body itself $\dot{\boldsymbol{\omega}} \times \boldsymbol{c}$ and the Coriolis effect $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{c})$. By substituting $\ddot{\boldsymbol{q}}$ in (3.29) yields the following result

$$\begin{aligned} \boldsymbol{f_{link}} &= \boldsymbol{m}(\ddot{\boldsymbol{q}} - \boldsymbol{g}) \\ &= \boldsymbol{m}((\ddot{\boldsymbol{p}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{c} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{c})) - \boldsymbol{g}) \\ &= \boldsymbol{m}\ddot{\boldsymbol{p}} + \dot{\boldsymbol{\omega}} \times (\boldsymbol{m} \cdot \boldsymbol{c}) + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\boldsymbol{m} \cdot \boldsymbol{c})) - \boldsymbol{m} \cdot \boldsymbol{g} \end{aligned}$$

$$\boldsymbol{f_{link}} = \boldsymbol{m}(\ddot{\boldsymbol{p}} - \boldsymbol{g}) + \dot{\boldsymbol{\omega}} \times (\boldsymbol{m} \cdot \boldsymbol{c}) + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\boldsymbol{m} \cdot \boldsymbol{c})) \tag{3.47}$$

substituting the equation (3.47) in (3.36) results in

$$\begin{aligned} \boldsymbol{\tau_{link}} &= \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) + \boldsymbol{c} \times (\boldsymbol{m}(\ddot{\boldsymbol{p}} - \boldsymbol{g}) + \dot{\boldsymbol{\omega}} \times (\boldsymbol{m} \cdot \boldsymbol{c}) + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\boldsymbol{m} \cdot \boldsymbol{c}))) \\ &= \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) + ((\boldsymbol{g} - \ddot{\boldsymbol{p}}) \times \boldsymbol{mc}) + \boldsymbol{mc} \times (\dot{\boldsymbol{\omega}} \times \boldsymbol{c}) + (\boldsymbol{m} \cdot \boldsymbol{c}) \times (\boldsymbol{\omega} \times \\ &(\boldsymbol{\omega} \times \boldsymbol{c})) \end{aligned}$$

Based on the product rule, the two representations of the above derivation $\boldsymbol{c} \times (\boldsymbol{m}(\ddot{\boldsymbol{p}} - \boldsymbol{g}))$ gives the same result as $(\boldsymbol{g} - \ddot{\boldsymbol{p}}) \times (\boldsymbol{m} \cdot \boldsymbol{c})$. In the above equation (3.47), inertia tensor is expressed about the center of mass point. Since, the center of mass vector is unknown in the recursive Newton-Euler formulation, it is important to shift the point from the center of mass $\boldsymbol{I_c}$ to the joint origin $_{\boldsymbol{J}}\boldsymbol{I_c}$. The axes of the coordinate frames that are attached to the link and the COM are parallel, it is possible to compute the inertia tensor with respect to the link's coordinate frame which can also be considered as the joint origin. The vector c in the equation (3.48) represents the distance between the point at the COM and the joint origin.

$$_{\boldsymbol{J}}\boldsymbol{I_c} = \boldsymbol{I_c} + \boldsymbol{m}[(\boldsymbol{c^T c})\boldsymbol{1_{3\times 3}} - (\boldsymbol{cc^T})] \tag{3.48}$$

The inertia tensor representation with respect to the link coordinate frame is represented in the Fig. 3.5. $\boldsymbol{I_c}$ represents the inertia tensor about the COM, $\boldsymbol{I_J}$ represents the inertia tensor about the joint origin.

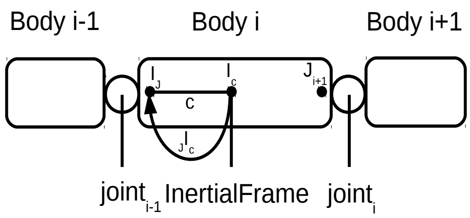

Figure 3.5: Parallel axis theorem

An et al. [1] uses the following simplification in order to keep the computation simpler in the Newton-Euler formulation. The total angular momentum of the rigid body can be written in a matrix form as follows

$$\boldsymbol{I} \cdot \boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \omega_x & \mathbf{0} & \omega_y & \omega_z & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \omega_x & \mathbf{0} & \omega_y & \omega_z \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I_{xx}} \\ \boldsymbol{I_{xy}} \\ \boldsymbol{I_{xz}} \\ \boldsymbol{I_{yy}} \\ \boldsymbol{I_{yz}} \\ \boldsymbol{I_{zz}} \end{bmatrix} \tag{3.49}$$

The following context proves that the above simplification holds true. The moment of inertia can be defined as

$$\boldsymbol{I} = \begin{bmatrix} \boldsymbol{I_{xx}} & \boldsymbol{I_{xy}} & \boldsymbol{I_{xz}} \\ \boldsymbol{I_{yx}} & \boldsymbol{I_{yy}} & \boldsymbol{I_{yz}} \\ \boldsymbol{I_{zx}} & \boldsymbol{I_{zy}} & \boldsymbol{I_{zz}} \end{bmatrix} \tag{3.50}$$

the inertia tensor is a 3×3 matrix describes the distributed mass of a rigid body that affects the relation between the angular velocity $\boldsymbol{\omega}$ and the angular momentum of the rigid body as described in the equation (3.37). The moment of inertia tensor can be described in the principal axis with the form

$$\boldsymbol{I} = \begin{bmatrix} \boldsymbol{I_{xx}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I_{yy}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I_{zz}} \end{bmatrix} \tag{3.51}$$

and the angular velocity vector is represented as

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{\omega_x} \\ \boldsymbol{\omega_y} \\ \boldsymbol{\omega_z} \end{bmatrix} \tag{3.52}$$

The product of the moment of inertia (3.50) and the angular part of the velocity vector (3.52) yields the form

$$\boldsymbol{I}\cdot\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{I_{xx}\cdot\omega_x} & \boldsymbol{I_{xy}\cdot\omega_y} & \boldsymbol{I_{xz}\cdot\omega_z} \\ \boldsymbol{I_{yx}\cdot\omega_x} & \boldsymbol{I_{yy}\cdot\omega_y} & \boldsymbol{I_{yz}\cdot\omega_z} \\ \boldsymbol{I_{zx}\cdot\omega_x} & \boldsymbol{I_{zy}\cdot\omega_y} & \boldsymbol{I_{zz}\cdot\omega_z} \end{bmatrix} \tag{3.53}$$

So, the equation (3.53) which is the total angular momentum of the rigid body with respect to the origin is proven equal to the form which is given in the equation (3.49)

$$= \begin{bmatrix} \boldsymbol{I_{xx}\cdot\omega_x} & \boldsymbol{I_{xy}\cdot\omega_y} & \boldsymbol{I_{xz}\cdot\omega_z} \\ \boldsymbol{I_{xy}\cdot\omega_x} & \boldsymbol{I_{yy}\cdot\omega_y} & \boldsymbol{I_{yz}\cdot\omega_z} \\ \boldsymbol{I_{xz}\cdot\omega_x} & \boldsymbol{I_{zy}\cdot\omega_y} & \boldsymbol{I_{zz}\cdot\omega_z} \end{bmatrix} \tag{3.54}$$

The xy products of inertia $\mathrm{I}_{\boldsymbol{xy}}$ and yx product of inertia $\mathrm{I}_{\boldsymbol{yx}}$ is equivalent since the moment of inertia tensor is a symmetric matrix. Hence the simplification of the equation is valid and proven. The equation (3.49) is equivalent to

$$\boldsymbol{I}\cdot\boldsymbol{\omega} = [\cdot\boldsymbol{\omega}] \begin{bmatrix} \boldsymbol{I_{xx}} \\ \boldsymbol{I_{xy}} \\ \boldsymbol{I_{xz}} \\ \boldsymbol{I_{yy}} \\ \boldsymbol{I_{yz}} \\ \boldsymbol{I_{zz}} \end{bmatrix} \tag{3.55}$$

In order to do the estimation of the link parameters, all the wrenches are summed up together in a single matrix in order to provide the relative references between the links. The wrenches between the links are related to the body fixed frame which is the point where the world coordinate frame is represented with the following formulation

$$\begin{bmatrix} \boldsymbol{F}_{i,i+1} \\ \boldsymbol{\tau}_{i,i+1} \end{bmatrix} = \begin{bmatrix} {}^{0}\boldsymbol{R}_i^{\boldsymbol{T}} & \boldsymbol{0} \\ [\boldsymbol{t}\times] \cdot {}^{0}\boldsymbol{R}_i^{\boldsymbol{T}} & {}^{0}\boldsymbol{R}_i^{\boldsymbol{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_{i+1,i+1} \\ \boldsymbol{\tau}_{i+1,i+1} \end{bmatrix} \tag{3.56}$$

where R represents the rotation matrix rotating the link i coordinate system to the world coordinate frame 0. Whereas the transpose is taken in this work since the force transforms between the coordinate frames generally happen in the backward recursion. In contrast it is a forward recursion in the case of motion transforms as discussed in the previous sections. The spacial force vector convention based on the article [7] is given as

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{f} \end{bmatrix} \tag{3.57}$$

where angular-before-linear representation is used in the transformations between the forces and motion vectors. f and n represents the linear and angular force vectors respectively. The formula for a force vector from the distal to proximal links can be computed based on an article [7] in the convention specified in the equation (3.57) as follows

$${}^{\boldsymbol{A}}\boldsymbol{X}_{\boldsymbol{B}}^{*} = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{T}} & [\boldsymbol{t}\times]\boldsymbol{R}^{\boldsymbol{T}} \\ \boldsymbol{0} & \boldsymbol{R}^{\boldsymbol{T}} \end{bmatrix} \tag{3.58}$$

R represents the orientation between the predecessor link's coordinate frame with respect to the successor link's coordinate frame. If the spatial force vector of the successor link is getting multiplied with the transform yields the spatial force vector of the current link and it can be symbolically given as

$${}^{\boldsymbol{A}}\boldsymbol{X}_{\boldsymbol{B}}^{*} * \boldsymbol{F} = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{n} + [\boldsymbol{t}\times]\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{f} \\ \boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{f} \end{bmatrix} \tag{3.59}$$

The linear-before-angular representation is used as it is discussed in the motion transforms can be specified in the following form.

$$\boldsymbol{F'} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{n} \end{bmatrix} \tag{3.60}$$

The equivalent transform for the spatial vector specified in the equation (3.60) is given in the following equation where the mapping occurs from the distal to proximal links.

$${}^{\boldsymbol{A}}\boldsymbol{X}_{\boldsymbol{B}}^{*} = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{T}} & \boldsymbol{0} \\ [\boldsymbol{t}\times]\boldsymbol{R}^{\boldsymbol{T}} & \boldsymbol{R}^{\boldsymbol{T}} \end{bmatrix} \tag{3.61}$$

If the acceleration (3.61)is getting multiplied with the spatial force vector (3.60) defined in this project yields the following result in a symbolic form.

$$
{}^{A}\boldsymbol{X}_{B}^{*} * \boldsymbol{F}' = \begin{bmatrix} \boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{f} \\ \boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{n} + [\boldsymbol{t}\times]\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{f} \end{bmatrix} \tag{3.62}
$$

By equating (3.59) (3.62), it is learned that the conventions of the spatial acceleration vector is different and the force vector is transformed based on the convention used in this work. The wrenches are combined together in a single matrix form which has estimation possibility for all the links of the manipulator and it can be represented as follows

$$
\begin{bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \boldsymbol{w_3} \\ \boldsymbol{w_4} \\ \boldsymbol{w_5} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_1} & \boldsymbol{T_{w}}_2^1\boldsymbol{A_2} & \boldsymbol{T_{w}}_3^1\boldsymbol{A_3} & \boldsymbol{T_{w}}_4^1\boldsymbol{A_4} & \boldsymbol{T_{w}}_5^1\boldsymbol{A_5} \\ \boldsymbol{0} & \boldsymbol{A_2} & \boldsymbol{T_{w}}_3^2\boldsymbol{A_3} & \boldsymbol{T_{w}}_4^2\boldsymbol{A_4} & \boldsymbol{T_{w}}_5^2\boldsymbol{A_5} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A_3} & \boldsymbol{T_{w}}_4^3\boldsymbol{A_4} & \boldsymbol{T_{w}}_5^3\boldsymbol{A_5} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A_4} & \boldsymbol{T_{w}}_5^4\boldsymbol{A_5} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A_5} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi_1} \\ \boldsymbol{\Phi_2} \\ \boldsymbol{\Phi_3} \\ \boldsymbol{\Phi_4} \\ \boldsymbol{\Phi_5} \end{bmatrix} \tag{3.63}
$$

where $\boldsymbol{w_i}$ represents the integral sum of all the cells in the ith row from the $\boldsymbol{K}$ matrix that represents a particular link of the manipulator. A represents the acceleration matrix which is represented in the equation (3.44). $\boldsymbol{T_w}$ represents the wrench transmission matrix which maps the wrenches between the links of the manipulator in the backward recursion as given in the equation (3.56). The above equation (3.63) can be re-written in a simple form as follows

$$
\boldsymbol{w} = \boldsymbol{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \cdot \boldsymbol{\Phi} \tag{3.64}
$$

where w represents all the wrenches that are mentioned in a single matrix form. K represents the kinematic matrix that takes the joint position, velocity and acceleration, each cell in the K matrix can be represented as $\boldsymbol{U_{ij}}$. The force-torque sensor is not available in the joints of the manipulator hence the torque can be measured only about the rotation axis of the joint. So, it is important to project the wrench into the joint's internal axis. The projection to the joint axis can be done by multiplying the subspace matrix S with the wrench

$$
\boldsymbol{\tau} = \boldsymbol{S} \cdot \boldsymbol{w} \tag{3.65}
$$

where, $\boldsymbol{\tau}$ represents the joint torque due to link, S [0,0,0,0,0,1] maps the wrench into torque since the forces can't be measured in the joint level and the projection vector depends on the axis of rotation of the individual joints. Each element of the K matrix will also be projected based on the axis of rotation of the joints with the use of the same subspace matrix S by

$$
\boldsymbol{K_{ij}} = \boldsymbol{S} \cdot \boldsymbol{U_{ij}} \tag{3.66}
$$

So the equation (3.64) can be re-written after the projection into the joint axis as

$$\boldsymbol{\tau} = \boldsymbol{K} \cdot \boldsymbol{\psi} \tag{3.67}$$

The technique used in this project to avoid the rank deficient problem is ridge regression which is explained briefly in the next section, the other possible method is closed-form dynamics equation and it is an complex procedure since the method generates more complexity in estimating inertial parameters in linear combinations. The above equation is linear in the unknown parameters [23]. K matrix which is a kinematic matrix depends only on the joint information i.e. the motion data, $\boldsymbol{\psi}$ is a matrix that comprises the ten unknown dynamic model parameters, $\boldsymbol{\tau}$ represents the torque that has been projected in the Cartesian space. The above equation will be changed for the model fitting by stacking up the data of all the trajectories vertically in a single matrix.

$$\boldsymbol{K} = [\boldsymbol{K_1} \cdots \boldsymbol{K_N}]^{\boldsymbol{T}} \tag{3.68}$$

and the same should be done for the torques

$$\boldsymbol{\tau} = [\boldsymbol{\tau_1} \cdots \boldsymbol{\tau_N}]^{\boldsymbol{T}} \tag{3.69}$$

N represents the number of trajectories or the total number of samples that are used in model fitting. The above equations (3.69), (3.68) will be used in the identification procedure.

3.5 Identification of the dynamic robot model parameters

3.5.1 Model fitting

Ridge regression [1] is used for better analysis of the multi regression data when there are many number of variables and the high collinearity. Linear least squares have a greater variance in the observations whereas the ridge regression produces the stable prediction and the variance is less. A set of linear equations can be written in a matrix form as

$$\boldsymbol{AX} = \boldsymbol{B} \tag{3.70}$$

The above equation (3.70) can be rewritten to create a form for the ordinary linear least square method

$$\begin{aligned} \boldsymbol{A^T AX} &= \boldsymbol{A^T B} \\ \boldsymbol{X} &= \boldsymbol{(A^T A)^{-1} A^T B} \end{aligned} \tag{3.71}$$

where X is a vector with unknown variables and the matrix multiplication of A(K-matrix) and X($\boldsymbol{\psi}$) gives the solution vector B($\boldsymbol{\psi}$). The equation (3.70) can be re-written as follows

$$\begin{bmatrix} \boldsymbol{K_{1,1}} & . & . & . & \boldsymbol{K1,n} \\ . & . & & . & . \\ . & & . & & . \\ . & . & & . & . \\ \boldsymbol{K_{m,1}} & . & . & . & \boldsymbol{K_{m,n}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi_1} \\ . \\ . \\ . \\ \boldsymbol{\psi_n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau_1} \\ . \\ . \\ . \\ \boldsymbol{\tau_m} \end{bmatrix} \tag{3.72}$$

where m represents the number of links $\boldsymbol{\psi}$ is the matrix of unknown dynamic model parameters which is 10m×1 matrix, $\boldsymbol{\tau}$ vector is known which is m×1 matrix and the kinematic matrix is in the size of m×10n matrix which is computed with the known joint informations($\boldsymbol{q}, \boldsymbol{\dot{q}}, \boldsymbol{\ddot{q}}$). To compute $\boldsymbol{\psi}$ matrix, K matrix has to be inverted, due to this inversion the matrix suffers from the rank deficiency problem. Hence the ridge regression method is used in this project to estimate the unknowns

$$\boldsymbol{\psi_{estimates}} = \boldsymbol{(K^T K + \lambda_{min} I_{10n})^{-1} K^T \tau} \tag{3.73}$$

where n represents number of links in the manipulator, $\boldsymbol{\lambda_{min}}$ is the smallest non-zero eigenvalue of the K$^{\boldsymbol{T}}$K matrix to avoid the singular matrix in the expectations. The inverse of the K$^{\boldsymbol{T}}$K results in the singular matrix which has been resolved by adding the smallest value to the diagonals of the K$^{\boldsymbol{T}}$K matrix. This resolves the rank deficient problem when the inverse of the K is computed.

3.5.2 Augmentation of the trajectory parameterization and optimization

The augmentation in this section is discussed based on the technique used in [23]. In [18], the random and valid trajectories were identified in the joint space to measure the joint angle, velocity and the acceleration from the manipulator joints. These trajectories were generated using the Moveit interface [21] where the data were not periodic but those were identified within the joint limits. The periodic and bandlimited data together with the differentiation in the frequency domain would improve the identification of the dynamic model parameter considerably. The following method generates periodic joint positions based on the frequency-domain for an individual joint as follows

$$\boldsymbol{q(t) = q_0 + \sum_{n=0}^{N} (a_n * sin(n * \omega_f * t)) + (b_n * cos(n * \omega_f * t))} \tag{3.74}$$

where the sampling frequency $\boldsymbol{\omega_f}$ is same for all the joints and it is given by the user, $q_{\mathbf{0}}$ represents the initial joint position, N represents the number of bins in the frequency space, t represents time that depends on the clock speed of the controller that is used to run the loop in order to produce the periodic joint data, a and b represent the coefficients of the trajectory parameterization. The coefficients selects the appropriate trajectory parameters that are identified using the d-optimality criterion. $\boldsymbol{\omega_f}$ represents the frequency of the robot workspace that needs to be covered in the specified time. The frequency of the signal for the number of bins in the frequency domain can be seen as $[\boldsymbol{[1 * \omega_f] \cdots [N * \omega_f]}]$. The equation (3.74) computes the periodic joint information(q(0)$\cdots$q(T)) by providing the initial joint angle then the time limit can be set to the loop in order to generate the periodic joint angle till the termination criterion is set. The similar technique is applied for all the joints of the manipulator in order to execute a particular trajectory.

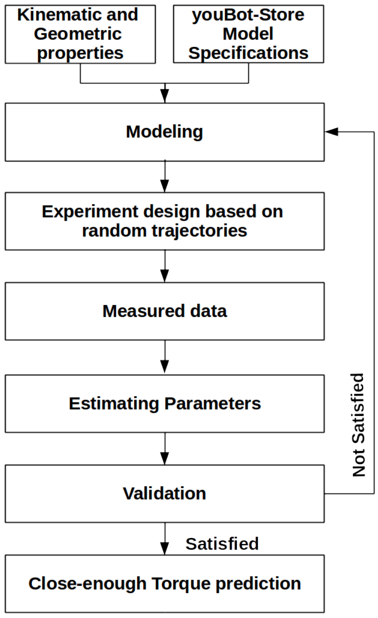

Figure 3.6: Without having trajectory optimization techniques.

The Fig 3.4 explains the method used to obtain the trajectories for the estimation procedure which is not periodic. The validation procedure of the above diagram is explained in the following section. This is the method used in this project to estimate the dynamic robot model parameters. To improve the efficiency of the estimation procedure, it is important to parameterize and optimize the trajectories the is used for the estimation as explained in the Fig 3.5.

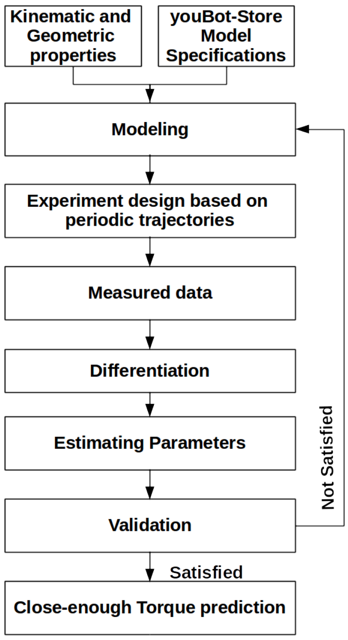

Figure 3.7: trajectory parameterization and optimization techniques.

The direct differentiation of the measured joint position based on time produces high frequency noises in the result. Hence, the digital signal processing concepts can be used to differentiate the measured joint position in the frequency domain. The measured joint position q(t) is obtained from the manipulator and then the digital signal processing is applied in order to eliminate the high frequency noises from the measured data. The Discrete Fourier Transform is applied to convert the time domain signal to frequency domain in order to eliminate the high frequency noises from the measured joint positions. This is achieved by applying the filter(low-pass filter) which filters the measured joint positions in frequency spectrum. The differentiation of the filtered joint positions are multiplied with $j2\boldsymbol{\pi}f_{\boldsymbol{s}}/P$ which is the continuous time frequency domain representation of the differentiator [23]. Eventually the Inverse Discrete Fourier Transformation is applied to get the joint velocity that is free of noise.

3.6 Static friction estimation and compensation

In this work, the static friction is measured from all the joints independently. The static friction is a kind of force that restricts the motion of a rigid body. It is not possible to overcome the static friction if the applied force is less than or equal to the friction force hence the rigid body stays at rest or in an equilibrium state. When the applied force is sufficient enough to overcome the static friction, then the object starts to move. The point at which the static friction of the rigid body has been overcome is called limiting point or break-away point. Once the body is in motion, there is a different kind of friction that acts on the rigid body which is called kinetic/dynamic friction since the effects are the result of the movement of all the links that are coupled together through the joints. In order to model and compensate the static friction in the youBot manipulator, the break-away friction has to be identified from the joints of the manipulator through experiments. The break-away friction is identified by eliminating the gravity force acting on the joints. In order to avoid the gravity force acting on the joints, it is mandatory to align the rotation axis of the manipulator joints with the direction of the gravitational acceleration. Henceforth the gravity effects are eliminated in the joints in order to identify the break-away friction for the individual joints of the manipulator. The important reason to compensate the gravity effect is to measure the friction term alone based on the velocity of each and every individual joints of the manipulator. Once the break away friction values are identified for all the individual joints, the frictional torque can be compensated through feed-forward/observer based friction compensation techniques.

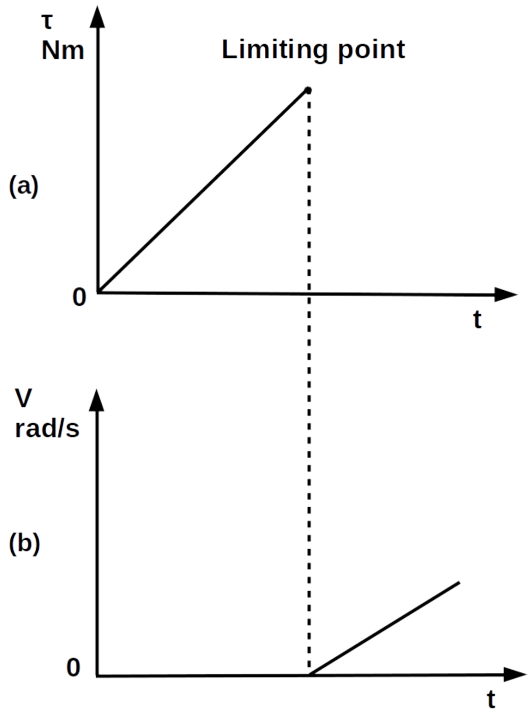

Figure 3.8: Static friction estimation procedure. $\boldsymbol{\tau}$ represents the torque applied overtime, V represents the velocity of the body. (a) represents the torque that is being applied to the joints over time. (b) monitoring the velocity over time

The above diagram describes the logic that has been implemented in this work to measure the break-away friction from all the joints of the manipulator. The sub-figure (a) in Fig. 3.8 shows that the torque has been gradually increased on each and every time step and the sub figure (b) explains that the point at which the applied torque is greater than the frictional force, the velocity of the body increases and it means that the rigid body is in motion. The threshold value (0.13 rad/s) has been used to check whether the rigid body starts to move or not. This is particularly useful in the automated experiments and this procedure is conducted for the number of experiments. It is not possible to decide the break-away friction torque on a particular joint based on only one experiment since the encoder measurements are not accurate. Eventually, the median value from the box-plot is identified from the data that has been collected through the experiments. The same procedure can be repeated to all the joints of the manipulator.

4 Experimentation

4.1 Estimation of dynamic robot model parameters

This section explains the experimental results of the identification/estimation of the dynamic robot model parameters.

Operating system - Ubuntu 14.04 LTS 64-bit.
Processor - Intel core I5-7200.

The Fig. 4.1 explains the procedure through which the results of the parameter estimation can be verified. The inputs to the estimation procedure can be given in two different ways. The first way involves the joint motion that is applied to the real robot's manipulator and the measured joint position **q**, velocity **qd** and acceleration **qdd** and the resulting torque $\boldsymbol{\tau}$ will be given as an input to the estimation procedure. Once the parameter estimation is yielded, it is not possible to verify the plausibility by mapping back to the parameters that have generated the measured joint data, torque. This is due to the reason that the joint data and torque have been generated by the real robot, so it is not possible to map the estimated parameters back to the dynamic parameters. The second method involves the Simbody visualizer that can verify the parameter estimation of the links where the joint motion, torque data are generated based on the model parameters given by the manufacturers. In this work, the construction of the youBot manipulator in the Simbody visualizer is not yet complete. After the manipulator is built on the visualizer, it is possible to verify the dynamic model parameters and also the forward dynamics can be verified.

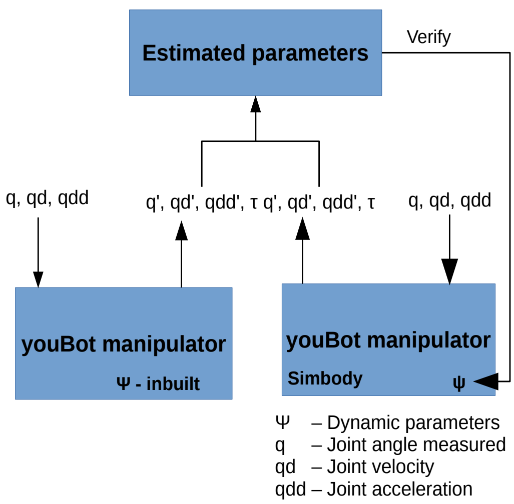

Figure 4.1: Experimentation overview

The manual plausibility check is not succeeded in this project, so the application of the verification method given in the Fig. 4.1 is not used in this work. The estimation results are given in the table 4.1 and the plausibility check which is done manually is not successful due to the reason that the mass value for link 3 and link 4 is not a positive number.

Table 4.1: Estimation of the link parameters

Parameters	**Link1**	**Link2**	**Link3**	**Link4**	**Link5**
m(kg)	0.0	0.4423282	-0.07506977	-0.18771262	0.0409276
m*c_x(kg.m)	0.0	0.02806447	-0.02850257	-0.00062882	-0.00758307
m*c_y(kg.m)	0.0	-0.00239462	0.01631021	-0.08223756	0.04153505
m*c_z(kg.m)	0.0	7.01280635	-0.40840137	-0.35556988	-0.06903857
Ixx(kg.m^2)	0.0	-0.89356512	0.24514397	-0.03275741	0.05140932
Ixy(kg.m^2)	0.0	0.32704962	0.0536501	-0.10738527	0.13045645
Ixz(kg.m^2)	0.0	0.54061164	0.29247126	0.21439377	0.03270133
Iyy(kg.m^2)	0.0	0.90142364	-0.36549833	-0.12854182	-0.00264338
Iyz(kg.m^2)	0.0	-0.02933403	-0.01265898	-0.09367053	0.00154423
Izz(kg.m^2)	-0.2390516	0.37695014	0.04190707	-0.45573522	0.11160058

It is not possible to find the model parameter near the base of the manipulator as these links are fixed i.e. link 0 and the only parameter $\boldsymbol{I_{zz}}$ for the link 1 can be estimated and this can be seen in the table 4.1. The implementation details need debugging and an update in order to identify the parameters properly.

4.2 Static friction estimation and compensation

This section discusses and presents the result of the static friction estimation. The PI controller is used in this work to keep the manipulator joints in the candle configuration. Since it is an automated testing with more than one experiment to find the average point at which the break-away friction happens, the joint has to move back to it's actual position which is the candle configuration. It introduces huge changes in the application of the position set point on the manipulator joints result in overshooting on the manipulator's step response as it is shown in Fig. 4.2. This problem is called controller windup which is introduced due to the integral term that accumulates the errors from the start till the step response is complete.

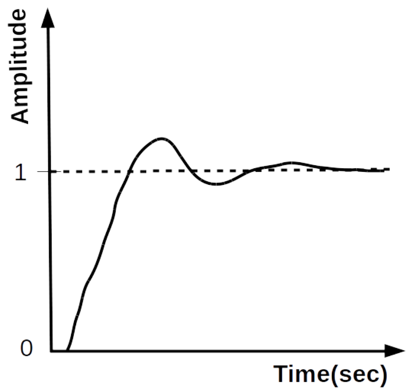

Figure 4.2: Step response

In this work, a PI controller [16] is used to handle the fast rise times in the step response and the fine tuning of the controller gains are achieved with an optimal values since high gains in the proportional control introduces instability and overshoot problems. A PI controller is a control loop which is facilitated with the feedback of the system's output calculates an error signal e(t) by computing the difference between the measured output from the system and the position setpoint(input) that is the desired set point of the system. The PI controller is used in this project to avoid the overshooting and to achieve the stability in control which is highly depending on the measured data. The simple PI control can be represented pictorially as given in the following figure

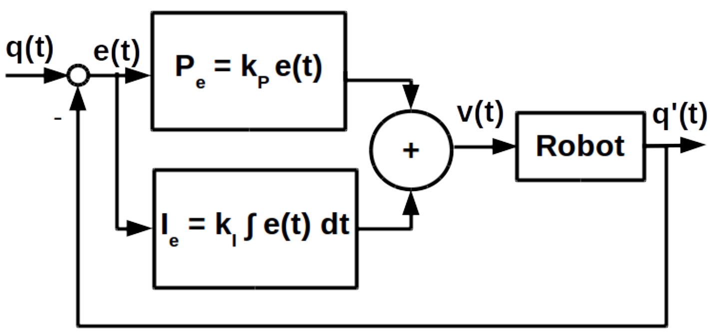

Figure 4.3: PI controller

The simple PI control operation can be mathematically represented as follows

$$\boldsymbol{v(t) = (k_P \cdot P_e) + (k_I \cdot \sum_{t=0}^{T} P_e(t))} \tag{4.1}$$

where the first part in the addition operation of the above equation represents the P control of the PI control and the second part is the integral term I of the PI control. v(t) represents the velocity that is applied to the manipulator joint's in every cycle, $\boldsymbol{k_P}$ and $\boldsymbol{k_I}$ represents the position, integral gains respectively. The variable T represents the total time taken by the controller to complete the execution of the set point. The position error can be computed as follows

$$\boldsymbol{P_e = (q - q')} \tag{4.2}$$

where q, q' represents the given and observed position setpoints respectively. The integral error term can be computed by adding up all the position errors over time which is nothing but the sum of all the previous errors that has been encountered in the step response and it can be represented in a mathematical form as given in the following equation

$$\boldsymbol{I_e = \sum_{t=0}^{T} P_e(t)} \tag{4.3}$$

where $\mathrm{I}_{\boldsymbol{e}}$ represents the integral error that accumulates the position error over the total time of T.

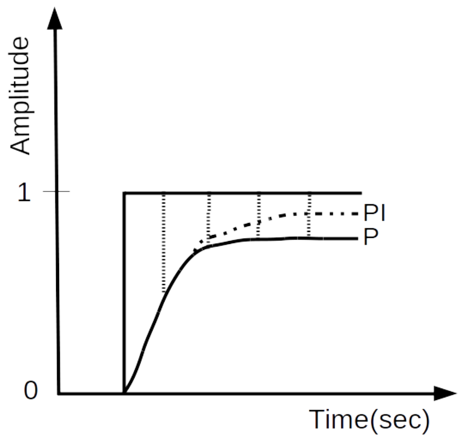

Figure 4.4: Integral part of the PI controller

The above Fig. 4.4 explains the error reduction mechanism when using only the P controller and then the PI controller. The solid curve in the Fig. 4.4 represents the error reduction with the use of P controller alone. Whereas, the dotted curve represents the PI controller which has the integral term that accumulates the position errors overtime and the overall error is reduced comparing the P controller.

4.2.1 Break-away friction measured in the youBot base joints

The break-away friction torque is estimated in wheel joints 3, 4 of the youBot base and the result of this testing is plotted in figure 4.2. There are 25 experiments conducted in two different wheels and the torque data is recorded in order to find the break-away torque in the respective joints. The testing on the youBot base is initiated without the PI controller in the application level, hence the overshooting problem occurs in the execution. Due to the sudden change in the position set point in the controller, there are no movements observed in the joints. In order to achieve the optimal control, a PI controller is used for testing purposes. The position gain is set to 1 and the integral is set to a very minimal value 0.001 since the error in the starting part of the response has a higher deviation which might make the controller overshoot the step response.

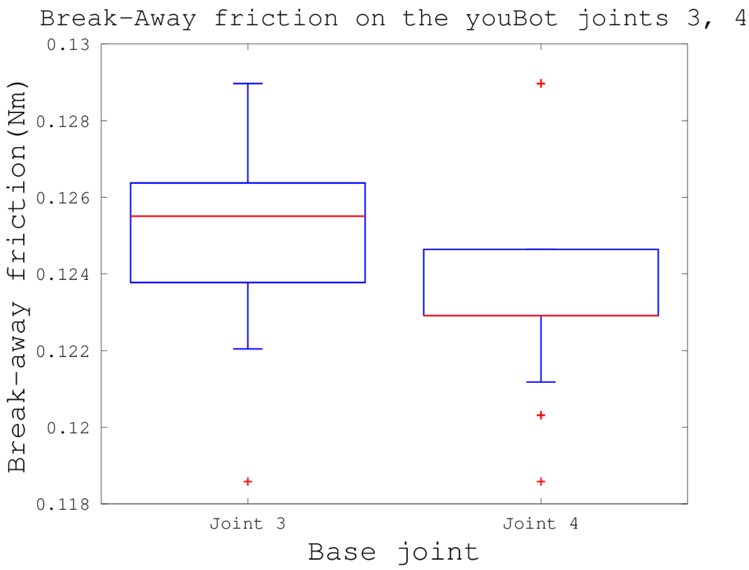

Figure 4.5: Break away friction on joints 3 and 4 in the youBot base

The median value of the joint 3 is approximately 0.126 Nm and the joint 4 resulted in 0.123 Nm based on the box plot. The pictorial representation is shown in the Fig. 4.5.

4.2.2 Break-away friction in the youBot manipulator joints

The break-away friction on the youBot manipulator has been experimented in two different ways since there are fixed joints present in the manipulator has no influence due to gravity. The joints 2, 3 and 4 have a strong influence due to the gravitational force hence the estimation of break-away friction alone is quite impossible. So, the manipulator has been attached to a table and it has been kept in a configuration where there are no gravity forces acting on joints 2, 3 and

4. The table has been positioned in the ground approximately at 90 degrees since the encoder is not used in this setup and the complete setup is shown in Fig. 4.6.

Figure 4.6: Experiment setup for measuring the break-away friction on the manipulator joints in the top view from the front side

Table 4.2: Automated testing is the median result(medium quartile) of the box plot.

Joint No.	**Break-away friction(Nm) based on automated testing**
1	1.260
2	0.956
3	0.486
4	0.300
5	0.177

The fixed-joints such as joint number 1 and 5 has no influence due to gravity, so the manipulator has been set in the candle configuration in order to measure the break-away friction.

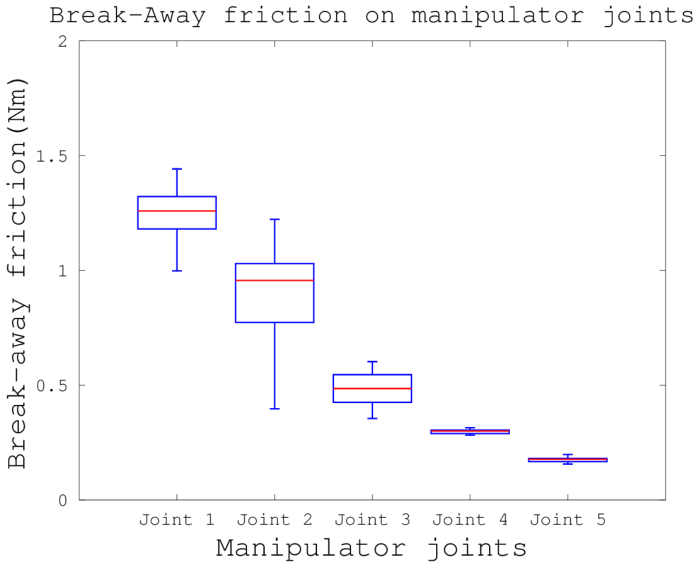

Figure 4.7: Break away friction on the manipulator joints. The red color line in the graph represents the median value that is observed from the list of experiments.

The identification of the break-away friction on all joints has been experimented manually to know the amount of force that is required to move the links that is attached with the joints. The manual identification of the break-away friction for the joints 1 to 5 are 1.000, 1.400, 0.700, 0.550 and 0.300 respectively. The torque values are started with 0 Nm and then slowly increased in order to avoid the damage that can occur due to excessive current to the joints. After measuring the torques manually, the approximate minimum and maximum torque limits of the joints are defined based on the measured torques. These limits are then used in the automated experiments and the setup is depicted in the Fig. 4.7. Twenty-five experiments are conducted in total in order to find the break-away friction of all the joints of the manipulator. The resulting torques observed from the automated experiments are box-plotted and the median value for all the joints are given in the table 4.2.

4.3 Evaluation

4.3.1 Validation of the identified dynamic robot model parameters

After the estimation of the dynamic model parameters, it is important to check the plausibility of the estimated parameters in two steps. The first step is to check the plausibility of the estimated parameters manually by comparing it with the youbot manipulator's specifications. An approximate mass threshold value of 0.1 kg is assumed based on the specifications provided by the manufacturers to verify the plausibility of the estimated mass for all the links. If the estimated mass value of the links going over the defined threshold rate the parameters are considered to be implausible. The second validation step involves a systematic way of verification which is explained in the Fig. 4.8.

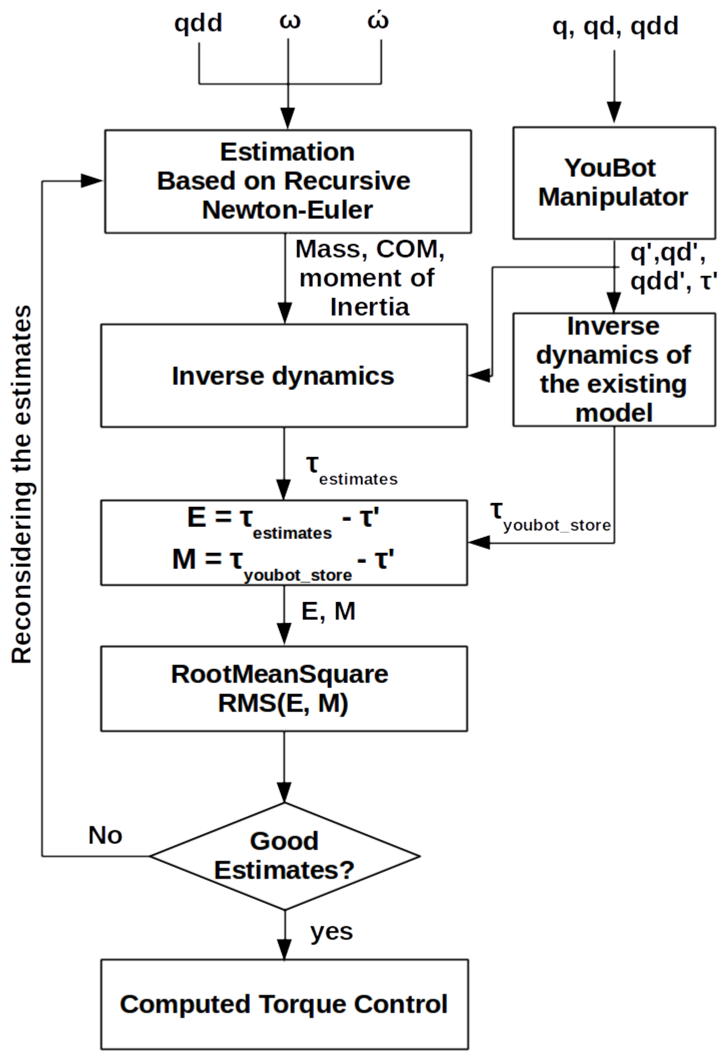

Figure 4.8: Validation of the estimated model parameters

where

- q, qd, qdd represents joint angle, velocity and acceleration respectively
- q', qd', qdd', $\boldsymbol{\tau}$' represents the measured joint information from the robot manipulator
- $\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}$ represents the body velocity and acceleration respectively which is computed based on the joint velocity, acceleration
- E represents the error between the torques generated from the estimated model and the actuator torques
- M represents the error between the torque generated from the youbot-store model and the actuator torque
- RMS means the root mean square method

The step by step procedure of the Fig. 4.8 is explained as follows

- As a result of the estimation procedure, it is seen that the mass, COM and moment of inertia parameters are obtained
- The estimated parameters are used in a model(e.g. URDF model) to compute the torques based on the given joint information($\boldsymbol{q', \dot{q}', \ddot{q}'}$).
  - At the same time, the state-of-the-art model(youBot-store) parameters are also used to generate the torques for the evaluation purposes
- Now, the next step computes E and M which is based on the torques generated by both the estimated and existing model with the actuator torques
- Then the RMS value will be identified for both E, M for comparison. This error can be plotted against time and the results are presented in the experimentation section.
- The final step is to check the estimates are good or bad by comparing the closeness of the torques generated by the torques to the actuator torques when comparing the same with the generated torques from the existing models. If the estimates are not good the estimation procedure has to be reconsidered to improve the accuracy in predicting torques

This validation procedure given in the Fig. 4.8 is not applied in this work, since the estimation results are not properly identified. The implementation issues need to be resolved in order to complete this validation procedure. The modular check has been done for the kinematic analysis and for the spatial force vector transforms and this mechanism seems to be proper based on the set of configuration checks on the manipulator.

5 Conclusions and Future work

This work presented the domain specific estimation procedure based on the modified recursive Newton-Euler formulation which depends on the joint motion and the torque data. The kinematic chain of the robot manipulator has been built and the forward kinematics for the position, velocity and acceleration has been implemented for the youBot manipulator. The dynamics of the robot has been studied extensively and the findings are briefed. The final results of the estimation procedure does not yield the plausible results. In order to check the correctness of the estimation procedure, the forward position, velocity and acceleration kinematics analysis have been checked for the set of configurations. The resulting kinematic analysis is matching with the mechanism that has been implemented in this work. One of the reasons that this algorithm suffers due to the fact that the velocity, acceleration are the derivatives of position, velocity respectively. The break-away friction is estimated from the youBot base, manipulator joints and these values can be used in the friction compensation techniques. The augmentation of the trajectory parameterization and optimization clearly states that the optimization is an important measure with the trajectories and to avoid the noises due to the direct differentiation of the joint position and velocity.

Future work

This section proposes the possible areas that can be improved in this field of research

- The estimation procedure has to be debugged in order to correct the implementation issues in the parameter estimation.
- The dynamic robot model parameters can be identified with n special test motions as it is described in this work. But the efficiency will be improved by fully covering the robot workspace in a periodic way which is helpful in the derivation of the velocity, acceleration for the better estimation of the system.
- An another important aspect that needs attention is to compensate the dynamic effects of the manipulator. Friction is one of those important effects that need to be addressed. To achieve this, kinetic friction needs to be completely modeled and compensated in the robot joints.
- After identifying the model parameters, the design of a control law for the computed torque-control that determine the joint torques with the use of the identified model. This technique is similar to the technique used in the article [23].

References

[1] Chae H An, Christopher G Atkeson, and John M Hollerbach. Estimation of inertial parameters of rigid body links of manipulators. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, pages 990–995. IEEE, 1985.

[2] Myo Thant Sin Aung, Ryo Kikuuwe, and Motoji Yamamoto. Friction compensation of geared actuators with high presliding stiffness. *Journal of Dynamic Systems, Measurement, and Control*, 137(1):011007, 2015.

[3] Herman Bruyninckx. Robot kinematics and dynamics. *Online at* `https://people.mech.kuleuven.be/~bruyninc/tmp/HermanBruyninckx-robotics.pdf`, 2010.

[4] Stephen La Vern Campbell, Jean-Philippe Chancelier, and Ramine Nikoukhah. *Modeling and simulation in Scilab/Scicos*. Springer, 2006.

[5] John J Craig. *Introduction to robotics: mechanics and control*, volume 3.

[6] Rhama Dwiputra, Alexey Zakharov, Roustiam Chakirov, and Erwin Prassler. Modelica model for the youbot manipulator. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, number 096, pages 1205–1212. Linköping University Electronic Press, 2014.

[7] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[8] L. Freidovich, A. Robertsson, A. Shiriaev, and R. Johansson. Lugre-model-based friction compensation. *IEEE Transactions on Control Systems Technology*, 18(1):194–200, Jan 2010.

[9] M. Gautier, A. Janor, and P. O. Vandanjon. Didim: A new method for the dynamic identification of robots from only torque data. In *2008 IEEE International Conference on Robotics and Automation*, pages 2122–2127, May 2008.

[10] Benjamin Keiser. Torque control of a kuka youbot arm. *Mémoire de maıtrise, Robotics and Perception Group, University of Zurich*, 2013.

[11] Pradeep Khosla. Estimation of robot dynamics parameters: Theory and application. 1987.

[12] YF Liu, J Li, ZM Zhang, XH Hu, and WJ Zhang. Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system. *Mechanical Sciences*, 6(1):15, 2015.

[13] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[14] H. Olsen and G. Bekey. Identification of robot dynamics. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1004–1010, Apr 1986.

[15] Martin M Olsen, Jan Swevers, and Walter Verdonck. Maximum likelihood identification of a dynamic robot model: Implementation issues. *The international Journal of robotics research*, 21(2):89–96, 2002.

[16] V. Parra-Vega, S. Arimoto, Yun-Hui Liu, G. Hirzinger, and P. Akella. Dynamic sliding pid control for tracking of robot manipulators: theory and experiments. *IEEE Transactions on Robotics and Automation*, 19(6):967–976, Dec 2003.

[17] M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands, Dordrecht, 1994.

[18] Jeyaprakash Rajagopal. Identification and evaluation of dynamic robot models. SS17 FH-BRS - Plöger, Schneider Supervising, April 2017.

[19] Michael A Sherman, Ajay Seth, and Scott L Delp. Simbody: multibody dynamics for biomedical research. *Procedia Iutam, Online at* `https://simtk.org/projects/simbody`, 2:241–261, 2011.

[20] Ruben Smits, H Bruyninckx, and E Aertbeliën. Kdl: Kinematics and dynamics library. *Avaliable:* `http://www.orocos.org/kdl`, 2011.

[21] Ioan A Sucan and Sachin Chitta. Moveit! *Online at* `http://moveit.ros.org`, 2013.

[22] J. Swevers, F. Al-Bender, C. G. Ganseman, and T. Projogo. An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on Automatic Control*, 45(4):675–686, Apr 2000.

[23] Jan Swevers, Walter Verdonck, and Joris De Schutter. Dynamic model identification for industrial robots. *IEEE Control Systems*, 27(5):58–71, 2007.

[24] KR Symon. Mechanics addison-wesley. *Reading, MA*, 1971.

[25] Guido Van Rossum and Fred L Drake. *Python library reference*. Centrum voor Wiskunde en Informatica, 1995.

[26] Kuka youBot developers. Dynamic robot model. *online at* `http://www.youbot-store.com/developers/kuka-youbot-kinematics-dynamics-and-3d-model-81`, Nov 2016.

[27] Yongliang Zhu and P. R. Pagilla. Static and dynamic friction compensation in trajectory tracking control of robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 2644–2649, 2002.