Автономная некоммерческая организация высшего образования
«Университет Иннополис»
(АНО ВО «Университет Иннополис»)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

по направлению подготовки
09.04.01 – «Информатика и вычислительная техника»

GRADUATION THESIS
(MASTER GRADUATE THESIS)

Field of Study
09.04.01 – «Computer Science»

Направленность (профиль) образовательной программы
«Робототехника»
Area of Specialization / Academic Program Title:
«Robotics»

Тема / Topic	**Податливое управление четырехногим роботом с использованием моторов переменной жесткости / Compliant control of quadruped robots with VSA**	
Работу выполнил / Thesis is executed by	**Джимох Хафиз Олафисайо / Jimoh Hafeez Olafisayo**	HAOJIMOH подпись / signature
Руководитель выпускной квалификационной работы / Graduation Thesis Supervisor	**Савин Сергей Игоревич / Savin Sergei Igorevich**	подпись / signature

Иннополис, Innopolis, 2021

Contents

List of Tables

List of Figures

Acknowledgement

In the name of Allah, the Most Beneficent, the Most Merciful;

I give thanks to Almighty Allah, Al-Aleem- The All-Knower, Al-Khabeer- The All-Aware who has dominion over all things and who has provided the knowledge to undergo this project. Indeed, He has created all in the most beautiful and perfect way without any mathematical equation, literature review or supervisor. This topic stems from understanding and taking inspiration from locomotion in biology and implementing or mimicking such structures in robots to make them walk without slipping, penetrating the ground or damaging their environments. In converse to what some scientists say that believing in existence of Allah and science cannot co-exist, my research work only proved to me that progress in science only comes from acknowledging the supremacy and perfection of Allah creation. Walking, jumping and standing without slipping is easy for humans and comes without much effort but difficult for robots to do autonomously even with current state of art technology.

Special thanks goes to my Supervisor, Sergei Savin for the mentorship and guidance throughout the course of this project. I so much appreciate how he made himself available to answer my questions and explain or point to resources that would help me. I also appreciate the entire robotics faculty of Innopolis University for the knowledge I have benefited from them.

I thank all my amazing parents, wonderful siblings and friends that have been available to provide emotional and psychological support throughout the course of this 2 year journey coming to an end.

Special shout-out goes to my beautiful wife, Safiyyah Idowu, a rare-gem I was blessed with and who our paths crossed towards the start of this journey.

Abstract

High-gain position controllers and Inverse dynamics controllers have proved over the years to be very efficient for control of fully actuated robots such as fixed base manipulators. However, implementing such techniques on floating base robots like bipeds and quadruped systems is non-trivial due to under-actuation, floating base dynamics of the system, highly non-linear dynamics, dynamically changing contact interactions, and closed loop kinematics.

A linear controller based on active and passive compliance using series elastic actuator (SEA) and variable stiffness actuator (VSA) was applied to the compliant control of quadruped robot by using a contact consistent iterative linear quadratic regulator (LQR) using a linearization of the full robot dynamics together with the contact constraints. This work further introduced an analytical formulation for the linearization of the SEA and VSA robot in terms of the linearized dynamics of the robot. The LQR controller gives advantage of taking into account the close loop kinematics and coupling effects between the joints to create an optimal feedback gain for the robot locomotion. Floating base inverse dynamics for feedforward control is achieved by using QR decomposition. During active compliance, we compare performance using computed torque controllers and LQR and evaluate cost of different tasks on several experiments with different initial positions. The results obtained shows that LQR control gives an optimal feedback gain with lower cost as compared to CTC. Also, LQR control based on passive compliance through elastic elements can track desired trajectory of the robot.

Chapter 1

Introduction

1.1 Motivation and Problem Description

Legged robots are an important branch of mobile robots. They are highly adaptable, flexible and have complex non-linear dynamics. Researchers have continued to study quadruped and bipeds robots over the past decades on how to manoeuvre its locomotion and make it have flexible and adaptive mobility and interact with its environment especially motion over very rough terrain. In particular, quadruped robots should be able to walk, trot and achieve other different motion types like jumping over an obstacle without slipping, damaging itself or its environment. Instead of robots trying to avoid contacts with the environment, we can use the environment to maintain balance, safely move and interact with environment. This would eventually pave way and allow robots to be deployed for use efficiently outside the laboratory.

Maintaining balance while in contact with environment is however difficult in legged robots as against fixed-base robots like industrial manipulators. This is because of the distinctive constraint in legged robots where the only means for it to control its momentum is to make and break contact with the environment which often occur between the robot's feet and the floor. To control momentum, a robot must be in contact with something, else it would be said to be in free fall. In some other more complex cases, there is contact between robot hands or body and the world like walls, doors and even other robots. Control of contact interactions is a major challenge in control of legged robots because of no prior knowledge of contact force, challenge of precise estimation of contact forces and changing contact force.

In summary, maintaining balance and ability to move over uneven terrain is challenging because of:

- Floating base dynamics of legged-robots
- Close loop kinematics chain between ground and trunk
- Highly non-linear dynamics of the robot
- Unknown contact force between the ground and robot feet

and these form the bane of locomotion in legged-robots

1.2 Problem Statement

Compliant control of robots involves when the states of the robot is constrained by the task such that the robot do not penetrate the surface in contact with and that the force between robot and contact point is consistent with friction and the robot do not eventually slip. If a rigid robot collide with its environment or another rigid contact surface, the robot may slip. Compliant control of legged robots have been well studied using force control techniques and several other techniques using force sensors to determine the ground reaction force fed back into the control loop [1]–[3] Compliance can also be achieved by implementing springs or pneumatic elements at end-effectors of robot joints which was once considered parasitic most especially in industrial robot joints [4]. Springs naturally provide mechanical energy which can be stored and this can be released into the robotic system to provide energy, reducing motor work and enable dynamic behaviors [5]. Series elastic actuators have also been widely used in robot locomotion and have achieved high quality force control because of the low impedance and friction in it. Variable Stiffness Actuators(VSAs) which are extensions of SEA are only different in that the stiffness of the string changes with respect to time. VSAs offer a special mechanical advantage of ability to adapt the system to tasks due to variability in the stiffness. There are a number of different types of VSAs based on implementation of the elastic elements and arrangement which can either be linear-variable stiffness or nonlinear-variable stiffness [6]. In the antagonistic joint setup of VSAs biologically inspired from muscles of living creatures, there are two equally-sized actuators that transmit forces in two different into directions.

This work would explore and investigate compliant control of quadruped robot by extending traditional control techniques and experiment with introducing compliance in legged robot locomotion passively using spring elements by using Series Elastic Actuators and Variable stiffness Actuators.

Chapter 2

Literature Review

2.1 Quadruped Robot Control

Legged locomotion and control is a challenging task [7] due to the highly non-linear dynamics nature of the robot, unknown contact force with ground and the floating base dynamics of the quadruped robot. The legs of these robots form closed loop kinematic chains between the ground and the trunk, thus there is dynamic coupling between these legs adding to the complexity of the dynamics. PID controller, LQR and Feedback linearization were applied to the control of two link hydraulic HyQ legs [8] with excellent result. Feedback linearization achieves the fastest response in this work however this approach can not be scaled for full body dynamics as it doesn't take into account the effect of constraints and floating base of the quadruped robot. There are several control approaches that are being developed for the control of legged robots. Position control techniques which are well studied and applied to LittleDog [9], and SILO4 walking robots [2] are highly stiff because of the high PID gains which makes them have poor tracking performance. Legged robots require a controller that is less stiff with high tracking performance than can handle the peculiar characteristic of switching contact constraint forces that makes the robot to be either under-actuated, fully-actuated or over-actuated depending on the number of contact interactions. The three cases as explained in [10] can be described as:

- Fully-actuated: Legged robots like quadrupeds and bipeds are fully actuated if there are as much degree of freedom of constraint as degree of freedom of free floating base. When a quadruped is standing on two feet or a humanoid is standing on one foot
- Under-actuated: When the free floating degree of freedom is more than the constraints, the robot is under-actuated. An example of this case is when a quadruped is standing on one foot or is in the aerial phase of running.
- Over-actuated: This occurs when the number of constraints is such that the serial kinematic tree of the robot is effectively turned into a parallel one, and there exist several combination of control actions that would achieve the same result.

For inverse dynamics control that is independent of measured forces, the following assumptions is usually satisfied:

- There is no motion at contact point. If $\mathbf{x}_c$ is point of contact with a surface, then:

$$\dot{\mathbf{x}}_c = \mathbf{0}, \quad \ddot{\mathbf{x}}_c = \mathbf{0} \quad \text{must be maintained}$$

- The robotic system must not be over constrained. i.e every constraint must be sufficiently satisfied.

- The system must be constrained such that it would eliminate under-actuation. In other words, the constraint must have a rank equal to 6 to cancel the under-actuation of degree of 6 of the free floating base. Thus, the kinematic constraint even if it linearly dependent must still have rank of 6.

Based on the above assumptions, it can be seen that the minimum number of constraints required is 6. High gain position control is not beneficial since the controller would everything possible to satisfy the position goal with all available force possible to achieve it. This might be good for fixed-base robots but highly disadvantageous for robots with task involving contact with environment and the contacts are not only precisely known but also stochastic. High gain position control can lead to unbalancing the robot eventually leading to slip. Furthermore, because of the close kinematic chain of legged robot, it required a precise kinematic model, environment and zero velocity when making ground contact. If during contact, the feet has non-zero velocity, there would be a perturbation on the feet and which would be transmitted to other parts of the body through the close-loop kinematic chain. As a result, low gain position controllers are preferred above high-gain position control.

Passivity based control mechanisms were used to control full sized bipedal humanoid robots [11], [12]. The advantage of this approach is that it is model-less, does not require force measurement at contact points and can accommodate simultaneous control processes using force, velocity or position based control. The contact forces are however computed under quasi-static assumption which is the major limitation of this approach in dynamic motions. Other works [13], [14] have implemented a full inverse dynamic model of robot using torque control. This approach is well suited for robots in dynamic motions and are used to compute feed-forward torques of the desired joint trajectories but the need for precise dynamic model makes them a bit challenging to implement on real hardware.

An extension of model based inverse dynamics is the Hierarchical inverse dynamics allow to break down series of dynamic robot behaviours to sequence of tasks where there is enforcement of hierarchies at each task since simultaneous achievement of several tasks are difficult to achieve or multiple objective function between task are incompatible. A weighting strategy was applied to iCuB robot [15] to define priorities between tasks which allow smooth transition between tasks, constrained and non con-constrained states for complex robot behaviours. Pseudo-inverse techniques are used in [16], but the presence of inequality constraints in friction cones constraints is their limitation. A number of previous works [17]–[19], use QP to compute inverse dynamics over different constraints by planning trajectories in operational space and then converting to joint torques using QP. For example, under constraints like dynamic consistency, friction cones, joint tracking amongst others, QP have been used like in the DARPA robotics challenge final for inverse dynamics and converting trajectory planning conversion from operational space to joint torque [18]. A development from this work is Hierarchical inverse dynamics controllers based on cascades of quadratic programs (QP). This approach have been used as feedback-controllers in torque controlled humanoid robot [20].

While feed-forward torques can be computed with inverse dynamics, there is still need for a feedback control mechanism that takes into account the dynamic coupling, contact interactions and the floating base dynamics. Orthogonal decomposition of the constraint jacobian rigid body have been used for inverse dynamics control of floating base robots [9]. Orthogonal decomposition allows to find the inverse dynamics equation independent of the unknown contact force. This technique has been applied in a bipedal humanoid and robot dog walking over rough terrains in [21] and [9].

Contact consistent LQR termed constrained LQR (CLQR) controllers have been used for whole-body control of humanoid bipedal robots [9], [22]. CLQR embeds the constraints in the linearized dynamics of the robot by projecting the linearized dynamics of the robot in the null-space of the constraint jacobian. It has the advantage of allowing us to find an optimal feedback control policy for the actuated joints while taking into account the dynamic coupling between the joints. Also, even though our floating base robot is under-actuated with degree of 6, CLQR still finds an optimal gain matrix for achieving different joint trajectories.

2.2 Series Elastic Actuator

Robots with String Elastic Actuators have deformable part introduced through a spring with a fixed stiffness. The stiffness parameter is desired or controlled and allows the actuator to absorbs energy from external impacts and release energy at the right time. Fig.2.1 shows an SEA robot joint structure which is based on the work in [23]. For the SEA robot, the actuator dynamics can be written as follows:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_e = \boldsymbol{\tau}_{ext} \tag{2.1}$$

$$\mathbf{J}\ddot{\mathbf{q}}_s - \boldsymbol{\tau}_e(\psi) = \boldsymbol{\tau} \tag{2.2}$$

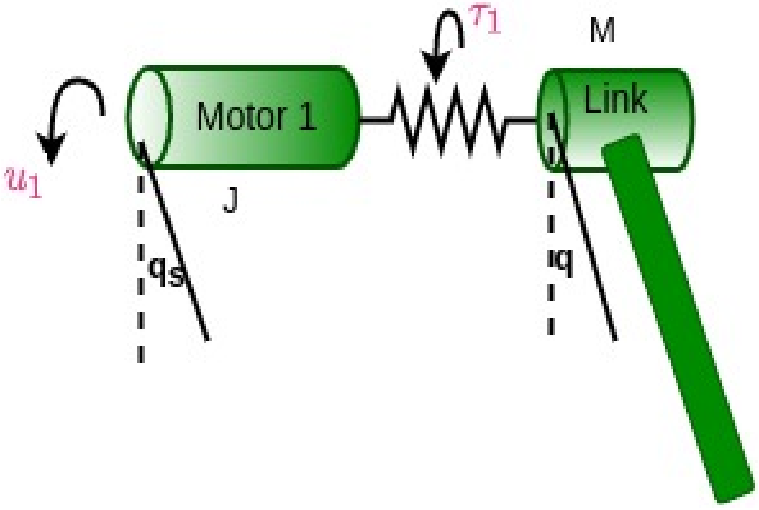

Figure 2.1: Series Elastic Actuator

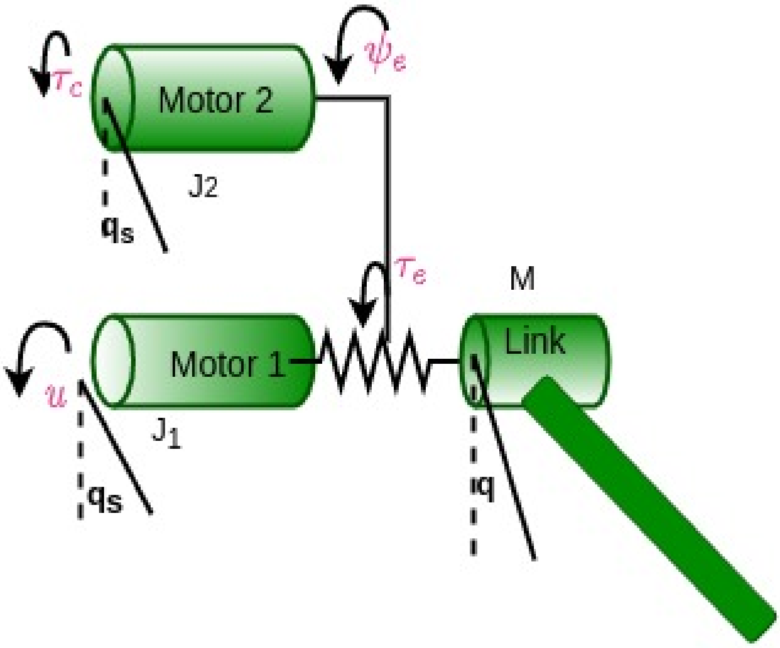

Figure 2.2: Variable Stiffness Actuator

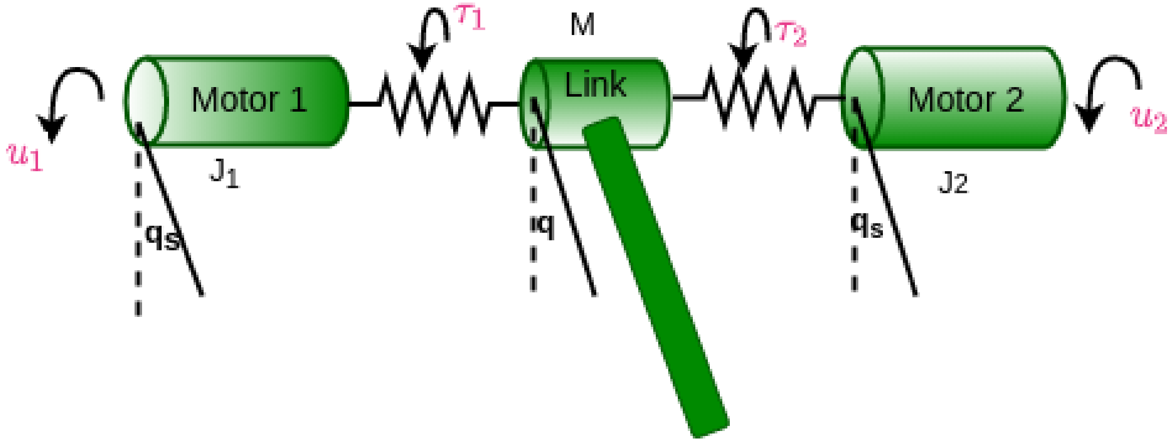

Figure 2.3: VSA- antagonistic

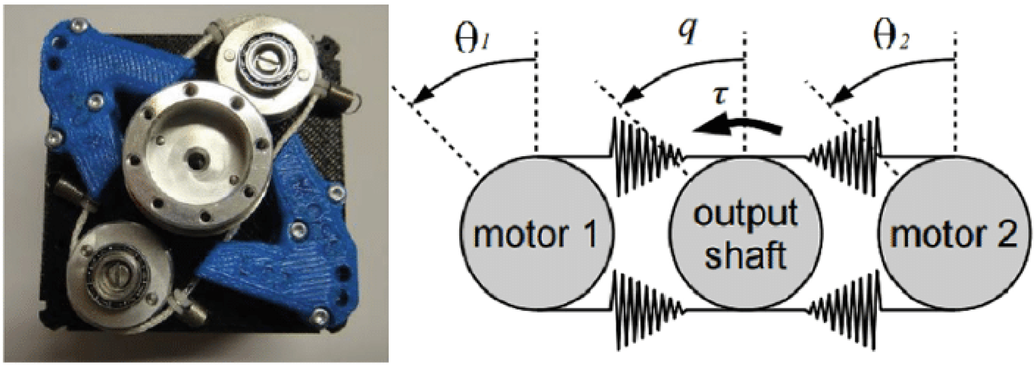

Figure 2.4: qbmove VSA

2.3 Variable Stiffness Actuator

In general, in addition to the SEA earlier discussed, there are two classes of VSA: the antagonistic VSA (Figure 2.2), and VSA with a serial spring (Figure 2.3). It consists of a motor which connects to the link by a nonlinear and variable elastic element. Feedback linearization has been used to control link motion desired stiffness in the antagonistic VSA setup of 1-DOF arm using VSA-II prototype developed by researchers at the University of Pisa [24]. Qbmove maker pro is one of the newest low-cost 3D printed VSA developed by the

Natural Motion Initiative. Qbmove maker pro is bidirectional spring antagonistic actuator design as open source making it affordable and can be used for research and educational purposes in robotics. Qbmove maker pro actuator and its structure based on the work in [25] is shown in Fig. 2.4. The elastic elements in VSAs have low damping coupled with its ideal stiffness. The advantages of the low dampings are:

- Proper estimation of torques acting on the elastic element
- String element can store energy and the energy can easily be recovered with little loss.

In general string stiffness can be constant and linear in the case of string elastic actuator, linear variable stiffness In summary, the antagonistic Variable stiffness Actuator model shown in Fig. 2.2, the dynamics is represented by the following set of equations:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_1(\psi_1) + \boldsymbol{\tau}_2(\psi_2) = \boldsymbol{\tau}_{ext} \tag{2.3}$$

$$\mathbf{J}_1\ddot{\mathbf{q}}_{s1} - \boldsymbol{\tau}_e(\psi) = \boldsymbol{u}_1 \tag{2.4}$$

$$\mathbf{J}_1\ddot{\mathbf{q}}_{s2} - \boldsymbol{\tau}_2(\psi) = \boldsymbol{u}_2 \tag{2.5}$$

For the Variable stiffness Actuator model shown in Fig. 2.3, the dynamics is represented by the following set of equations.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_e(\mathbf{q}_s, \phi) = \boldsymbol{\tau}_{ext} \tag{2.6}$$

$$\mathbf{J}_1\ddot{\mathbf{q}}_{s1} - \boldsymbol{\tau}_e(\mathbf{q}_s, \phi) = \boldsymbol{u}_1 \tag{2.7}$$

$$\mathbf{J}_1\ddot{\mathbf{q}}_{s1} - \boldsymbol{\psi}_e(\mathbf{q}_s, \phi) = \boldsymbol{\tau}_c \tag{2.8}$$

Existing works in [5], [6], [24] based on different forms of VSA robot have results that shows that VSAs are capable of compliant control in bipedal robots and industrial manipulators and by extension we can apply same technique to quadruped robots. This provides a baseline preliminary result and justification for this project work. While we achieve control of robots with active compliance through control laws previously discussed, it comes with some few challenges which makes it a bit unreliable. The work done by [26] opined that in active compliance, there can be sensor failure , the system may suffer from the kinematic singularity, there is an upper limit on the desired stiffness in position control and it may be limited by sampling frequency in estimating the reaction forces. Passive compliance on the other hand is the intrinsic mechanical structural compliance due to the finite stiffness of the robot structure like the robot base, links, actuators, strings, joint drive mechanism, grippers as well as other assembled parts. Control forces in passive compliance could be a bit hard to compute and regulate , it however offers advantage of being generally stable, fast response, cheap and can be set with a stiffness. In next chapter, we study and introduce control mechanisms for passive compliance using series elastic actuator and variable stiffness actuators.

Chapter 3

Methodology

3.1 Robot dynamics: Lagrangian Formulation

The lagrangian formulation of dynamics of mechanical systems is used to choose coordinates $q \in \mathbb{R}^N$ that describes the robot configuration. These coordinates are called the generalized coordinates and it represents the degree of freedom of our robot. These generalized coordinates are then used to define the generalized torques $\boldsymbol{\tau} \in \mathbb{R}^N$. A Lagrangian function $\mathcal{L}$, which is the difference between the total kinetic energy $\mathcal{T}$ and the total potential energy $\mathcal{U}$, of the system at each instant:

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{3.1}$$

The equations of motion can now be expressed in terms of the Lagrangian as follows:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \boldsymbol{\tau}_i \qquad i = 1, ..., N \tag{3.2}$$

where $\boldsymbol{\tau}$= non-conservative generalized forces performing work on q_i

The equations above in (3.2) allows us to find a relationship between the generalized forces applied to the manipulator and the joint positions, velocities and accelerations. With this, one can make a derivation of the dynamic model of the manipulator by making use of the derivation of the kinetic energy and potential energy of the mechanical system which is shown in the following subsection.

3.1.1 Kinetic energy of a rigid body

The kinetic energy of an infinitesimal small point masses which belongs to a rigid body B is given by:

$$\mathcal{T} = \frac{1}{2}\int_B (v^T(x,y,z)v(x,y,z)dm \tag{3.3}$$

The kinematic relation is made up of the linear component and rotational component as given by:

$$v = v_c + \omega \times r = v_c + S(\omega)r \tag{3.4}$$

where:

S	:	a skew symmetric matrix
x, y, z	:	cartesian coordinate in 3D plane
v	:	velocity of the body
ω	:	angukar velocity of body B
r	:	radius of body B to centre of mass.

$$\mathcal{T} = \frac{1}{2}\int_B (v_c + S(\omega)r)^T[(v_c + S(\omega)r)]dm \tag{3.5}$$

$$\mathcal{T} = \frac{1}{2}\int_B v_c^T v_c dm + \int_B v_c^T S(\omega) r dm + \frac{1}{2}\int_B r^T S^T(\omega) S(\omega) r dm \tag{3.6}$$

recall that $a^Tb = trace(ab^T)$

$$\mathcal{T} = \frac{1}{2}\int_B v_c^T v_c dm + \cancelto{0}{v_c^T S(\omega)\int_B rdm} + \frac{1}{2}\int_B trace(S(\omega)r.r^T S^T(\omega))dm \quad (3.7)$$

When all the vectors of infinitesimal masses are referred to a mathbfdy centre of mass, their sum is equivalent to:

$$\int_B rdm = 0$$

$$\mathcal{T} = \frac{1}{2}\int_B v_c^T v_c dm + \frac{1}{2}trace\Big\{S(\omega)\int_B \Big(r.r^T dm\Big)S^T(\omega)\Big\} \quad (3.8)$$

$$\mathcal{T} = \frac{1}{2}mv_c^T v_c + \frac{1}{2}trace\Big\{S(\omega)F_c S^T(\omega)\Big\} \quad (3.9)$$

$$\mathcal{T}_i = \frac{1}{2}m_i v_{ci}^T v_{ci} + \frac{1}{2}\omega_i^T I_{ci}\omega_i \quad (3.10)$$

The above relation is referred to as Konig theorem and it states that the kinetic energy of a rigid body is the sum of the translational kinetic energy and the rotational kinetic energy of the whole body.

$$\mathcal{T}_i = \frac{1}{2}\dot{\mathbf{q}}^T \underbrace{\Big(\sum_{i=1}^{N} m_i J_{Li}^T(q)J_{Li}(q) + J_{\omega i}^T(q)J_{\omega i}(q)\Big)\dot{\mathbf{q}}}_{\mathbf{M}(\mathbf{q},\dot{\mathbf{q}})} \quad (3.11)$$

$$\mathcal{T}_i = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \quad (3.12)$$

where $\mathbf{M}$ = robot generalized inertia matrix (symmetric, positive definite and $\forall\,\mathbf{q}$, $\mathbf{M}$ is always invertible.

3.1.2 Lagrangian Formulation

The dynamics of the robot manipulator provides a description of the relationship between the actuator torques and the motion of the robotic system. With Lagrange formulation, the equations of motion can be derived in a systematic way independently of the reference coordinate frame. The formulation described in this section is based on the formulation as described in [27]. Lagrangian of the mechanical system can be defined as a function of the generalized coordinates:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = \boldsymbol{\tau}_i \qquad\qquad i = 1, ..., N \tag{3.13}$$

where $\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q)$

Defining each term in (3.13):

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{1}{2}\dot{q}^T \sum_{i=1}^{N} \frac{\partial \mathbf{M}}{\partial q}\dot{q} - \frac{\partial \mathcal{U}}{\partial q_i} \tag{3.14}$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} = \mathbf{M}\dot{\mathbf{q}} \tag{3.15}$$

$$\frac{d}{dt}\Big(\mathbf{M}\dot{\mathbf{q}}\Big) - \frac{1}{2}\dot{\mathbf{q}}^T\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathcal{U}}{\partial \mathbf{q}_1} = \boldsymbol{\tau} \tag{3.16}$$

Differentiating $\mathbf{M}\dot{\mathbf{q}}$ in the above equation, we obtain:

$$\mathbf{M}\ddot{\mathbf{q}} + \dot{\mathbf{M}}\dot{\mathbf{q}} - \frac{1}{2}\dot{\mathbf{q}}^T\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathcal{U}}{\partial q_i} = \boldsymbol{\tau} \tag{3.17}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \sum_{i=1}^{N}(\frac{\partial h_i}{\partial \mathbf{q}})\dot{\mathbf{q}}_i\dot{\mathbf{q}} - \frac{1}{2}\dot{\mathbf{q}}^T\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathcal{U}}{\partial \mathbf{q}_i} = \boldsymbol{\tau} \tag{3.18}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \underbrace{\sum_{i=1}^{N}(\frac{\partial h_i}{\partial \mathbf{q}})\dot{\mathbf{q}}_i\dot{\mathbf{q}} - \frac{1}{2}\dot{\mathbf{q}}^T\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\dot{\mathbf{q}}}_{\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}} + \underbrace{\frac{\partial \mathcal{U}}{\partial \mathbf{q}_i}}_{\mathbf{g}(q)} = \boldsymbol{\tau} \tag{3.19}$$

$$\mathbf{M}\ddot{q} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{3.20}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \boldsymbol{\tau} \tag{3.21}$$

3.2 Lagrange Equations with reaction forces

$$m_i\ddot{\mathbf{r}}_i = \mathbf{f}_i, \quad i = 1, \; ... \; m \tag{3.22}$$

Assume we have a reaction force $\lambda \in \mathbb{R}^3$, acting on the point $\mathbf{r}_1$. We can easily see, that the equation (3.22) will attain form:

$$M\ddot{\mathbf{r}} = f + \begin{bmatrix} I_{3\times 3} \\ 0_{3(m-1)\times 3} \end{bmatrix} \lambda \tag{3.23}$$

After multiplying (3.23) by the jacobian $\partial\mathbf{r}/\partial\mathbf{q}$ it takes the form:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{T}}{\partial \mathbf{q}} = \boldsymbol{\tau} + \left(\frac{\partial \mathbf{r}_1}{\partial \mathbf{q}}\right)^{\top} \lambda \tag{3.24}$$

Notice that by the same logic we can show that for reactions force in $\lambda \in \mathbb{R}^3$ (and indeed, any external force) applied to the point $\mathbf{r}_j$, Lagrange equations will take the form:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{T}}{\partial \mathbf{q}} = \boldsymbol{\tau} + \left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{q}}\right)^{\top} \lambda \tag{3.25}$$

In general, in the presence of any conservative force or potential energy, we have:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} + \left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{q}}\right)^{\top} \lambda \tag{3.26}$$

Thus, for any reaction force that enforces a constraint $g(q) \in \mathcal{R}^k$, the general form of the Lagrange equation is:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} + \left(\frac{\partial \mathbf{g}_j}{\partial \mathbf{q}}\right)^{\top} \lambda \tag{3.27}$$

In order to describe a system with explicit constraints $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^k$, we need to explicitly include them in the dynamics formulation:

$$\begin{aligned} \frac{d}{dt}(\frac{\partial \mathcal{T}}{\partial \dot{\mathbf{q}}}) - \frac{\partial \mathcal{T}}{\partial \mathbf{q}} &= \boldsymbol{\tau} + (\frac{\partial \mathbf{g}}{\partial \mathbf{q}})^{\top} \lambda \\ \mathbf{g}(\mathbf{q}) &= 0 \end{aligned} \tag{3.28}$$

The dynamics of the robot under constrained reaction force is thus given as:

$$\mathbf{M}\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{g}(q) = \boldsymbol{\tau} + \mathbf{F}_c^T \lambda \tag{3.29}$$

where:

$\lambda \in R^K$	:	Lagrange multiplier
$\mathbf{M} \in \mathbb{R}^{n\times n}$	:	generalized inertia matrix
$\mathbf{F}_c \in \mathbb{R}^{m\times n}$	:	geometric Jacobian matrix(eg from contacts)
$\mathbf{C} \in \mathbb{R}^{n\times n}$	:	centrifugal and Coriolis forces
$\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^{n\times 1}$	:	generalized coordinates
$\boldsymbol{\tau} \in \mathbb{R}^n$	:	external generalized forces
$\mathbf{g}(q) \in \mathbb{R}^n$	:	gravitational terms

3.3 Inverse dynamics with PD control

$$\boldsymbol{\tau} = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) \tag{3.30}$$

At steady state, we have:

$$\cancelto{0}{\mathbf{M}\ddot{q}(q,\dot{q})} + \cancelto{0}{\mathbf{C}(q,\dot{q})}\dot{q} + \mathbf{g}(q) = \cancelto{0}{k_p(q^* - q)} + \cancelto{0}{k_d(\dot{q}^* - \dot{q})} \tag{3.31}$$

To compensate for steady-state offset due to gravity, the actuator torque is applied on the robot joints as:

$$\boldsymbol{\tau} = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{3.32}$$

3.4 Floating base kinematics and dynamics

Free-floating robots like the quadruped and humanoid shown in (put figure) are have an n_b un-actuated base coordinates q_b and n_j actuated joint coordinates q_j. A typical quadruped would have 12 actuated joint coordinates.

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{bmatrix} \tag{3.33}$$

The un-actuated floating base coordinate can be represented as:

$$\mathbf{q}_b = \begin{bmatrix} \mathbf{q}_{bP} \\ \mathbf{q}_{bR} \end{bmatrix} \tag{3.34}$$

q_{bP} and q_{bR} can be parametized using the minimal representation of the cartesian coordinate and euler angle representation respectively. Thus, the dimension of a legged robot with floating base system would be $n_b + n_j$, where $n_b = 6$

$$\begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{C}_1(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}_1(\mathbf{q}) \\ \mathbf{C}_2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{c1} \\ \mathbf{F}_{c2} \end{bmatrix} \boldsymbol{\lambda} \tag{3.35}$$

we have:

$$\begin{cases} \mathbf{F}_c \dot{\mathbf{q}} = \mathbf{0} \\ \mathbf{F}_c \ddot{\mathbf{q}} + \dot{\mathbf{F}}_c \dot{\mathbf{q}} = \mathbf{0} \end{cases}$$

3.5 Floating Base Inverse Dynamics Control

The floating base framework for legged robots provides a representation of a legged robot rigid-body system unattached to the world, and is necessary to describe the complete dynamics of the system with respect to an inertial frame. Floating base link is generally unactuated thus it introduces unactuation into the dynamics of the robot.

When the robot is in contact with the environment, the equations of motion with respect to an inertial frame are given by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(, \dot{\mathbf{q}})^{\cdot} + \mathbf{g}(\mathbf{q}) = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{F}_c^T \boldsymbol{\lambda} \tag{3.36}$$

where:

$\lambda \in R^K$	:	Lagrange multiplier
$\mathbf{M} \in \mathbb{R}^{n\times n}$	:	generalized inertia matrix
$\mathbf{F}_c \in \mathbb{R}^{m\times n}$	:	geometric Jacobian matrix(eg from contacts)
$\mathbf{C} \in \mathbb{R}^{n\times n}$	:	centrifugal and Coriolis forces
$\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^{n\times 1}$	:	generalized coordinates
$\boldsymbol{\tau} \in \mathbb{R}^{n}$	:	external generalized forces
$\mathbf{g}(q) \in \mathbb{R}^{n}$	:	gravitational terms
$\mathbf{S} = [\mathbf{I}_{n\times n} \quad \mathbf{0}_{n\times 6}]$	:	actuated joint selection matrix

3.6 Computed Torque Control

Recall that:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{3.37}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \boldsymbol{\tau} \tag{3.38}$$

We can represent $\boldsymbol{\tau} = \mathbf{Tu}$ and rewrite equations of dynamics in the normal form where $\mathbf{T}$ is the control selection matrix (map)

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{Tu} - \mathbf{c}) \tag{3.39}$$

Let us define *control error* $\mathbf{e}$ as follows:

$$\mathbf{e} = \mathbf{q}^* - \mathbf{q} \tag{3.40}$$

Then we can find its second derivative as: $\ddot{\mathbf{e}} = \ddot{\mathbf{q}}^* - \ddot{\mathbf{q}}$:

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}}^* - \mathbf{M}^{-1}(\mathbf{Tu} - \mathbf{c}) \tag{3.41}$$

If error dynamics is *stable*, it means the error will approach zero as the time approaches infinity. We can decide that we want error dynamics to have this form:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{3.42}$$

where $\mathbf{K}_d$ and $\mathbf{K}_p$ are diagonal positive-definite matrices.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{T}\mathbf{u} = \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c} \tag{3.43}$$

$$\mathbf{u} = \mathbf{T}^{+}(\mathbf{M}(\ddot{\mathbf{q}}^{*} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c}) \tag{3.44}$$

This is called a *computed torque controller* (CTC), and it assumes that $(\mathbf{M}(\ddot{\mathbf{q}}^{*} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c})$ is in the column space of $\mathbf{T}$.

3.7 QR Decomposition

The inverse dynamics and computed torque controller describe in section 3.6 and 3.3 have been used extensively for fixed base robot manipulators like industrial manipulators that are fully actuated. However, this approached do not work for the ill-posed floating base systems. For the particular case of our under-actuated robot dynamics in the floating base dynamics, we would like to compute the actuated joint torques, $\boldsymbol{\tau}$ that will realize the desired joint trajectory. Solving for $\boldsymbol{\tau}$ requires full knowledge of the constraint forces, λ. However, these constraint forces depend on the actuation torques applied. Due to this codependance between λ and $\boldsymbol{\tau}$, there are potentially infinitely many solutions for $\boldsymbol{\tau}$, and because of under-actuation, certain values of desired acceleration may yield no solution at all.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{T}\mathbf{u} + \mathbf{F}^{\top}\boldsymbol{\lambda} \tag{3.45}$$

It should be noted that (3.45) contains n+6 equations but with only a possible n control torques which makes it under-actuated. Orthogonal decomposition allows to eliminate constraint forces and fully represent the robot dynamics in a reduced fully actuated form. If we assume $\mathit{Rank}(\mathbf{F}) = \mathbf{k}$, then there are $\mathbf{k}$ linearly independent constraints and F_c is full row rank. We can then represent the constraint Jacobian $\mathbf{F}^{\top}$ as its QR decomposition:

$$\mathbf{F}^{\top} = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}$$

where $\mathbf{Q}$ is orthogonal that is $\mathbf{Q}^{\top}\mathbf{Q} = \mathbf{Q}\mathbf{Q}^{\top} = \mathbf{I}$ and $\mathbf{R}$ is an upper triangle matrix of rank $\mathbf{k}$. If R have strictly positive diagonal elements, then Q and R are going to be unique. The full derivation of the QR decomposition based inverse dynamics based on the work in [3] is shown in the rest of this section.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \mathbf{T}\mathbf{u} + \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\lambda \tag{3.46}$$

$$\mathbf{Q}^{\top}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) = \mathbf{Q}^{\top}\mathbf{T}\mathbf{u} + \begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\lambda \tag{3.47}$$

Introducing switching variables to select the constrained dynamics and unconstrained dynamics and

$$\mathbf{S}_1 = \begin{bmatrix}\mathbf{I_{k\times k}} & \mathbf{0_{k\times(n+6-k)}}\end{bmatrix}$$

and

$$\mathbf{S}_2 = \begin{bmatrix} \mathbf{0}_{(\mathbf{n+6-k})\times \mathbf{k}} & \mathbf{I}_{(\mathbf{n+6-k})\times(\mathbf{n+6-k})} \end{bmatrix}$$

and multiplying equations by one and the other we get two systems:

$$\begin{cases} \mathbf{S}_1\mathbf{Q}^\top(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) = \mathbf{S}_1\mathbf{Q}^\top\mathbf{T}\mathbf{u} + \mathbf{R}\,\lambda \\ \mathbf{S}_2\mathbf{Q}^\top(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) = \mathbf{S}_2\mathbf{Q}^\top\mathbf{T}\mathbf{u} \end{cases} \tag{3.48}$$

The main advantage we achieved is that now we can calculate both **u** and λ

$$\mathbf{u} = (\mathbf{S}_2\mathbf{Q}^\top\mathbf{T})^+\mathbf{S}_2\mathbf{Q}^\top(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) \tag{3.49}$$

Expression for λ is:

$$\lambda = \mathbf{R}^{-1}\mathbf{S}_1\mathbf{Q}^\top(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) \tag{3.50}$$

We can notice a pseudo-inverse, implying that the no-residual solution does not have to exist.

3.8 Constrained LQR

To use a linear controller like LQR, we need to linearize our system along each point in the trajectory of desired motion.

3.8.1 Linearizing the dynamics

Recall that a linear system of equation is given as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.51}$$

The problem with most dynamical system is that they are non linear thus it is impossible to express such dynamical equation in form of (3.51). Therefore if we linearize such system, we can have our expression in form of (3.51)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{Tu} \tag{3.52}$$

$$\ddot{\mathbf{q}} = \mathbf{M^{-1}}[\mathbf{Bu} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q})] \tag{3.53}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix} \tag{3.54}$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.55}$$

To proceed with the problem of linearization, we use Taylor' series expansion formulation to linearize as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.56}$$

$$\dot{\mathbf{x}} \approx \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{x=x^*, u=u^*} (\mathbf{x} - \mathbf{x}^*) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right]_{x=x^*, u=u^*} (\mathbf{u} - \mathbf{u}^*) \tag{3.57}$$

Linearizing the manipulator at the stable equilibrium points, such that $\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = 0$ we obtain the standard linear state space form as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}[\mathbf{T}\mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})] \end{bmatrix} \tag{3.58}$$

Thus we have:

$$A = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M^{-1}} \left[\sum_j \frac{\partial B_j}{\partial q} u_j - \frac{\partial g(q)}{\partial q}\right] & \mathbf{0} \end{bmatrix}_{x=x^*, u=u^*} (\mathbf{x} - \mathbf{x}^*) \tag{3.59}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{T} \end{bmatrix}_{x=x^*, u=u^*} (\mathbf{u} - \mathbf{u}^*) \tag{3.60}$$

It would be observed from (3.59) that we do not have any expression $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ because at equilibrium point $\dot{\mathbf{q}}^* = 0$, all of $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = 0$. The cost function for the LQR is defined as :

$$J = \int_0^\infty \left[x^T Q x + u^T R u\right] dt, \quad Q = Q^T \succcurlyeq 0, \quad R = R^T \succcurlyeq 0$$

The LQR optimal control allows us to find a feedback matrix gain K for our states. The feedback control law is given as $u(t) = -\mathbf{K}\mathbf{x}$ Now ,we want to address the case of the robot under a contact reaction force.

$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{T}\mathbf{u} + \mathbf{F}_c^T \lambda \\ \mathbf{x}_c = \mathbf{0} \end{cases} \tag{3.61}$$

recall that for a contact point c:

$$\mathbf{x_c} = \mathbf{0}$$

$$\dot{\mathbf{x}}_c = \mathbf{F}_c\dot{\mathbf{q}} = 0 \tag{3.62}$$

$$\begin{cases} \mathbf{F}_c\dot{\mathbf{q}} = \mathbf{0} \\ \mathbf{F}_c\ddot{\mathbf{q}} + \dot{\mathbf{F}}_c\dot{\mathbf{q}} = \mathbf{0} \end{cases} \tag{3.63}$$

$$\begin{bmatrix} \mathbf{F}_c & \mathbf{0} \\ \dot{\mathbf{F}}_c & \mathbf{F}_c \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.64}$$

.

Projecting the null-space of the kinematic constraint of Eq.3.118 using the projection matrix N,

$$N = null\left(\begin{bmatrix} \mathbf{F}_c & \mathbf{0} \\ \dot{\mathbf{F}}_c & \mathbf{F}_c \end{bmatrix}\right) \tag{3.65}$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{c} \tag{3.66}$$

The cost function for the LQR is defined as :

$$J = \textstyle\int_0^\infty \left[x^TQx + u^TRu\right] dt, \quad Q = Q^T \succcurlyeq 0, \quad R = R^T \succcurlyeq 0$$

$$\dot{\mathbf{z}} = \mathbf{A}_N\mathbf{z} + \mathbf{B}_N\mathbf{u} + \mathbf{c}_N, \tag{3.67}$$

where **Q** and **R** are matrices defining cost function for the LQR problem. feedback control law:

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x} \tag{3.68}$$

$$\mathbf{A}_N = \mathbf{N}^\top \mathbf{A}\mathbf{N} \quad \mathbf{B}_N = \mathbf{N}^\top \mathbf{B} \quad \mathbf{c}_N = \mathbf{N}^\top \mathbf{c} \quad \mathrm{k} = \mathbf{K}_{\mathbf{N}}\mathbf{N}$$

3.9 Nested Quadratic Programming

The nested quadratic programming approach described in this chapter is based on the work in [28] where same technique was applied to in-pipe biped robots by taking into accounts of equality, inequality mechanical constraints and bounds on control actions . Recall the dynamics of our robot under mechanical constraint is described as:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{T}\mathbf{u} + \mathbf{F}_c^T \lambda \\ \mathbf{F}_c\ddot{\mathbf{q}} - \dot{\mathbf{F}}\dot{\mathbf{q}} = 0 \end{cases} \tag{3.69}$$

We also recall the relaxed version of the dynamics in (3.70) is :

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \boldsymbol{\tau} \tag{3.70}$$

An assumption is made that a controller like PD produced the torque($\boldsymbol{\tau}$) and $\boldsymbol{\tau}^*$ signifies an inner layer controller which produces reference values for the control actions and then adjusts them using the information from constraints, bounds and dynamics of the system.

We then proceed with the formulation by rewriting (3.70) in terms of control error, $\mathbf{e} = \mathbf{q}^* - \mathbf{q}$, where $\mathbf{q}^*$ is vector of the desired values of the generalized coordinates:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} - \mathbf{M}\ddot{\mathbf{e}} + \mathbf{c} = \mathbf{T}\mathbf{u} + \mathbf{F}_c^T \lambda \\ \mathbf{F}_c\ddot{\mathbf{q}} - \mathbf{F}_c\ddot{\mathbf{e}} + \dot{\mathbf{F}}_c\dot{\mathbf{q}} = 0 \\ \boldsymbol{\tau}^* = \mathbf{T}\mathbf{u} + \mathbf{F}_c^T \lambda \end{cases} \tag{3.71}$$

Vectors $\boldsymbol{\tau}, \mathbf{c}$ and, $\mathbf{q}$ have length n , while vectors $\mathbf{F}_c$ have lengths k1. u is the vector of control actions (parameters that can be directly influenced by the control system), and T is a linear transformation that maps u to the space of generalized forces. Vector u has length m Since equation (3.71) form our constraint equation for optimization task, we note that the decision variables are: $\ddot{\mathbf{e}}$, $\ddot{\lambda}$ and $\ddot{\mathbf{u}}$ and there are n+m+k optimization parameters in total. For a floating base quadruped robot with two point contacts, we note the following as an example:

1. n has a size of 18 representing the generalized coordinate
2. m representing the number of actuated joint is 12
3. k =6 since the size of each point contact is 3
4. the total number of decision variables is n+m+k (18+12+6)=36

The optimization problem can become easily in feasible especially when $m < n$, in such case, we need to modify the outer controller equation in the (3.71) with a set of slack variable s as:

$$\boldsymbol{\tau}^* = \mathbf{Tu} + \mathbf{F_c}^T \lambda + s \tag{3.72}$$

The slack variable s represents the difference between the reference value of generalized forces $\boldsymbol{\tau}^*$ and what we can actually obtain taking into account the constraints. If equations (3.72) and (3.71) can be satisfied simultaneously, then the optimal value of s is a zero vector. Thus, after adding s as a slack variable, the new set of optimization or decision variable is given as:

$$\mathbf{x} = \begin{bmatrix} \ddot{\mathbf{e}} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ s \end{bmatrix} \tag{3.73}$$

A general quadratic programming problem is defined as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{P}\mathbf{x} + \mathbf{Q}^\top \mathbf{x}, \\ \text{subject to} \quad & \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ \mathbf{C}\mathbf{x} = \mathbf{d}. \end{cases} \end{aligned} \tag{3.74}$$

where $\mathbf{P}$ is positive-definite and $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ describe a *convex region.*

$$\mathbf{x} = \begin{bmatrix} \ddot{\mathbf{e}} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ s \end{bmatrix} \tag{3.75}$$

The QP formulation for the robot dynamics based on definition previously expressed in (3.74) and constraints in (3.71) is defined as:

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^\top \mathbf{P}\mathbf{x}, \\
&\text{subject to} \begin{cases} \begin{bmatrix} \mathbf{M} & \mathbf{T} & \mathbf{F_c^T} & \mathbf{0} \\ \mathbf{F_c^T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} & \mathbf{F_c^T} & \mathbf{I} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \ddot{\mathbf{e}} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ s \end{bmatrix} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} \\ \mathbf{F}_c \ddot{\mathbf{e}} + \dot{\mathbf{F}}_c \dot{\mathbf{q}} \\ \boldsymbol{\tau}^* \end{bmatrix} \\ \mathbf{A}\mathbf{x} \le \mathbf{b}. \end{cases}
\end{aligned} \tag{3.76}$$

where P is a positive definite matrix, **A** and **b** are a matrix and a vector used to describe all inequality constraints imposed on the variables, and **I** is a $n \times n$ identity matrix

$$\begin{bmatrix} \mathbf{M} & \mathbf{T} & \mathbf{F_c^T} & \mathbf{0} \\ \mathbf{F_c^T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} & \mathbf{F_c^T} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{e}} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} \\ \mathbf{F}_c \ddot{\mathbf{e}} + \dot{\mathbf{F}}_c \dot{\mathbf{q}} = 0 \\ \boldsymbol{\tau}^* \end{bmatrix} \tag{3.77}$$

3.10 Manipulator with SEA dynamics

Recall that the general equation of motion of rigid manipulator is :

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \tag{3.78}$$

The manipulator with a Series Elastic Actuator joint dynamics is given as:

$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{K}(\mathbf{q} - \mathbf{q}_s) = \mathbf{0} \\ \mathbf{J}\ddot{\mathbf{q}}_s - \mathbf{K}(\mathbf{q} - \mathbf{q}_s) = \mathbf{u} \end{cases} \tag{3.79}$$

where:

$\mathbf{M} \in \mathbb{R}^{n \times n}$	:	generalized inertia matrix
$\mathbf{C} \in \mathbb{R}^{n \times n}$	:	centrifugal and Coriolis forces
$\mathbf{g}(q) \in \mathbb{R}^{n \times 1}$	:	gravitational terms
$\mathbf{u} \in \mathbb{R}^{n \times 1}$	:	external generalized forces
$\mathbf{J} \in \mathbb{R}^{n \times 1}$	:	elastic actuator inertia matrix
$\mathbf{K} \in \mathbb{R}^{n \times n}$	:	stiffness coefficient elastic actuator
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$	:	robot joints generalized coordinates
$\mathbf{q}_s, \dot{\mathbf{q}}_s, \ddot{\mathbf{q}}_s \in \mathbb{R}^{n \times 1}$	:	elastic joints generalized coordinates

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} \tag{3.80}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \qquad \mathbf{x}_s = \begin{bmatrix} \mathbf{q}_s \\ \dot{\mathbf{q}}_s \end{bmatrix} \tag{3.81}$$

We know **A** and **B** by linearizing the dynamics in (3.78). Let $\mathbf{C}_1$ represent a switching variable and n represents the size of **q**. We assume in this case **q** and $\mathbf{q}_s$ have the same size and all robot joints have elastic actuators.

$$\mathbf{v} = \mathbf{K}(\mathbf{q}_s - \mathbf{q}) \tag{3.82}$$

Let:

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{C}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}$$

I and **0** are square matrices of size n.

$$\mathbf{J}\ddot{\mathbf{q}}_s - \mathbf{K}(\mathbf{q} - \mathbf{q}_s) = \mathbf{u} \tag{3.83}$$

$$\ddot{\mathbf{q}}_s = -\mathbf{J}^{-1}\mathbf{K}(\mathbf{q}_s - \mathbf{q}) + \mathbf{J}^{-1}\mathbf{u} \tag{3.84}$$

$$\ddot{\mathbf{q}}_s = -\mathbf{J}^{-1}\mathbf{K}(\mathbf{C}_1\mathbf{x}_s - \mathbf{C}_1\mathbf{x}) + \mathbf{J}^{-1}\mathbf{u} \tag{3.85}$$

Our new linear state representation considering elastic actuator dynamics is given as:

$$\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} \tag{3.86}$$

where:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \mathbf{q}_s \\ \dot{\mathbf{q}}_s \end{bmatrix} \tag{3.87}$$

$$\dot{\mathbf{z}} = \begin{bmatrix} \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}(\mathbf{C}_1\mathbf{x}_s - \mathbf{C}_1\mathbf{x}) \\ [\mathbf{0} \quad \mathbf{I}]\mathbf{x}_s \\ -\mathbf{J}^{-1}\mathbf{K}(\mathbf{C}_1\mathbf{x}_s - \mathbf{C}_1\mathbf{x}) + \mathbf{J}^{-1}\mathbf{u} \end{bmatrix} \tag{3.88}$$

$$\dot{\mathbf{z}} = \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{K}\mathbf{C}_1 & \mathbf{B}\mathbf{K}\mathbf{C}_1 \\ \mathbf{0} & \mathbf{C}_2 \\ \mathbf{J}^{-1}\mathbf{K}\mathbf{C}_1 & -\mathbf{J}^{-1}\mathbf{K}\mathbf{C}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{J}^{-1} \end{bmatrix} \mathbf{u} \tag{3.89}$$

$$\dot{\mathbf{z}} = \underbrace{\begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{K}\mathbf{C}_1 & \mathbf{B}\mathbf{K}\mathbf{C}_1 \\ \mathbf{0} & \mathbf{C}_2 \\ \mathbf{J}^{-1}\mathbf{K}\mathbf{C}_1 & -\mathbf{J}^{-1}\mathbf{K}\mathbf{C}_1 \end{bmatrix}}_{\tilde{\mathbf{A}}} \mathbf{z} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{J}^{-1} \end{bmatrix}}_{\tilde{\mathbf{B}}} \mathbf{u} \tag{3.90}$$

where $\mathbf{0}$ and $\mathbf{I}$ are matrices of appropriate sizes. The feedback component of $\mathbf{u}$ is given as:

$$\mathbf{u}_{fb} = -\mathbf{K}_s(\mathbf{x} - \mathbf{x}^{\mathbf{d}}) \tag{3.91}$$

$\mathbf{K}_s$ is obtained from solving algebraic ricatti equation The feed-forward force is given as:

$$\mathbf{u}_{ff} = \tilde{\mathbf{B}}^{+}(\dot{\mathbf{z}_d} - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{C}}) \tag{3.92}$$

$$\tilde{\mathbf{C}} = \dot{\mathbf{z}}_d - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{B}}\mathbf{u}_{ff} \tag{3.93}$$

where $\mathbf{z_d}$ is a vector the desired trajectories of joint states and elastic actuator states.

Thus, $\mathbf{u}$ in (3.90) is given as:

$$\mathbf{u} = \mathbf{u}_{fb} + \mathbf{u}_{ff} \tag{3.94}$$

where $\mathbf{u}_{ff}$ is feed-forward inverse dynamics expressed in (3.92) and K is optimal gain computed from LQR riccati equation.

3.11 SEA with constraints

For the robot with SEA joint, the dynamics is expressed as:

$$\begin{cases} \mathbf{M}(\mathbf{q}_1)\ddot{\mathbf{q}}_1 + \mathbf{C}(\mathbf{q_1}, \dot{\mathbf{q_1}}) + \mathbf{g}(\mathbf{q_1}) + \mathbf{K}(\mathbf{q}_1 - \mathbf{q}_2) = \mathbf{F}_c^T \lambda \\ \mathbf{J_s}\ddot{\mathbf{q}}_2 - \mathbf{K}(\mathbf{q}_1 - \mathbf{q}_2) = \mathbf{Tu} \end{cases} \tag{3.95}$$

$$\underbrace{\begin{bmatrix} \mathbf{F}_c & \mathbf{0} & \mathbf{0} \\ \dot{\mathbf{F}}_c & \mathbf{F}_c & \mathbf{0} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \mathbf{x}_s \end{bmatrix}}_{\dot{\mathbf{z}}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.96}$$

Combining the constraint equation above in (3.96) with the linearized form of dynamics in (3.86) with its terms previously defined in (3.90) and (3.93), we have:

$$\begin{cases} \dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} + \bar{\mathbf{F}}\,\lambda \\ \mathbf{G}\dot{\mathbf{z}} = \mathbf{0} \end{cases} , \quad \bar{\mathbf{F}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{H^{-1}}\mathbf{F}_c^T \end{bmatrix} \tag{3.97}$$

$$\mathbf{u} = -\mathbf{K}_s(\mathbf{z} - \mathbf{z}_d) + \tilde{\mathbf{B}}^{+}(\dot{\mathbf{z}_d} - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{C}}) \tag{3.98}$$

$$\tilde{\mathbf{C}} = \dot{\mathbf{z}}_d - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{B}}\mathbf{u}_{\mathit{ff}} \tag{3.99}$$

$$\begin{bmatrix} \dot{\mathbf{z}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\bar{\mathbf{F}} \\ \mathbf{G} & \mathbf{0} \end{bmatrix}^{+} \begin{bmatrix} \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} \\ \mathbf{0} \end{bmatrix} \tag{3.100}$$

where $\mathbf{I}$ and $\mathbf{0}$ are matrices of appropriate sizes. where:

$$\mathbf{u}(t) = \mathbf{u}_{\mathit{ff}} - \mathbf{K}(\mathbf{x} - \mathbf{x}^{*}) \tag{3.101}$$

where $\mathbf{u}_{\mathit{ff}}$ is feed-forward inverse dynamics expressed in (3.92) and $\mathbf{K}$ is optimal gain computed from LQR riccati equation.

3.12 Agonistic-Antagonistic Variable-Stiffness-Actuator

$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \\ \mathbf{J}_1\ddot{\mathbf{q}}_{s1} + \boldsymbol{\tau}_1 = \mathbf{u}_1 \\ \mathbf{J}_2\ddot{\mathbf{q}}_{s2} + \boldsymbol{\tau}_2 = \mathbf{u}_2 \end{cases} \tag{3.102}$$

$$\boldsymbol{\tau}_i = \boldsymbol{\sigma}_i \sinh(\boldsymbol{a}_i(\mathbf{q}_{si} - \mathbf{q})) \quad , \quad i{=}1{,}2 \tag{3.103}$$

The stiffness of the VSA motor is given as:

$$\mathbf{K}_i = -\frac{\partial \boldsymbol{\tau}(\mathbf{q}_s - \mathbf{q}, \boldsymbol{\sigma})}{\partial \mathbf{q}} = \quad \boldsymbol{\sigma}_i \mathbf{a}_i \mathrm{cosh}(\boldsymbol{a}_i(\mathbf{q}_{si} - \mathbf{q})) \quad , \quad i{=}1{,}2 \tag{3.104}$$

$$\mathbf{v} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \tag{3.105}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}, \qquad \mathbf{x}_{s1} = \begin{bmatrix} \mathbf{q}_{s1} \\ \dot{\mathbf{q}}_{s1} \end{bmatrix}, \qquad \mathbf{x}_{s2} = \begin{bmatrix} \mathbf{q}_{s2} \\ \dot{\mathbf{q}}_{s2} \end{bmatrix}, \qquad \mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{s1} \\ \mathbf{x}_{s2} \end{bmatrix} \tag{3.106}$$

$$\ddot{\mathbf{q}}_{s1} = -\mathbf{J}_1^{-1}\boldsymbol{\tau}_1 + \mathbf{J}_1^{-1}\mathbf{u}_1 \tag{3.107}$$

$$\ddot{\mathbf{q}}_{s2} = -\mathbf{J}_2^{-1}\boldsymbol{\tau}_2 + \mathbf{J}_2^{-1}\mathbf{u}_2 \tag{3.108}$$

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v}$$

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2) \tag{3.109}$$

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}) + \mathbf{g}(\mathbf{z}, \mathbf{u}) \tag{3.110}$$

$$\dot{\mathbf{z}} = \begin{bmatrix} \mathbf{A}\mathbf{x} + \mathbf{B}(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2) \\ \mathbf{C}_2\mathbf{x}_{s1} \\ -\mathbf{J}_1^{-1}\boldsymbol{\tau}_1 - \mathbf{J}_1^{-1}\boldsymbol{u}_1 \\ \mathbf{C}_2\mathbf{x}_{s2} \\ -\mathbf{J}_2^{-1}\boldsymbol{\tau}_2 - \mathbf{J}_2^{-1}\boldsymbol{u}_2 \end{bmatrix} \tag{3.111}$$

$$\mathbf{g}(\mathbf{z}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{J}_1^{-1}\mathbf{u}_1 \\ \mathbf{0} \\ \mathbf{J}_2^{-1}\mathbf{u}_2 \end{bmatrix} \tag{3.112}$$

$$\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} \tag{3.113}$$

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} \mathbf{A}\mathbf{x} + \mathbf{B}(\boldsymbol{\sigma}_1 \mathrm{sinh}(\boldsymbol{a}_1(\mathbf{C}_1\mathbf{x}_{s1} - \mathbf{C}_1\mathbf{x})) + \mathbf{B}(\boldsymbol{\sigma}_2 \mathrm{sinh}(\boldsymbol{a}_2(\mathbf{C}_1\mathbf{x}_{s2} - \mathbf{C}_1\mathbf{x})) \\ \mathbf{C}_2\mathbf{x}_{s1} \\ -\mathbf{J}_1^{-1}(\boldsymbol{\sigma}_1 \mathrm{sinh}(\boldsymbol{a}_1(\mathbf{C}_1\mathbf{x}_{s1} - \mathbf{C}_1\mathbf{x}))) \\ \mathbf{C}_2\mathbf{x}_{s2} \\ -\mathbf{J}_2^{-1}(\boldsymbol{\sigma}_1 \mathrm{sinh}(\boldsymbol{a}_2(\mathbf{C}_1\mathbf{x}_{s2} - \mathbf{C}_1\mathbf{x}))) \end{bmatrix} \tag{3.114}$$

$$\mathbf{g}(\mathbf{z}) = \begin{bmatrix} 0 \\ 0 \\ \mathbf{J}_1^{-1}\mathbf{u}_1 \\ 0 \\ \mathbf{J}_2^{-1}\mathbf{u}_2 \end{bmatrix} \tag{3.115}$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{K}_1\mathbf{C}_1 - \mathbf{B}\mathbf{K}_2\mathbf{C}_1 & \mathbf{B}\mathbf{K}_1\mathbf{C}_1 & \mathbf{B}\mathbf{K}_2\mathbf{C}_1 \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \\ \mathbf{J}_1^{-1}\mathbf{K}_1\mathbf{C}_1 & -\mathbf{J}_1^{-1}\mathbf{K}_1\mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 \\ -\mathbf{J}_2^{-1}\mathbf{K}_2\mathbf{C}_1 & \mathbf{0} & -\mathbf{J}_2^{-1}\mathbf{K}_2\mathbf{C}_1 \end{bmatrix} \tag{3.116}$$

$$\tilde{\mathbf{B}} = \frac{\partial \mathbf{g}}{\partial \mathbf{u}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{J}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \tag{3.117}$$

Constraint dynamics for the VSA robot follows a similar formulation as was done in Section 3.11 is given by

$$\underbrace{\begin{bmatrix} \mathbf{F}_c & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dot{\mathbf{F}}_c & \mathbf{F}_c & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \mathbf{x}_{s1} \\ \mathbf{x}_{s2} \end{bmatrix}}_{\dot{\mathbf{z}}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.118}$$

.

$$\begin{cases} \dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} + \bar{\mathbf{F}}\,\lambda \\ \mathbf{G}\dot{\mathbf{z}} = \mathbf{0} \end{cases}, \quad \bar{\mathbf{F}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{H}^{-1}\mathbf{F}_c^{\mathrm{T}} \end{bmatrix} \tag{3.119}$$

$$\mathbf{u} = -\mathbf{K}_s(\mathbf{z} - \mathbf{z}_d) + \underline{\tilde{\mathbf{B}}^{+}(\dot{\mathbf{z}_d} - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{C}})} \tag{3.120}$$

$$\tilde{\mathbf{C}} = \dot{\mathbf{z}}_d - \tilde{\mathbf{A}}\mathbf{z}_d - \tilde{\mathbf{B}}\mathbf{u}_{\mathit{ff}} \tag{3.121}$$

$$\begin{bmatrix} \dot{\mathbf{z}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\bar{\mathbf{F}} \\ \mathbf{G} & \mathbf{0} \end{bmatrix}^{+} \begin{bmatrix} \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}} \\ \mathbf{0} \end{bmatrix} \tag{3.122}$$

Chapter 4

Implementation

4.1 Simulation

The implementation of the different control mechanisms was computed with a computer program in MATLAB and no part of this work was implemented on real hardware. The assumptions make in simulation , though ideal, is sufficient enough to evaluate control techniques developed. The simulation pipeline is to take an existing 3d model of a robot in URDF, STL, DAE format which describes the robot kinematics fully and use it to generate an analytical kinematic description of the robot. We then proceed to generate the dynamics of the robot using the euler-lagrange equation of motion formulation.

4.2 Experiment on Three-Link Manipulator(RRR)

Computed torque Controller (CTC) with QR decomposition for Inverse dynamics and Constrained Linear Quadratic Regulator control experiments were done on three link (RRR) manipulator. The reason for this is to have a base line experiment to test control laws ans ascertain control objectives are met. Three link manipulator has special characteristics of being fully-actuated, absence of close-loop kinematics and a non-floating base systems. These characteristics are what we have in a typical quadruped robots that makes it difficult to control and have highly complex non-linear dynamics.

For the rigid joint RRR manipulator, 100 different initial positions were specified sampled from a gaussian distribution. The same task were defined for all initial positions and the associated costs with different controller types were compared together. This allows us to compare effect of initial positions on trajectory tracking of robot state.

The initial positions is sampled from a gaussian distribution with various standard deviations and the result obtained for each deviation is compared.

$$x(0) = x_0 + \sigma \tag{4.1}$$

$$z \sim \mathcal{N}(x_0 + \sigma) \tag{4.2}$$

4.3 Experiment on Cheetah quadruped robot model

The MIT cheetah robot model as described in [29] and [30] was used to test and simulate control experiments. The robot was tested for different tasks while ensuring it remains compliant throughout the task trajectory. CTC and CLQR were experimented with accordingly. All experiments were carried out in MATLAB making use of the quadruped robot model and deriving the robot

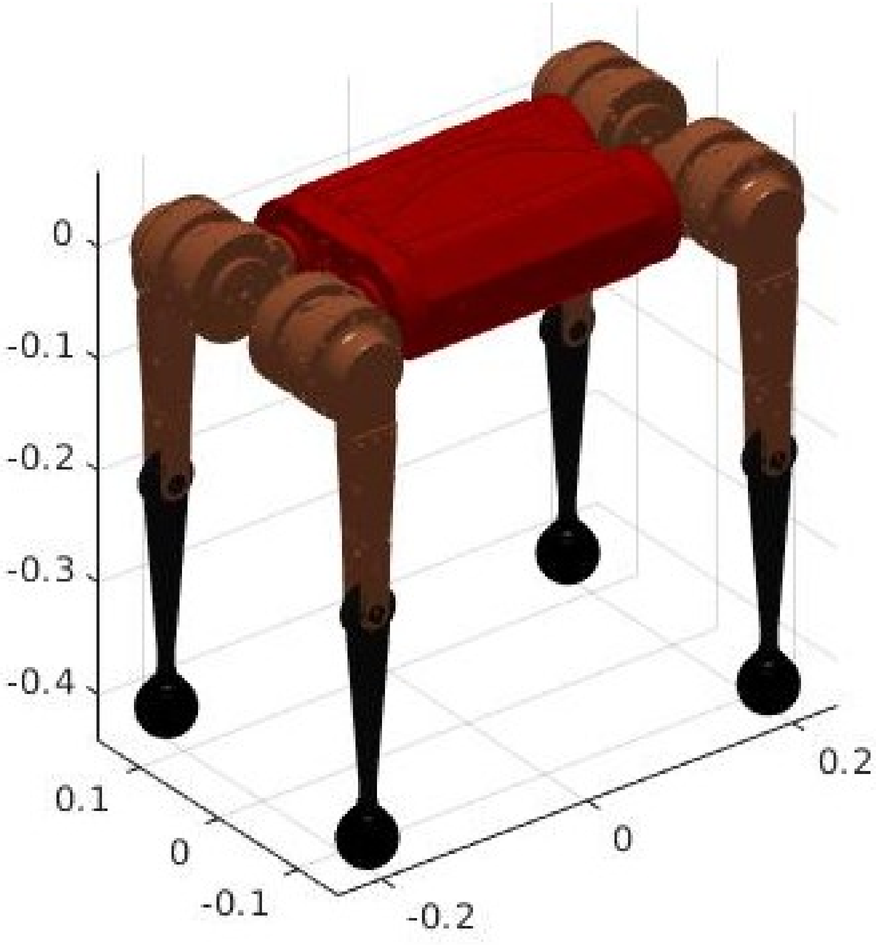

Figure 4.1: Cheetah quadruped model

description forward kinematics and jacobians and robot dynamics symbolically.

Effect of different initial positions on costs and tracking performance was also analyzed for the cheetah quadruped robot. Initial positions are samples taken from a gaussian distributions with different standard deviations ranging from 0.00001 to 0.1.

Chapter 5

Evaluation and Discussion

5.1 Results

The control CTC and CLQR control experiments were implemented on a three link manipulator and the 18 degree of freedom cheetah robot. The performance while robot was starting from different initial positions sampled from a Gaussian distribution of different deviations was evaluated on a cost function defined as:

$$\mathbf{x}_{cost} = max \mid\mid \mathbf{x}(t_i) - \mathbf{x}^*(t_i) \mid\mid \tag{5.1}$$

$$\mathbf{u}_{cost} = max \mid\mid \mathbf{u}_i - \mathbf{u}_i^* \mid\mid = max \mid\mid \mathbf{K}(\mathbf{x}(t_i) - \mathbf{x}^*(t_i)) \mid\mid \tag{5.2}$$

where $\mathbf{x}$ and $\mathbf{x}^*$ denotes actual joint positions and desired joint positions, and $\mathbf{K}$ denotes the feedback control gain.

5.1.1 Result on Three-Link Manipulator

The result presented in Table I and II show the cost expressed in (5.1) and (5.2) respectively. Even though CTC controller was able to track our desired trajectory, it was doing it at much more cost which we desire to minimize and the actuators are extending more torques to achieve desired motion while CLQR by its nature finds an optimal gain to track desired trajectory as previously discussed in previous chapters.

Table I: x_{cost} on three link manipulator

Sigma	CTC	CLQR
0.0001	3.2972	4.2956
0.001	8.2837	4.408
0.01	35.39	9.2947
0.1	263.87	57.833

Table II: u_{cost} on three link manipulator

Sigma	CTC	CLQR
0.0001	30.696	4.2996
0.001	30.081	4.3216
0.01	25.339	4.9306
0.1	204.62	21.123

5.1.2 Result on Cheetah Quadruped robot

The results in Table III and IV show the numeric values of costs on position and actuator torques respectively. It can be seen in both results that, the cost due to using the CLQR controller is lower than CTC which is expected since CLQR finds an optimal feedback gain to track the desired trajectory. As the standard deviation sigma(σ) of the initial position sample distribution increases, the cost increases and much more deviation is observed between desired trajectory and the actual trajectory due to control action.

Fig. 5.2, 5.3, 5.4 show results obtained on the cheetah robot with rigid joints, SEA joint and VSA joint respectively. Fig. 5.5 and Fig. 5.6 shows the

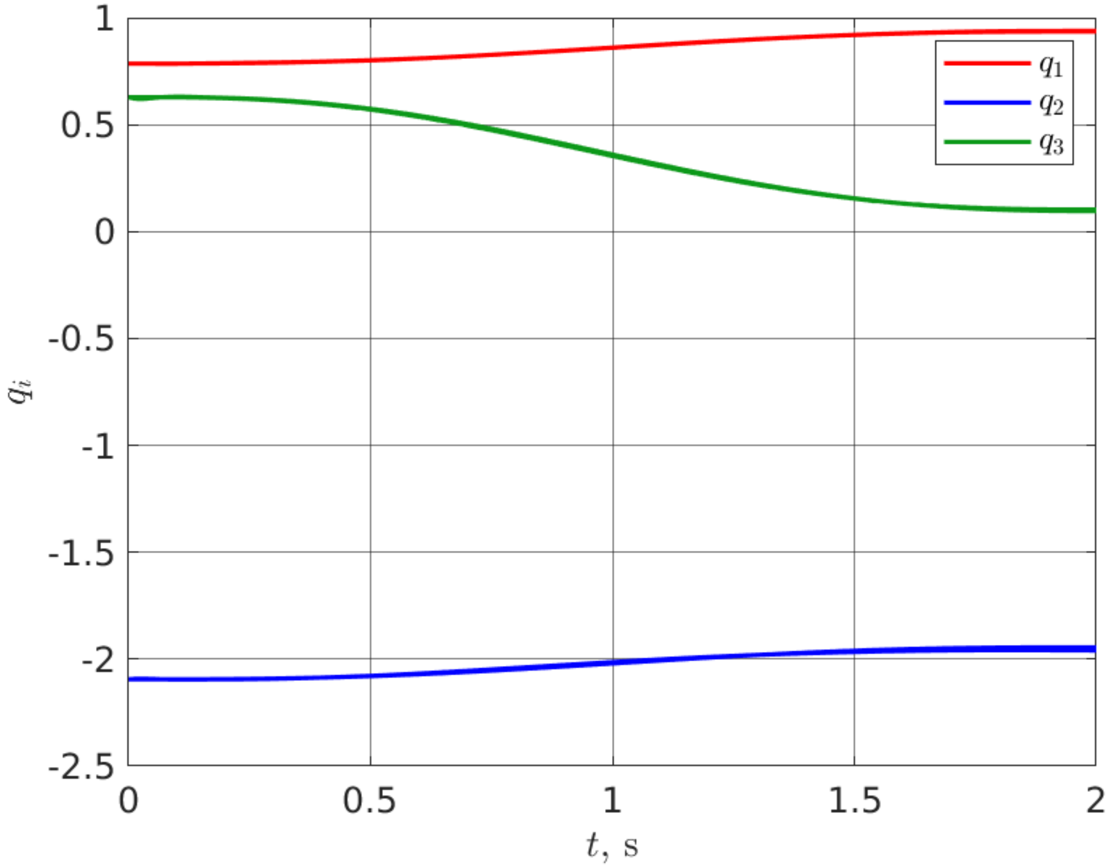

Figure 5.1: Trajectory of the RRR manipulator

initial and final pose of the robot in animation.

Table III: x_{cost} on cheetah quadruped

Sigma	CTC	CLQR
0.0001	0.11991	0.14218
0.001	0.12275	0.14683
0.01	0.21808	0.18284
0.1	2.2156	2.6047

...

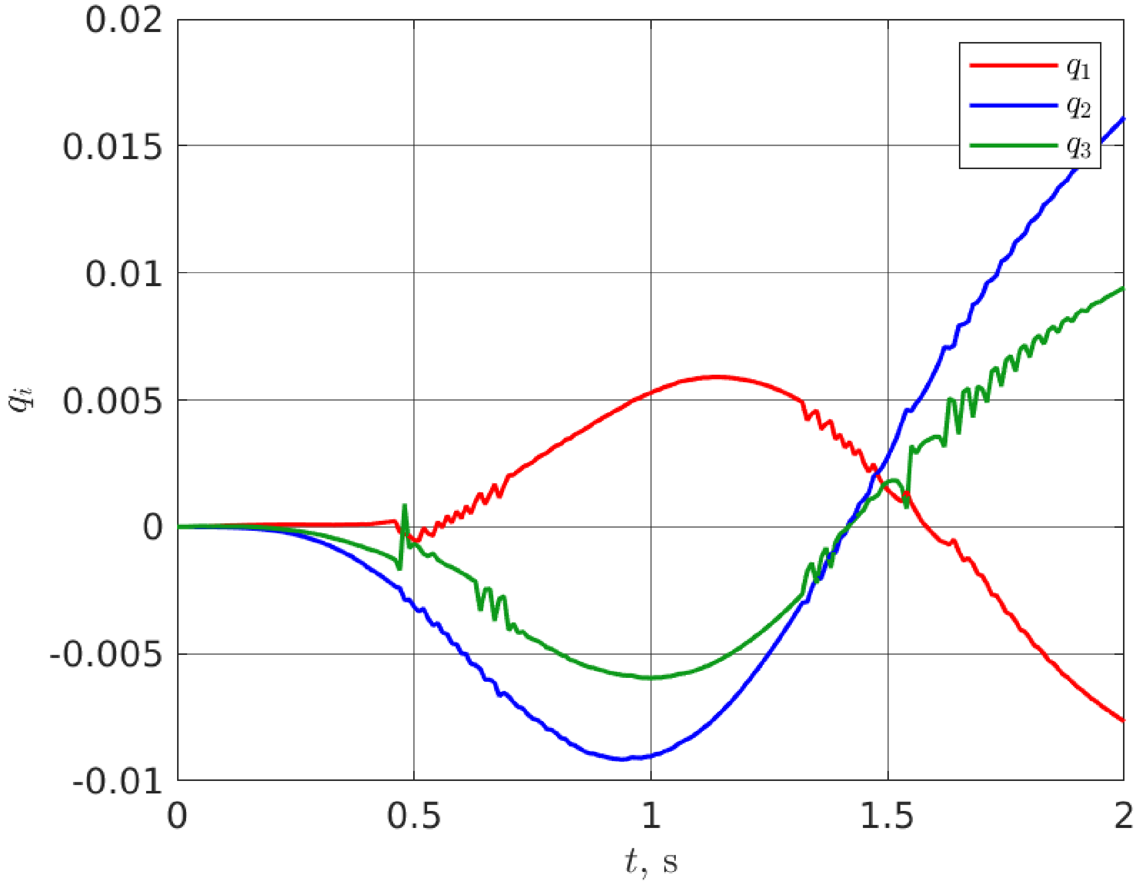

Figure 5.2: Position error on rigid joint

Table IV: u_{cost} on cheetah quadruped

Sigma	CTC	CLQR
0.0001	52.974	16.013
0.001	53.144	15.985
0.01	91.30	16.013
0.1	291.75	20.001

5.2 Discussion

We are able to achieve compliant trajectory control of a quadruped robot using CTC and CLQR control technique. These approaches were able to stabilize and track the position of the robot joints. Linearized dynamics of SEA and VSA were equally derived in terms of the linearized form of the rigid robot.

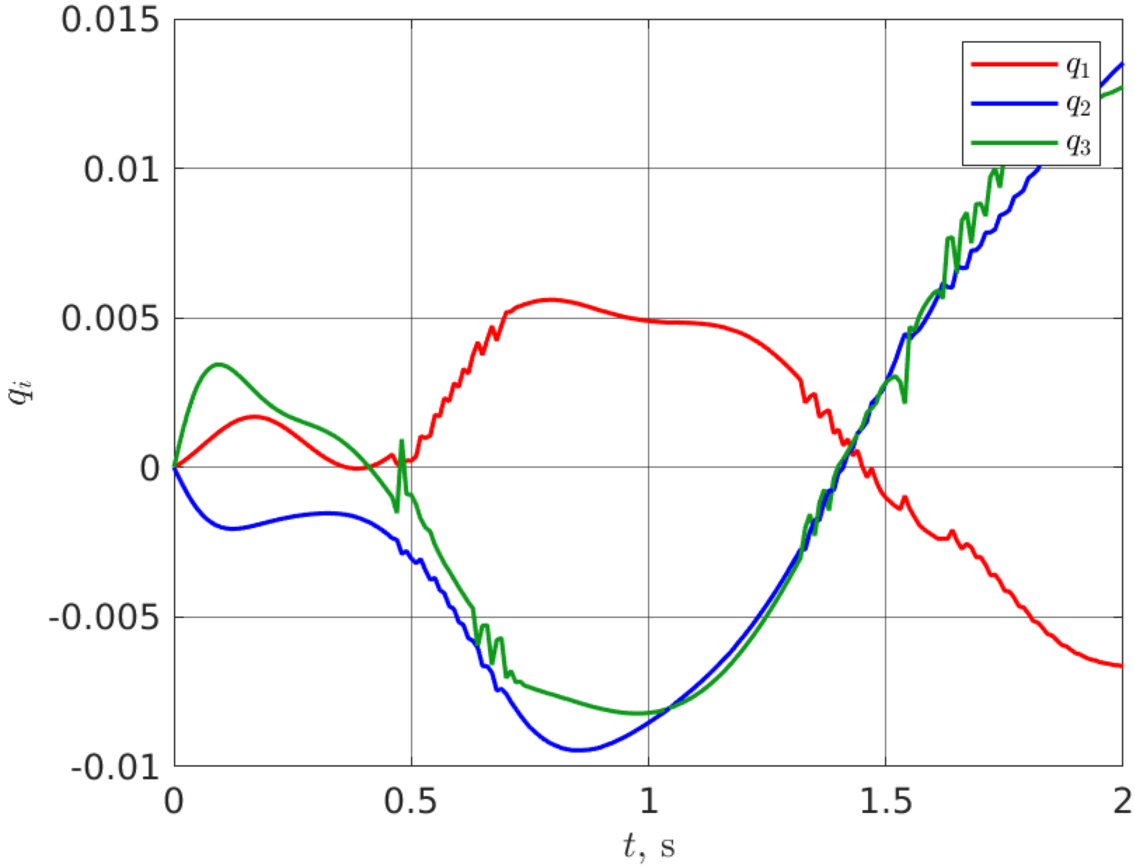

Figure 5.3: Position error due to SEA

Using the linearized dynamics definition for the iterative LQR technique and the kinematic jacobian of the constraint allows to track the trajectory. In the rigid manipulator experiment, we can see that the initial position of the robot indeed has an effect on how well out controller is able to track desired trajectory. It is very important that initial position, the robot satisfies all the kinematic constraint. From the result, we see that as initial position drifts more from the true initial position that satisfies kinematic and contact constraint, the cost increases.

Furthermore, the results from the SEA and VSA robots shows us that indeed compliance can be introduced to robotic system and legged robots through elastic elements. Using the linearized dynamics representation and LQR control

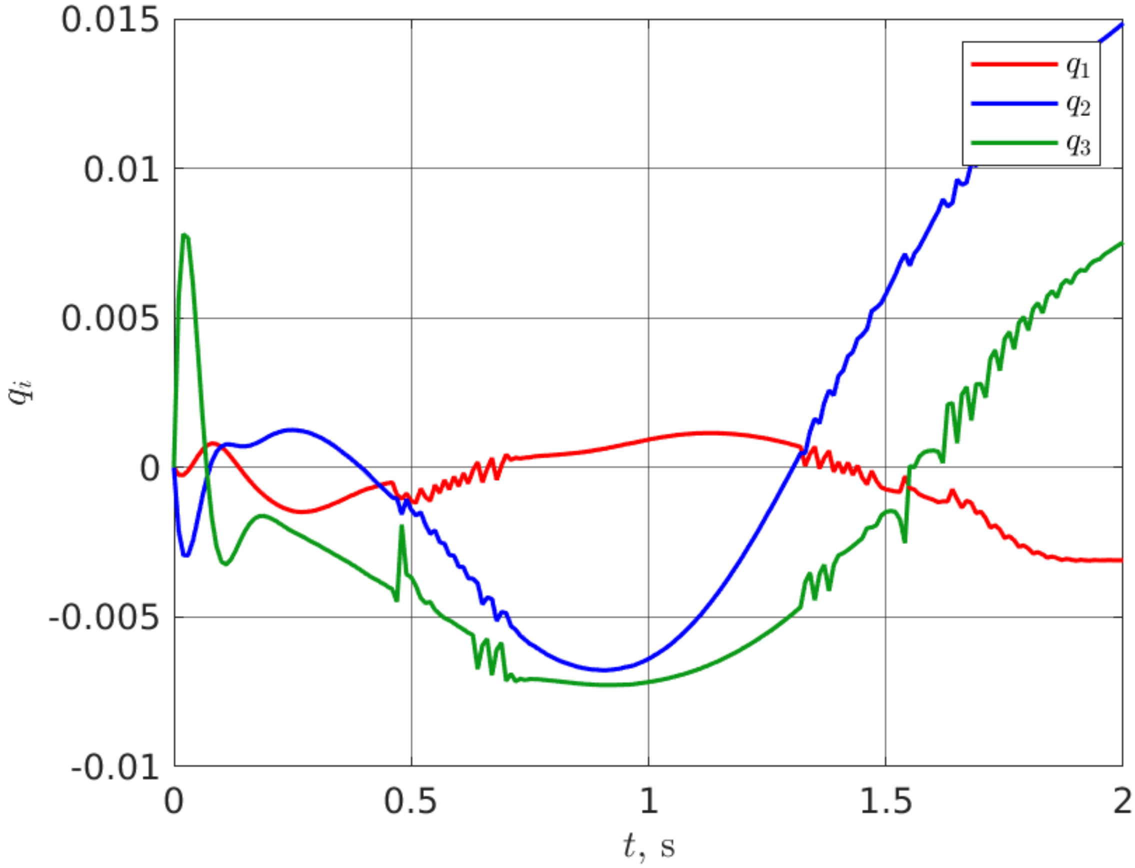

Figure 5.4: Position error due to VSA

law, we are able to track and stabilize the desired trajectory.

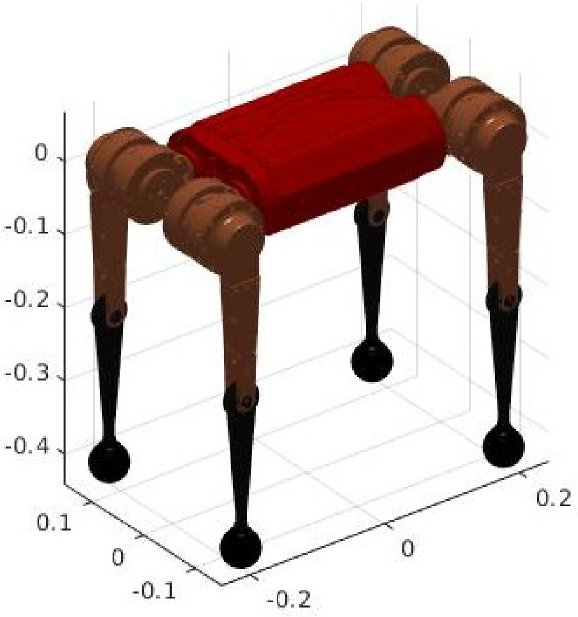

Figure 5.5: Initial pose

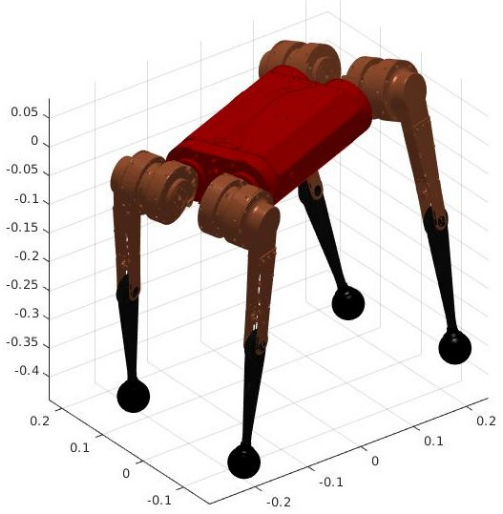

Figure 5.6: final pose

Chapter 6

Conclusion

Some limitations still exist to this work which would be further worked on within the research group. First, the control law derived from the linearized robot dynamics was applied on the linearized dynamics. This approach is not very practical and would give too accurate position tracking. Since out robotic system is ideally non-linear, the usefulness of the linearized dynamics is only to derive a linear controller that would be applied on non-linear dynamics and not the linearized dynamics. However, there is currently no computation tool, to the best of my knowledge, that allow us to take a robot model, derive its non linear-dynamics from the model, derive linearized dynamics, derive a linear controller and apply the linear controller on the non-linear system. ROS is the closest software that exists but it doesn't give the flexibility to carry out multiple experiments on different robots using the same controller.

Secondly, this work has only experimented with one out of several feed-foward dynamics that could have been experimented with for the SEA and VSA robot. A comparison of feed-forward laws need to be done and the most appropriate selected based on its performance.

Overall, in this work we are able to see that contact points are not enemies in robotic systems but necessary elements to ensure compliance of legged robots. Sufficient number of points must be in contact with the ground to cancel out the under-actuation introduced by the floating-base dynamics. Moreover, elastic actuators with constant and varying stiffness help to introduce compliance naturally into our system and designing controllers like CLQR can help us track desired trajectory and overall ensure stability of the quadruped robot

Bibliography cited

[1] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *2009 IEEE/RSJ international conference on Intelligent robots and systems*, IEEE, 2009, pp. 814–820.

[2] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 814–820. DOI: 10.1109/IROS.2009.5354681.

[3] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *2010 IEEE international conference on robotics and automation*, IEEE, 2010, pp. 3406–3412.

[4] G. Buondonno and A. De Luca, "Efficient computation of inverse dynamics and feedback linearization for vsa-based robots," *IEEE robotics and automation letters*, vol. 1, no. 2, pp. 908–915, 2016.

[5] F. Petit, "Analysis and control of variable stiffness robots," Ph.D. dissertation, ETH-Zürich, 2014.

[6] F. Petit, W. Friedl, H. Höppner, and M. Grebenstein, "Analysis and synthesis of the bidirectional antagonistic variable stiffness mechanism," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 684–695, 2014.

[7] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 93–100.

[8] M. Focchi, E. Guglielmino, C. Semini, T. Boaventura, Y. Yang, and D. G. Caldwell, "Control of a hydraulically-actuated quadruped robot leg," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4182–4188. DOI: 10.1109/ROBOT.2010.5509697.

[9] S. Mason, L. Righetti, and S. Schaal, "Full dynamics lqr control of a humanoid robot: An experimental study on balancing and squatting," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 374–379. DOI: 10.1109/HUMANOIDS.2014.7041387.

[10] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 1085–1090.

[11] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[12] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *2011 11th IEEE-*

RAS International Conference on Humanoid Robots, IEEE, 2011, pp. 26–33.

[13] M. Hutter, M. A. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart, "Hybrid operational space control for compliant legged systems," *Robotics*, p. 129, 2013.

[14] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, K. Kaneko, M. Morisawa, E. Yoshida, and F. Kanehiro, "Vertical ladder climbing by the hrp-2 humanoid robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 671–676.

[15] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 1283–1290.

[16] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.

[17] R. Tedrake, M. F. Fallon, S. Karumanchi, S. Kuindersma, M. E. Antone, T. Schneider, T. M. Howard, M. R. Walter, H. Dai, R. Deits, *et al.*, "A summary of team mit's approach to the virtual robotics challenge.," in *ICRA*, 2014, p. 2087.

[18] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 1028–1035.

[19] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *2010 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2010, pp. 1248–1255.

[20] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[21] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3406–3412. DOI: 10.1109/ROBOT.2010.5509646.

[22] S. Mason, N. Rotella, S. Schaal, and L. Righetti, "Balancing and walking using full dynamics lqr control with contact constraints," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2016, pp. 63–68.

[23] O. Stasse and T. Flayols, "An overview of humanoid robots technologies," *Biomechanics of Anthropomorphic Systems*, pp. 281–310, 2019.

[24] R. Schiavi, G. Grioli, S. Sen, and A. Bicchi, "Vsa-ii: A novel prototype of variable stiffness actuator for safe and performing robots interacting with humans," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, 2008, pp. 2171–2176.

[25] B. Z. Lukić, K. M. Jovanović, and G. S. Kvaščcev, "Feedforward neural network for controlling qbmove maker pro variable stiffness actuator," in *2016 13th Symposium on Neural Networks and Applications (NEUREL)*, 2016, pp. 1–4. DOI: 10.1109/NEUREL.2016.7800116.

[26] W. Wang, R. N. Loh, and E. Y. Gu, "Passive compliance versus active compliance in robot-based automated assembly systems," *Industrial Robot: An International Journal*, 1998.

[27] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[28] S. Savin and L. Vorochaeva, "Nested quadratic programming-based controller for in-pipe robots," in *Proceeding of the International Conference On Industrial Engineering*, 2017.

[29] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2245–2252.

[30] S. Rutishauser, "Cheetah: Compliant quadruped robot," *Biologically Inspired Robotic Group*, 2008.