School of Engineering, Discipline of Mechanical Engineering and Mechatronics, pp. 1–26

Final Year Project Part A

May 2022

Darcy Byrne [1]

[1] *Student of Mechatronics Engineering,*
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA
Student Number: 3256634
E-mail: D.Byrne@uon.edu.au

Abstract

The intention of this particular, half finished, report is to convey the volume of work I have completed this semester, in order to prove my elligiblity to continue my Final Year Project in semester two.
Despite blank sections, such as background, it can be seen from the figures presented that I have a firm understanding as to bipedal locomotion using Quasi Static and ZMP walking strategies.
Given the work I have completed thus far I am interested in exploring a Spring Loaded Inverted Pendulum (SLIP) Model of walking. This may or may not address the 'springy' of the NuGus with 3D printed legs, however, I feel developing a sim of such a model is a natural progression from ZMP and for myself.
I concede my ZMP implementation is a little rushed, somewhat 'cart before the horse', and not lacking a few logic errors, but it is close. I intend to refactor all the matlab script I have written to date.

Acknowledgements

Thank you Joel for supporting me to bite off more than I could chew last year, and guiding me back to a logical progression of knowledge for this year's FYP.

Citations for self reference ... Have yet to drop them anywhere in this text
[4] - A realtime pattern generator for biped walking
[3] - ZMP
[5] - Preview Control Discrete
[7] - Fundamentals of Mechanical Systems Vec Calc etc.
[9] - MCHA 4100
[1] - Robotics: Modelling Planning Control
[8] - Legged Robots that balance
[6] - The WABOT-1 -¿ First known Quasi Static Biped ?
[10] - Is Dynamic Control Needed in Robotic Systems ?
[2] - Design and Control of a Quadruped Walking Vehicle ?

Contents

List of Figures

List of Tables

1. Introduction

NuBots require more Mechatronics Students for the long march to a team of functioning robots. NuBots have not excelled internationally for many years, and continue to struggle with the locomotion problem.
I have no experience solving the locomotion problem, despite my interest in it. In order to accelerate developement of walking solutions, an interface between matlab and a NuGus, allowing the running of open loop trajectories and joint variables in realtime, would be greatly appreciated. Additionally, if a solution to the lack of rigidity of the 3D printed legs could make use of their elasticity, the NuBots team could field its full team of robots.
Developing my understanding of the locomotion problem, by simulating well known solutions, will make me an asset to the team.

2. Background

2.1. Mathematical Notation

2.2. Numerical Optimisation

2.3. Kinematics

2.4. Kinematic Chains

2.5. Walking

3. Models

Each model is a simplistic representation of a bipedal robot. As the focus is simulating different walking strategies, each model is comprised of the lower half of a humanoid, that being the feet, legs and waist. Both models were constructed via a series of homogenous transforms that describe the postions of each ankle in global coordinates.

3.1. 2D Model

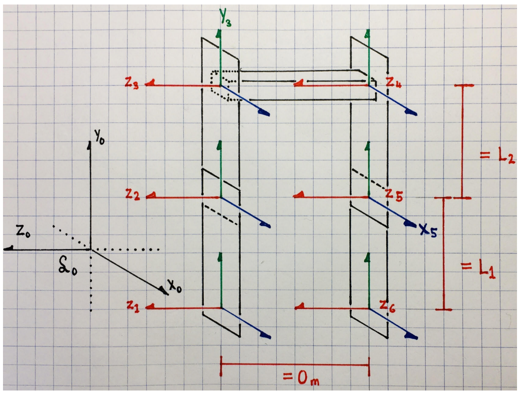


Figure 1: Sketch defining coordinate systems **0** to **6**.

Beginning with a sketch, seen in Figure 1, the following series of Homogenous Transforms, equation(3.1) forms a kinematic chain that describes the position of the right ankle, the end effector, with respect to the left ankle.

$$\mathbf{T}_6^1 = \mathbf{A}_2^1(\phi_1, +L_1)\mathbf{A}_3^2(\phi_2, +L_2)\mathbf{A}_4^3(\phi_3, -H)\mathbf{A}_5^4(\phi_4, -L_2)\mathbf{A}_6^5(\phi_5, -L_1)\mathbf{A}^6(\phi_6) \tag{3.1}$$

Subsequently, the kinematic chain from the right ankle to the postion of the left ankle, the end effector when the right ankle is fixed, is equation(3.2):

$$\mathbf{T}_1^6 = (\mathbf{T}_6^1)^{-1} \tag{3.2}$$

An additonal homogenous transform was applied to each equation to express the end effectors in global coordinates.

$$\begin{aligned} \mathbf{T}_6^{Global} &= \mathbf{A}_1^0\mathbf{T}_6^1 \\ \mathbf{T}_1^{Global} &= \mathbf{A}_6^0\mathbf{T}_1^6 \end{aligned} \tag{3.3}$$

...where;

$$\mathbf{A}_1^0 = \begin{bmatrix} \mathbf{I} & \dots & \dots & \text{LeftAnkle}_x \\ \vdots & \ddots & \dots & \text{LeftAnkle}_y \\ \vdots & \vdots & \ddots & \text{LeftAnkle}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

... and;

$$\mathbf{A}_6^0 = \begin{bmatrix} \mathbf{I} & \dots & \dots & \text{RightAnkle}_x \\ \vdots & \ddots & \dots & \text{RightAnkle}_y \\ \vdots & \vdots & \ddots & \text{RightAnkle}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Equations(3.3) are the Foward kinematic Model (FKM) which were then written in Matlab, with the joint angles found in table 1 and the parameters found in table 2.

Table 1: Joint Angles for 2D model

Joint Angles		
Joint	Angle ($rads$)	Variable Name
1	$-\frac{\pi}{6}$	ϕ_1
2	$+\frac{2\pi}{6}$	ϕ_2
3	$-\frac{\pi}{6}$	ϕ_3
4	$+\frac{\pi}{6}$	ϕ_4
5	$-\frac{2\pi}{6}$	ϕ_5
6	$+\frac{\pi}{6}$	ϕ_6

Table 2: Link Lengths for 2D model

Link Lengths			
Link	Length (m)	Variable Name	Joints
2	0.4	L_1	1 to 2
3	0.4	L_2	2 to 3
4	0.0, however, ≈ 0.2	H	3 to 4
5	0.4	L_2	4 to 5
6	0.4	L_1	5 to 6

The model was then plotted in both 2D and 3D (figure 2). Although 2D, plotting the model in 3D with an arbitrary distance between the hip joints, 3 and 4, provided more context regarding the location and orientation of the right side (blue crosses). Additonally, plotting the 2D model in 3D reduced the workload for developing the 3D model as a vast proportion of the matlab script code be reused.

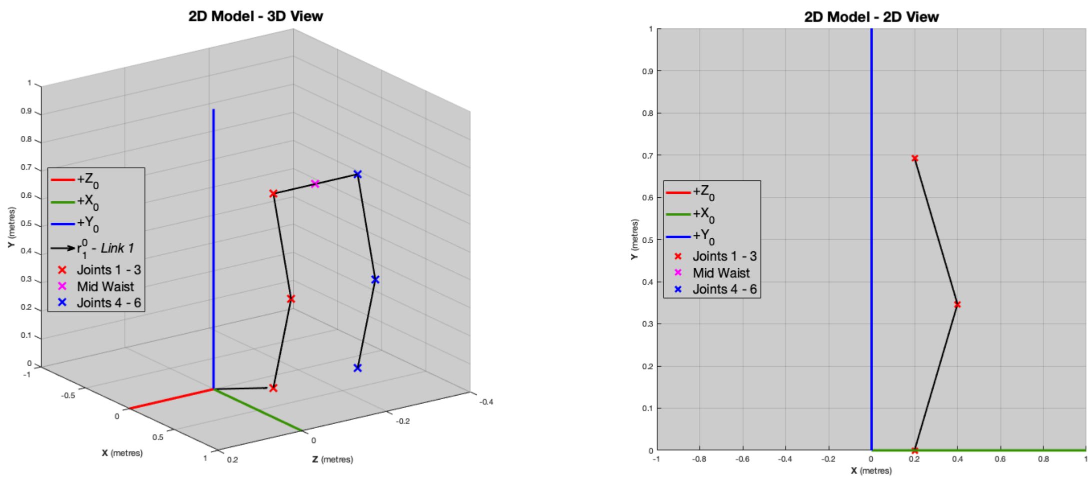


Figure 2: 2D model plotted in 3D (left) and 2D (right).

3.2. 3D Model

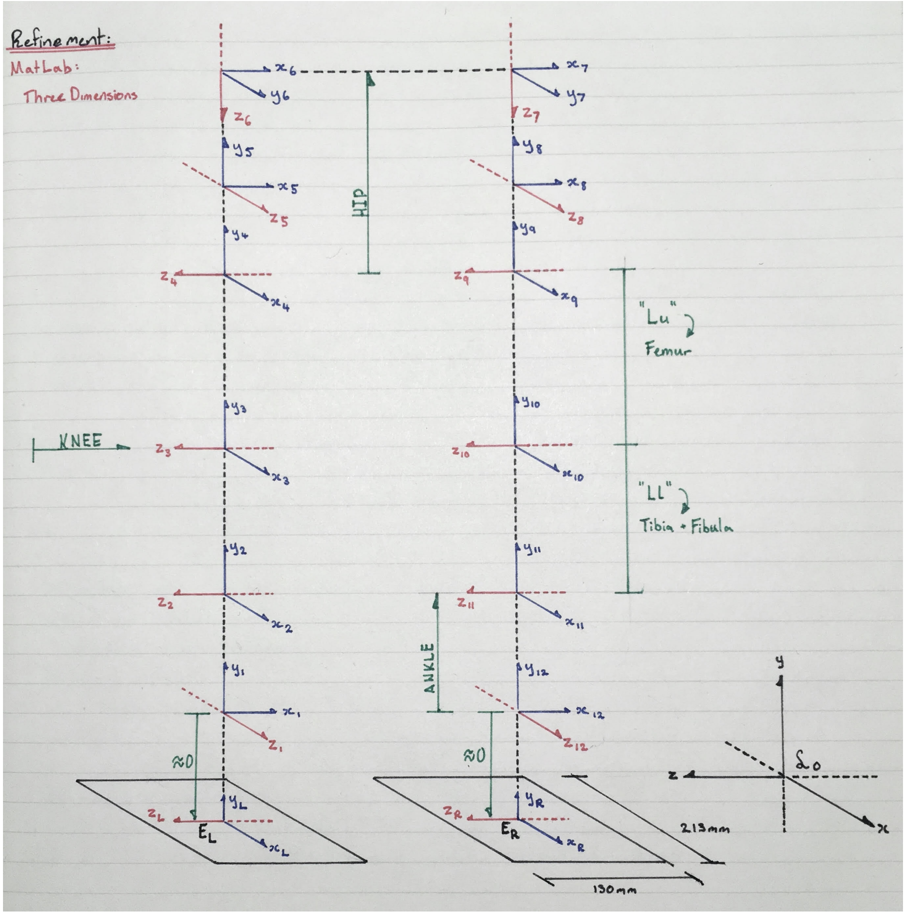


Figure 3: Sketch defining coordinate system **0**,
and coordinate systems $\mathbf{E_L}$ through to $\mathbf{E_R}$.

Beginning with a sketch, seen in Figure 3, the following series of Homogenous Transforms, equation(3.6) forms a kinematic chain that describes the position of the right foot, the end effector, with respect to the left foot.

$$\begin{aligned}\mathbf{T}_{E_R}^{E_L} =& \mathbf{A}_1^{E_L}\mathbf{A}_2^1(\phi_1)\mathbf{A}_3^2(\phi_2, +L_{lower})\mathbf{A}_4^3(\phi_3, +L_{upper})\mathbf{A}_5^4(\phi_4)\mathbf{A}_6^5(\phi_5)\mathbf{A}_7^6(\phi_6)\\ &\dots \mathbf{A}_8^7(\phi_7)\mathbf{A}_9^8(\phi_8)\mathbf{A}_{10}^9(\phi_9, -L_{upper})\mathbf{A}_{11}^{10}(\phi_{10}, -L_{lower})\mathbf{A}_{12}^{11}(\phi_{11})\mathbf{A}_{E_R}^{12}(\phi_{12})\end{aligned} \tag{3.6}$$

Subsequently, the kinematic chain from the right foot to the postion of the left foot, the end effector when the right foot is fixed, is equation(3.7):

$$\mathbf{T}_{E_L}^{E_R} = (\mathbf{T}_{E_R}^{E_L})^{-1} \tag{3.7}$$

An additonal homogenous transform was applied to each equation to express the end effectors in global coordinates.

$$\begin{aligned}\mathbf{T}_{E_R}^{Global} &= \mathbf{A}_{E_L}^0\mathbf{T}_{E_R}^{E_L}\\ \mathbf{T}_{E_L}^{Global} &= \mathbf{A}_{E_R}^0\mathbf{T}_{E_L}^{E_R}\end{aligned} \tag{3.8}$$

...where;

$$\mathbf{A}_{E_L}^{0} = \begin{bmatrix} \mathbf{R}(\boldsymbol{\Theta})_{E_L}^{0} & \mathbf{r}_{E_L}^{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.9}$$

... and;

$$\mathbf{A}_{E_R}^{0} = \begin{bmatrix} \mathbf{R}(\boldsymbol{\Theta})_{E_R}^{0} & \mathbf{r}_{E_R}^{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.10}$$

Equations(3.8) are the FKM which were then written in Matlab, with the joint angles found in table 3 and the parameters found in table 4.

Table 3: Joint Angles for 3D model

Joint Angles		
Joint	Angle ($rads$)	Variable Name
1	0	ϕ_1
2	$-\frac{\pi}{6}$	ϕ_2
3	$+\frac{2\pi}{6}$	ϕ_3
4	$-\frac{\pi}{6}$	ϕ_4
5	0	ϕ_5
6	0	ϕ_6
7	0	ϕ_7
8	0	ϕ_8
9	$+\frac{\pi}{6}$	ϕ_9
10	$-\frac{2\pi}{6}$	ϕ_{10}
11	$+\frac{\pi}{6}$	ϕ_{11}
12	0	ϕ_{12}

Table 4: Link Lengths for 3D model

Link Lengths			
Link	Length (m)	Variable Name	Joints
3	0.4	L_{lower}	3 to 4
4	0.4	L_{upper}	4 to 5
6	0.2	H	6 to 7
9	0.4	L_{upper}	9 to 10
10	0.4	L_{lower}	10 to 11

The model was then plotted in 3D (figure 4). Distances between joints at the ankle and hips represent physical displacements between axes found on most dual servo motors. Additionally, these distances provide context as to the movement of the sets of joints that make up the hip joint and the ankle joint.

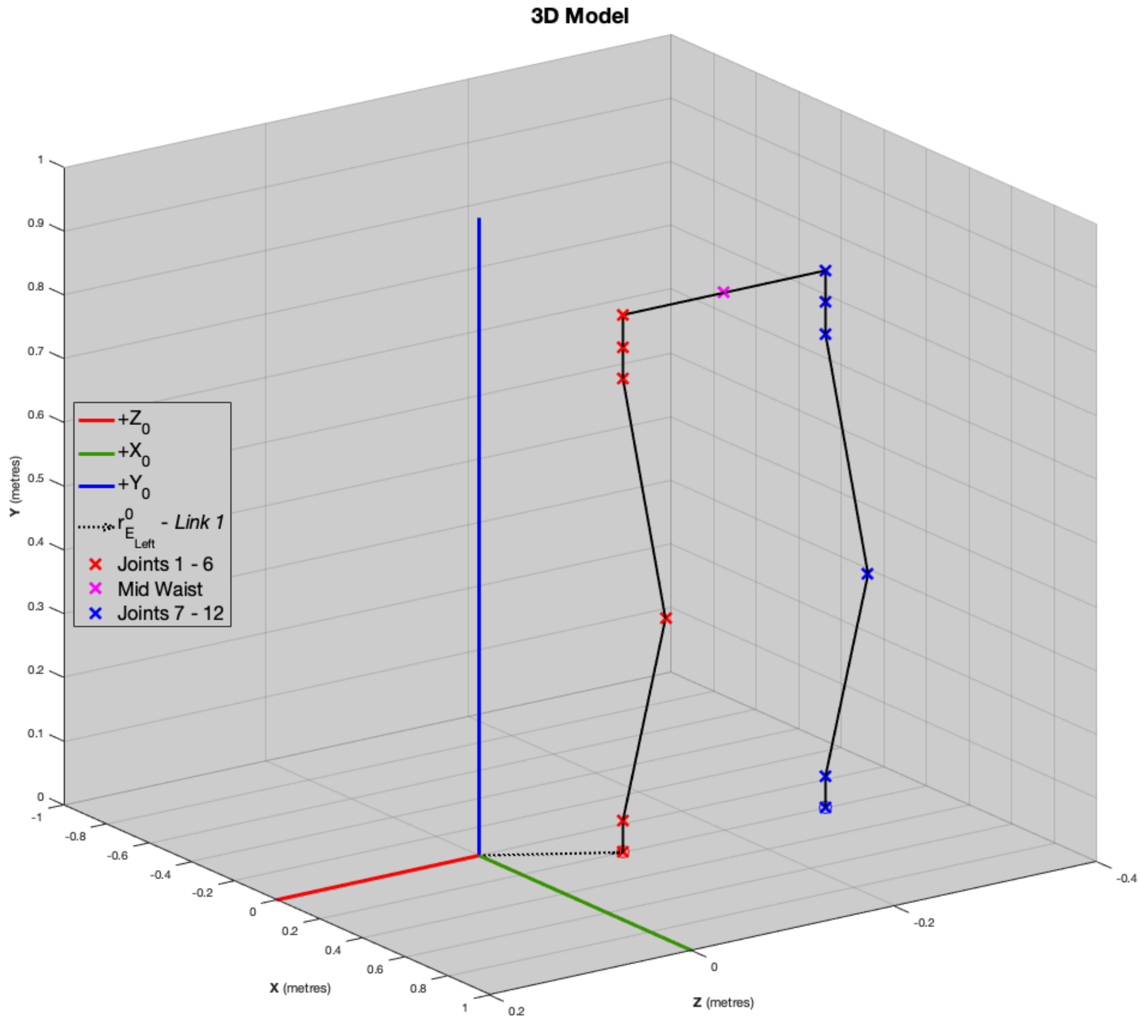


Figure 4: 3D model plotted using FKM.

4. Quasi Static Locomotion

4.1. Quasi Static concept

The core idea of Quasi-Static locomotion is that the centre of mass (CoM) remains over the support polygon (SP) for all time. This is achieved using the FKM, Inverse Kinematic Model (IKM), Trajectory Generation and Numerical Optimisation.
For both the 2D and the 3D model the same numerical optimisation functions were used. Equation(4.1) is recursively optimised while one foot is fixed and the other moves forward to the next step. During this phase, the FKM and IKM control the movement of the end effector, that being the foot in motion.

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \ \mathbf{K}_q\|\mathbf{q_0} - \mathbf{q}\| \ + \ \mathbf{K}_e\|k(\mathbf{q}) - \mathbf{x}_e\| \ + \ \mathbf{K}_{CoM}\|\mathbf{r}^0_{CoM} - \mathbf{r}^0_{\text{Fixed}}\| \tag{4.1}$$

Equation(4.2) is recursively optimised during the double support phase, moving the CoM from the trailing foot/support polygon, to the forward foot. Throughout this phase the CoM remains over the support polygon. During this phase the FKM and IKM form a parallel kinematic chain, with both controlling the mid waist.

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \ \mathbf{K}_q\|\mathbf{q_0} - \mathbf{q}\| \ + \ \mathbf{K}_e\|k(\mathbf{q})\| \ + \ \mathbf{K}_{CoM}\|\mathbf{r}^0_{CoM} - \mathbf{r}^0_{\text{Forward foot}}\| \tag{4.2}$$

Thus, during this phase the FKM takes the form in equation(4.3):

$$\mathbf{k}(\mathbf{q}) = \mathbf{A}^0_{E_L}\mathbf{T}^{E_L}_{\text{MidWaist}}(\mathbf{q}) - \mathbf{A}^0_{E_R}\mathbf{T}^{E_R}_{\text{MidWaist}}(\mathbf{q}) = \mathbf{0} \tag{4.3}$$

Trajectory Generation -¿ MCHA4100 -¿ Same Trajectories for both models

4.2. 2D model implementation

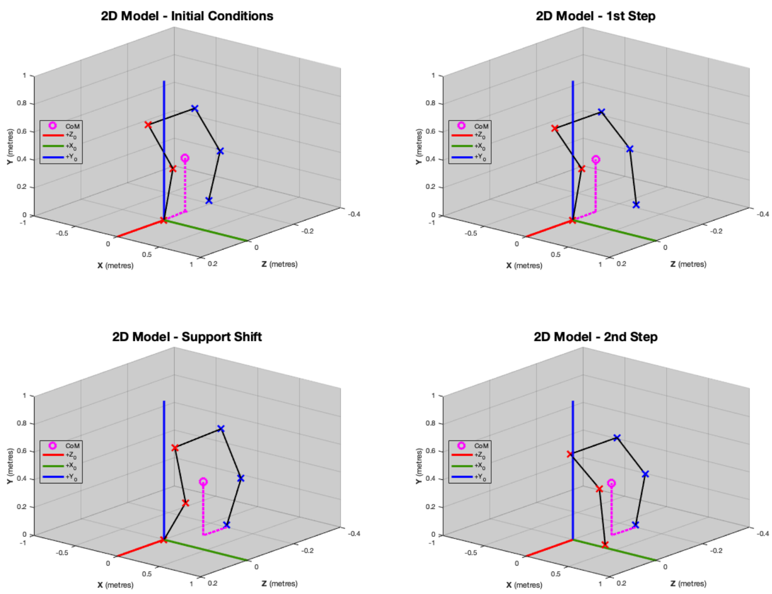


Figure 5: 2D model, plotted in 3D, Quasi Static walk sequence.

4.3. 3D model implementation

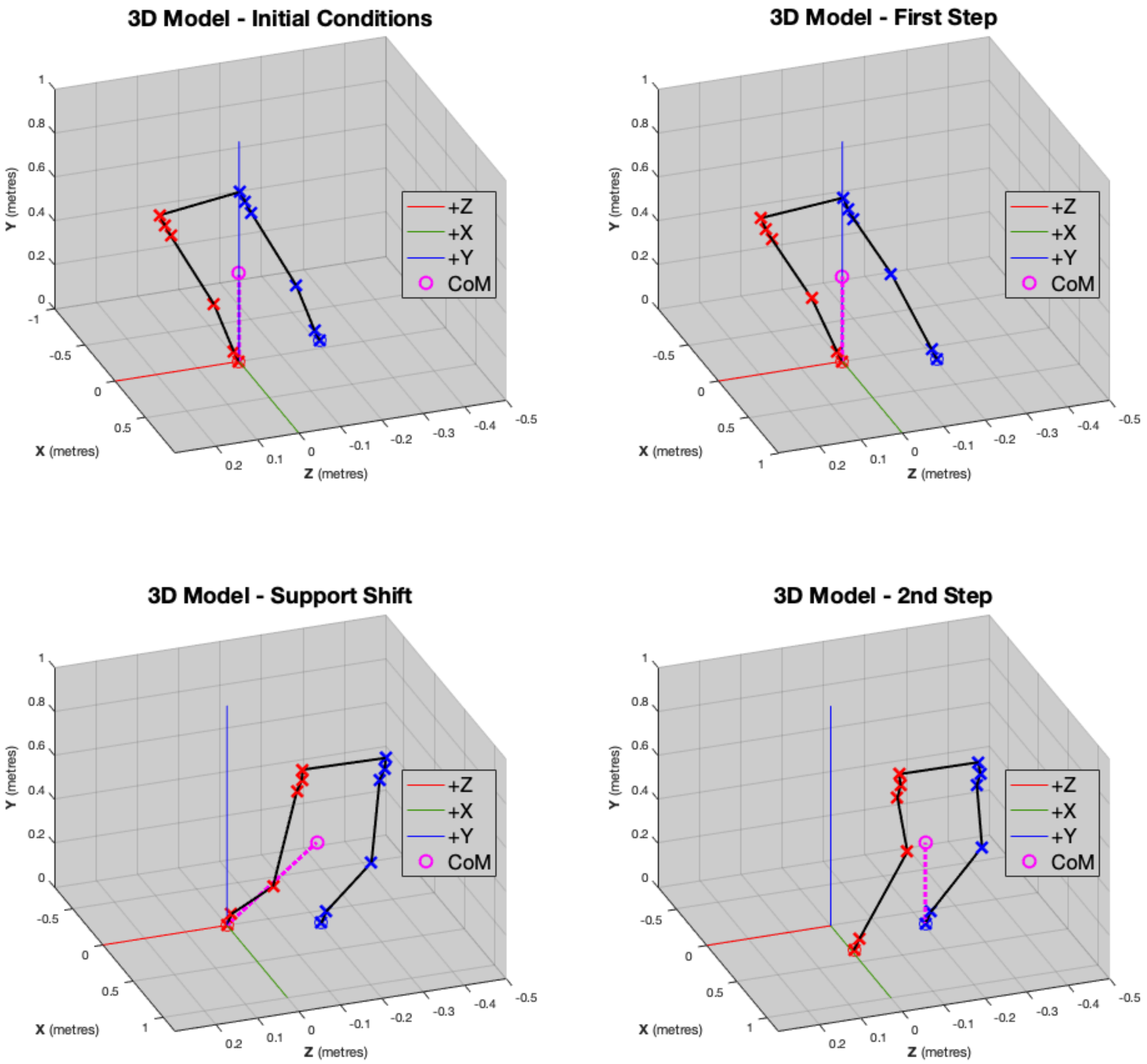


Figure 6: 3D model Quasi-Static walk sequence.

5. Zero Moment Point Locomotion

5.1. Zero Moment Point concept

5.2. 3D Linear Inverted Pendulum

$$\begin{aligned} \frac{\delta}{\delta t}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_{(x,y)} \\ \mathbf{p}_{(x,y)} &= \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \end{aligned} \tag{5.1}$$

$$\begin{aligned} \mathbf{x}_{(k+1)} &= \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{u}_k \\ \mathbf{p}_k &= \mathbf{C_d}\mathbf{x}_k \end{aligned} \tag{5.2}$$

... where:

$$\mathbf{A_d} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \ \mathbf{B_d} = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \ \mathbf{C_d} = \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} \end{bmatrix} \tag{5.3}$$

5.3. Preview Control

$$J_u = \sum_{i=k}^{\infty} \mathbf{Q}_e\|\mathbf{y}(k) - \mathbf{y_d}(k)\| \ + \ \mathbf{Q}_x\|\mathbf{x}(k) - \mathbf{x}(k-1)\| \ + \ \mathbf{R}\|\mathbf{u}(k) - \mathbf{u}(k-1)\| \tag{5.4}$$

$$\mathbf{u}^*(k) = \ -\mathbf{G_e}\sum_{i=0}^{k}[\mathbf{y}(k) - \mathbf{y_d}(k)] \ - \mathbf{G_x}\mathbf{x}(k) \ - \sum_{l=1}^{N_l}\mathbf{G_d}(l)\mathbf{y_d}(k+l) \tag{5.5}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{C_d}\mathbf{B_d} \\ \mathbf{B_d} \end{bmatrix}, \ \tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{C_d}\mathbf{A_d} \\ \mathbf{A_d} \end{bmatrix}, \ \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q_e} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_x} \end{bmatrix}, \ \tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \ \tilde{\mathbf{A}} = [\tilde{\mathbf{I}} \quad \tilde{\mathbf{F}}] \tag{5.6}$$

... where the gains, $\mathbf{G_e}$, $\mathbf{G_x}$ and $\mathbf{G_d}(l)$ are given by;

$$\begin{aligned} \mathbf{G_e} &= [\mathbf{R} + \tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{B}}]^{-1}\tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{I}} \\ \mathbf{G_x} &= [\mathbf{R} + \tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{B}}]^{-1}\tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{F}} \\ \mathbf{G_d}(l) &= -[\mathbf{R} + \tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{B}}]^{-1}\tilde{\mathbf{B}}^\intercal(\tilde{\mathbf{A}}_\mathbf{c}^\intercal)^{(l-1)}\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{I}} \end{aligned} \tag{5.7}$$

... where;

$$\tilde{\mathbf{A}}_\mathbf{c} = \tilde{\mathbf{A}} - \tilde{\mathbf{B}}[\mathbf{R} + \tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{B}}]^{-1}\tilde{\mathbf{B}}^\intercal\tilde{\mathbf{K}}_\mathbf{d}\tilde{\mathbf{A}} \tag{5.8}$$

... and $\tilde{\mathbf{K}}_\mathbf{d}$ is a solution to the Discrete Time Algebraic Riccati Equation.

5.4. 3D Model Implementation

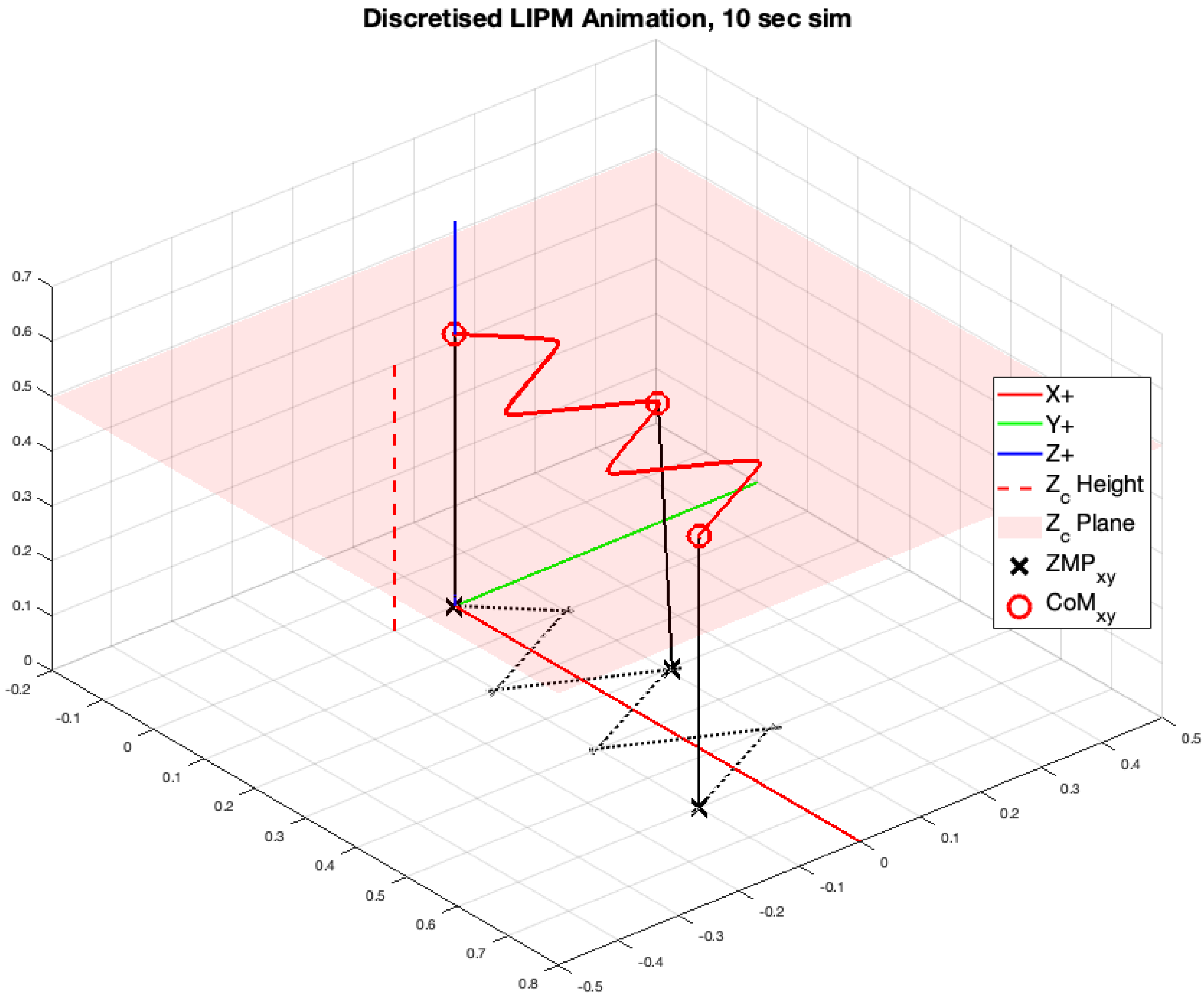


Figure 7: 3D LIPM with Preview Control.

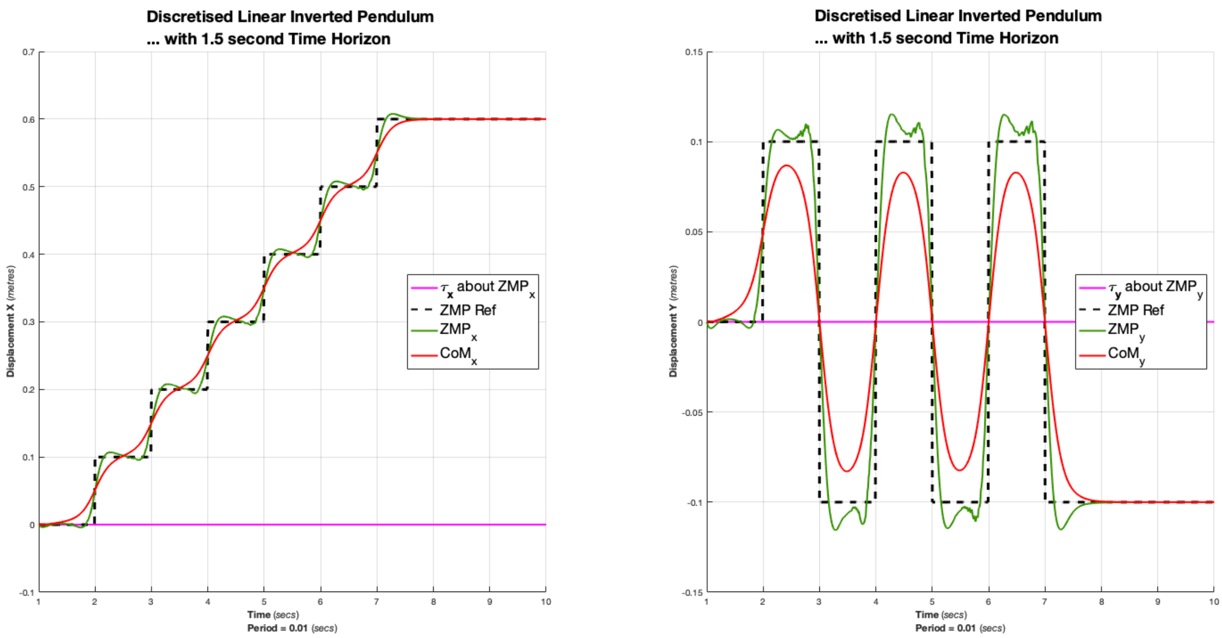


Figure 8: 3D LIPM X Y components.

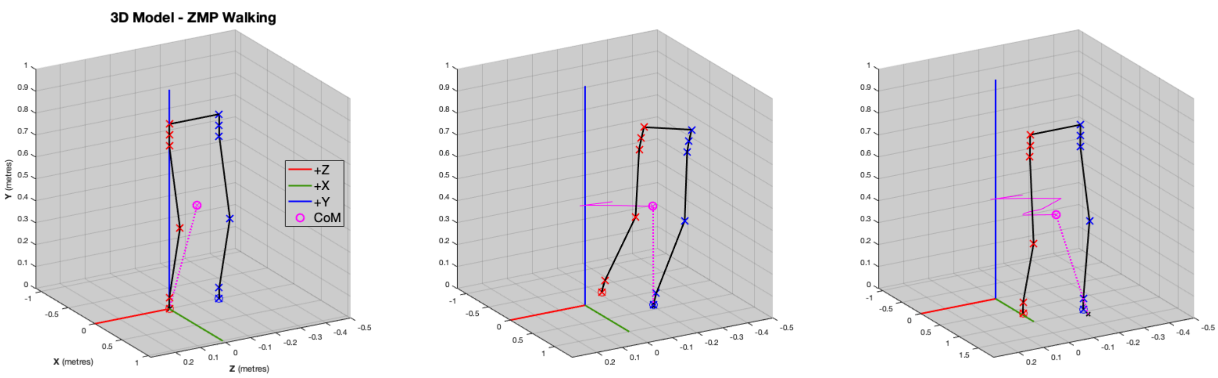


Figure 9: 3D model using ZMP walking with Preview Control.

6. Novel Method for Trajectory Tracking

6.1. Concept

7. Results

7.1. 2D Model

7.2. 3D Model

8. Conclusion

References

[1] Grimble, M. J., Johnson, M. A., Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. [2009], *Robotics*, London Springer London.

[2] Hirose, S. [1984], 'A study of design and control of a quadruped walking vehicle', *The International Journal of Robotics Research* **3**, 113–133.

[3] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K. and Hirukawa, H. [2003], 'Biped walking pattern generation by using preview control of zero-moment point', *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)* **3**.

[4] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K. and Hirukawa, H. [2002], 'A realtime pattern generator for biped walking', *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)* **4**.

[5] KATAYAMA, T., OHKI, T., INOUE, T. and KATO, T. [1985], 'Design of an optimal controller for a discrete-time system subject to previewable demand', *International Journal of Control* **41**, 677–699.

[6] Kato, I. [1973], 'The wabot-1', *Bulletin of Science and Engineering Research Laboratory, Waseda University* **Special Issue on WABOT**.

[7] Perez, T. [2014], *Fundamentals of mechanical systems*, The University of Newcastle.

[8] Raibert, M. H. [1986], *Legged robots that balance*, Mit Press.

[9] Renton, C. [2021], 'Lecture notes: Mcha4100 - mechatronics systems'.

[10] Vukobratović, M. and Stokić, D. [1983], 'Is dynamic control needed in robotic systems, and, if so, to what extent?', *The International Journal of Robotics Research* **2**, 18–34.

A. Handwritten Notes

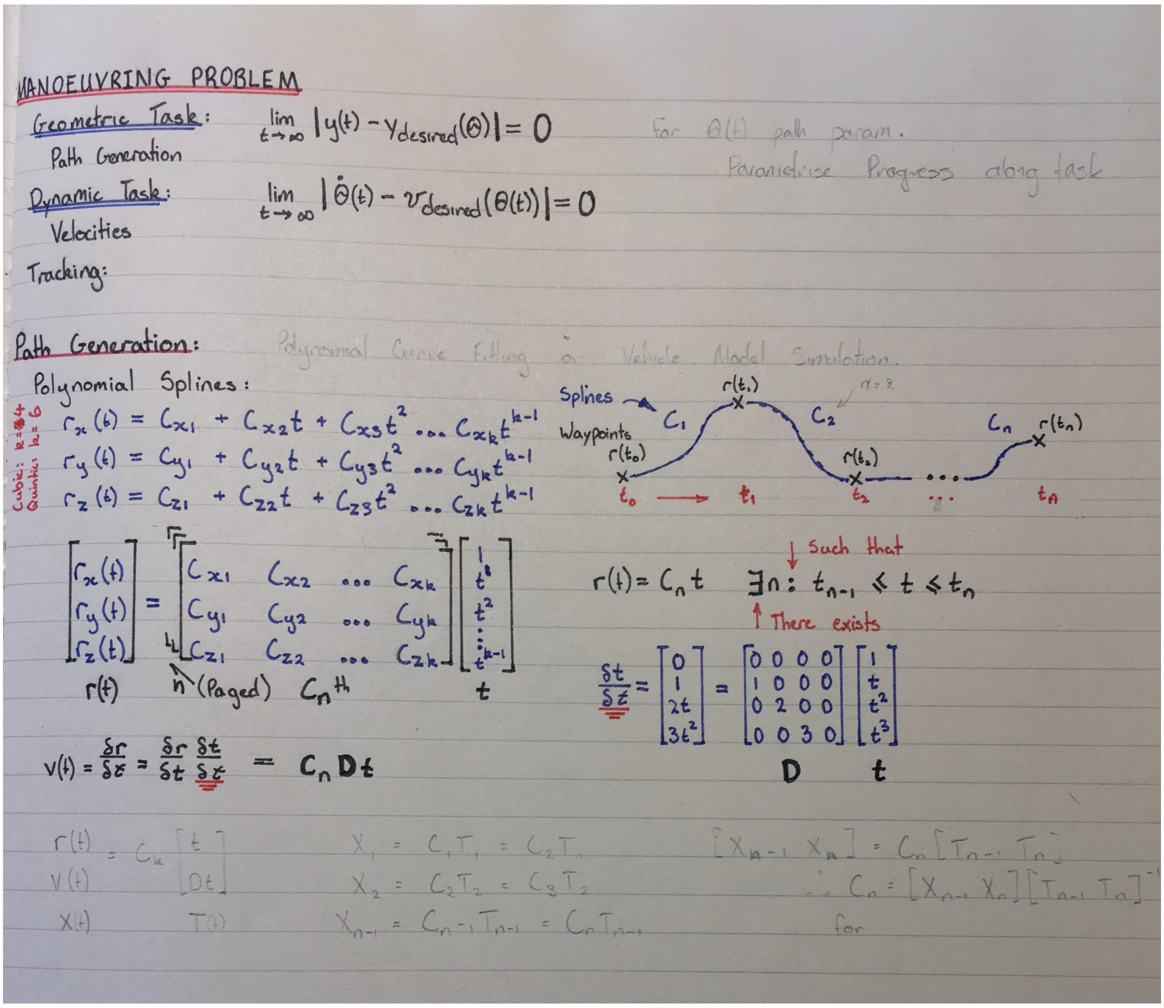


Figure 10: Trajectory Generation.

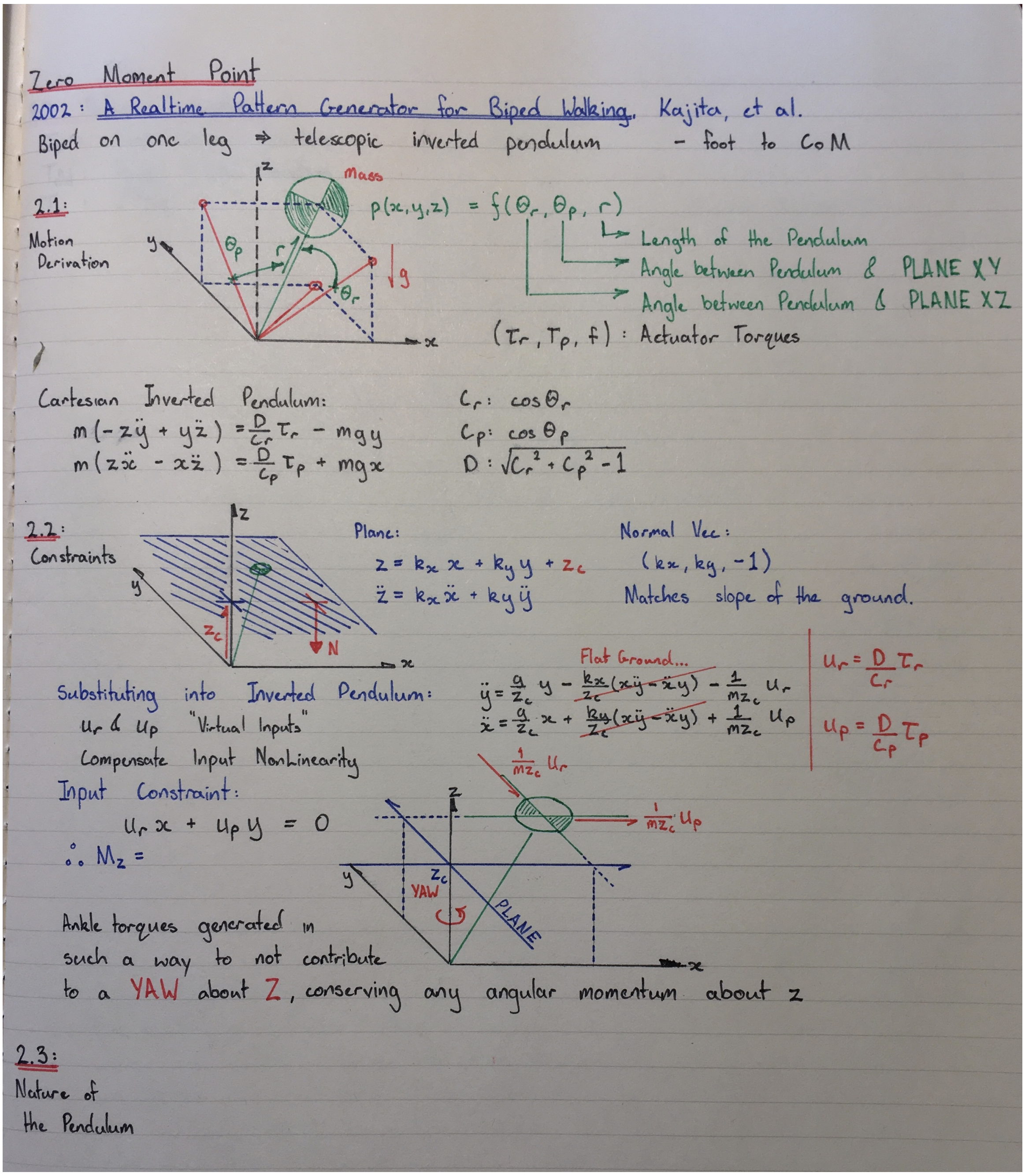
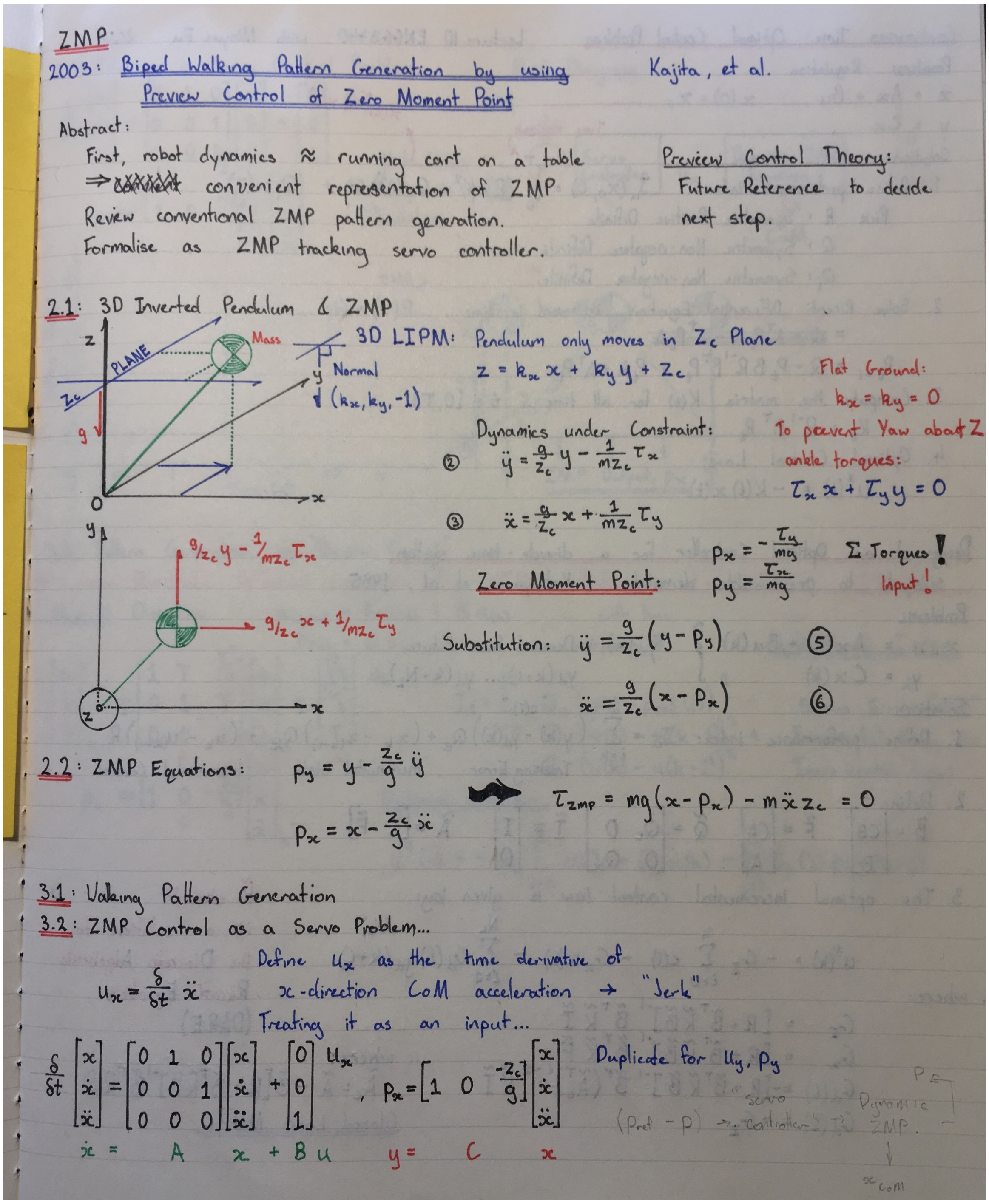


Figure 11: ZMP1.


ZMP

2003: Biped Walking Pattern Generation by using Preview Control of Zero Moment Point — Kajita, et al.

Abstract:

First, robot dynamics ≈ running cart on a table
⇒ ~~XXXXXX~~ convenient representation of ZMP
Review conventional ZMP pattern generation.
Formalise as ZMP tracking servo controller.

Preview Control Theory:
Future Reference to decide next step.

2.1: 3D Inverted Pendulum & ZMP

3D LIPM: Pendulum only moves in Z_c Plane

Normal $(k_x, k_y, -1)$

$z = k_x x + k_y y + z_c$

Flat Ground: $k_x = k_y = 0$

Dynamics under Constraint:

② $\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x$

③ $\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} \tau_y$

To prevent Yaw about Z ankle torques:
$\tau_x x + \tau_y y = 0$

Zero Moment Point: $p_x = -\frac{\tau_y}{mg}$, $p_y = \frac{\tau_x}{mg}$ — Σ Torques! Input!

Substitution: $\ddot{y} = \frac{g}{z_c}(y - p_y)$ ⑤

$\ddot{x} = \frac{g}{z_c}(x - p_x)$ ⑥

2.2: ZMP Equations: $p_y = y - \frac{z_c}{g}\ddot{y}$

$p_x = x - \frac{z_c}{g}\ddot{x}$ ⟹ $\tau_{zmp} = mg(x - p_x) - m\ddot{x} z_c = 0$

3.1: Walking Pattern Generation

3.2: ZMP Control as a Servo Problem...

$u_x = \frac{\delta}{\delta t}\ddot{x}$ — Define u_x as the time derivative of x-direction CoM acceleration → "Jerk". Treating it as an input...

$$\frac{\delta}{\delta t}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x, \quad p_x = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}$$

$\dot{x} = A x + B u$, $y = C x$

Duplicate for u_y, p_y

$(p_{ref} - p) \rightarrow$ servo controller $\rightarrow$ Dynamic ZMP $\rightarrow x_{CoM}$

Figure 12: ZMP2.

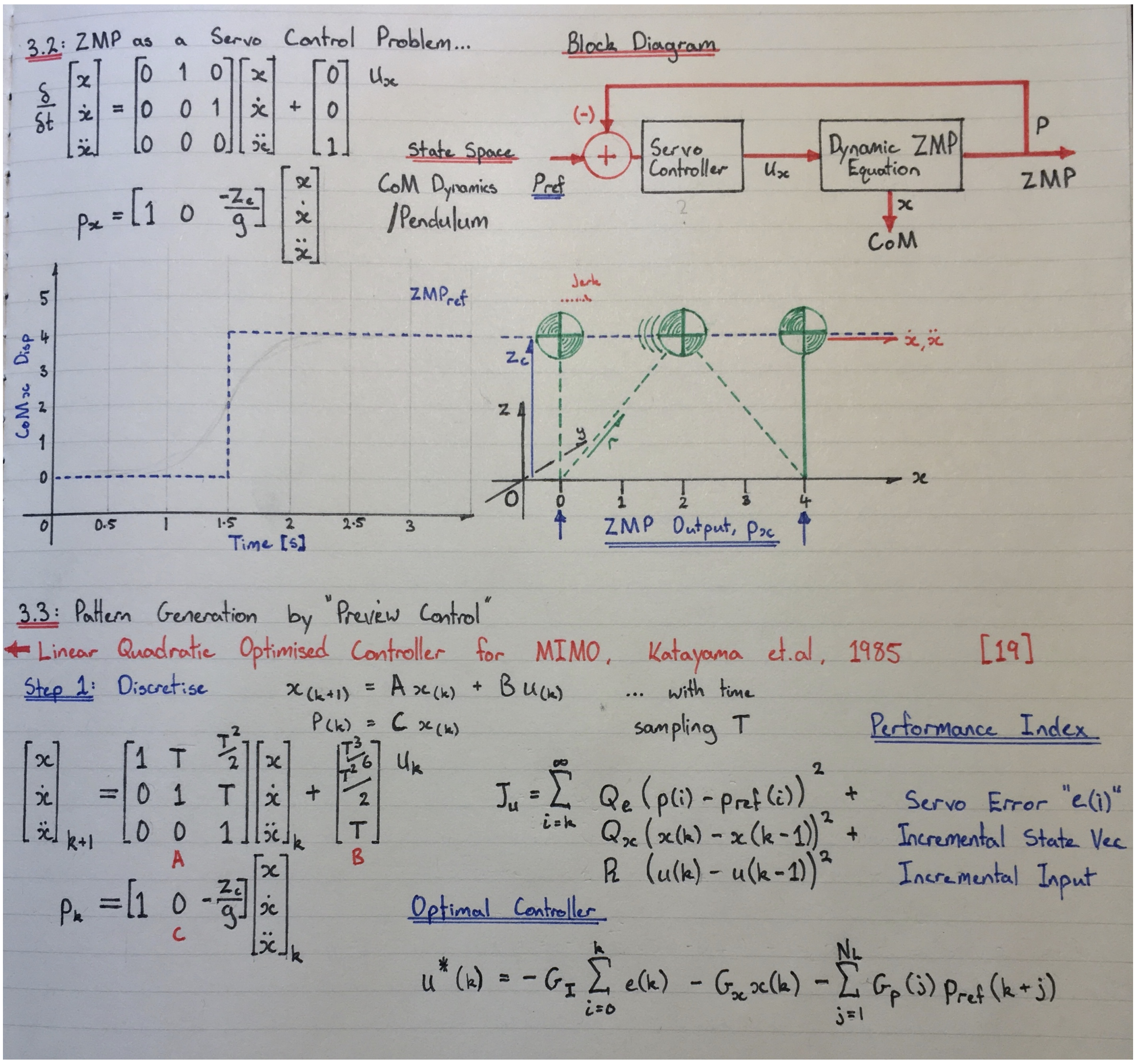


Figure 13: ZMP3.

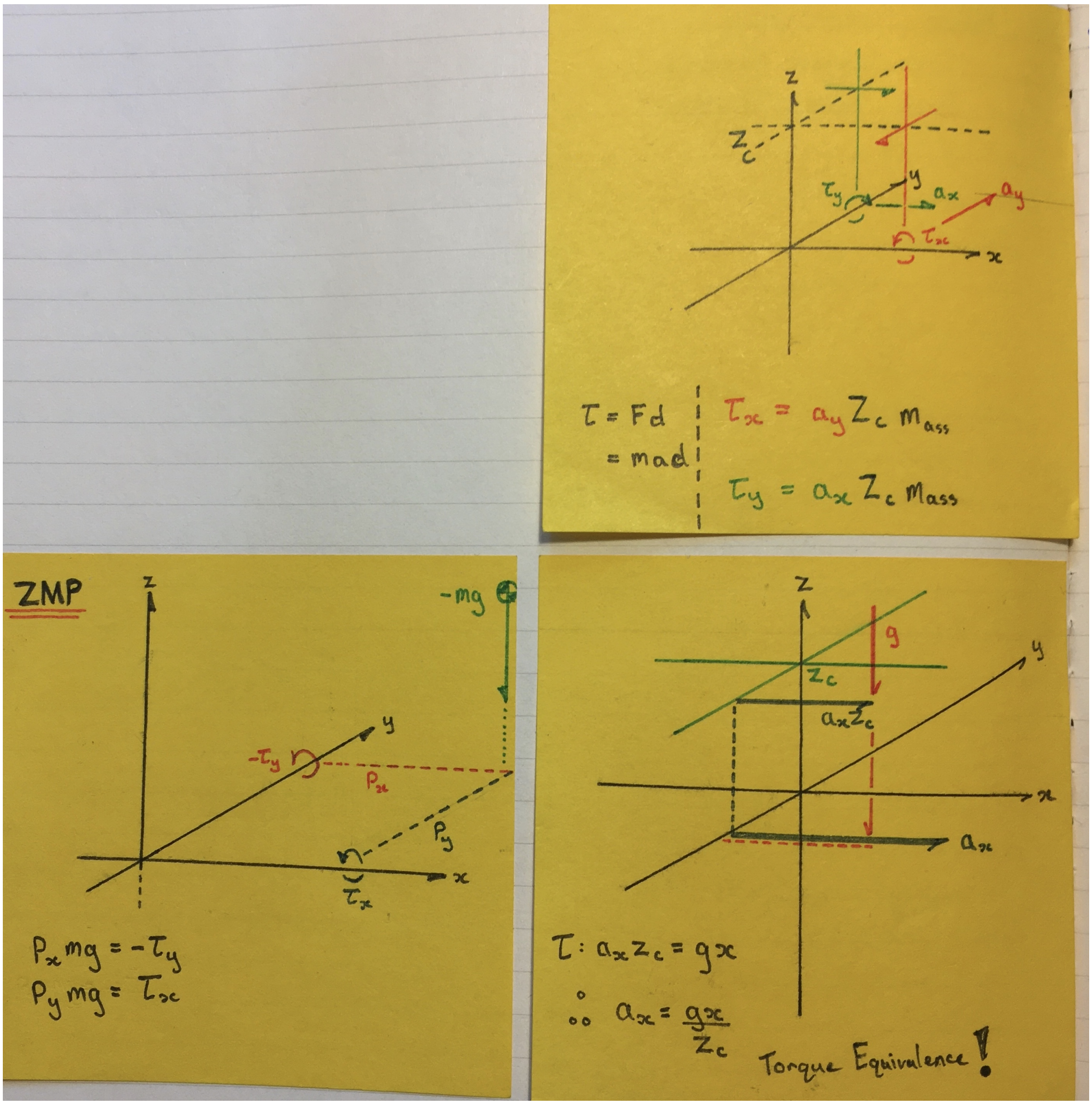
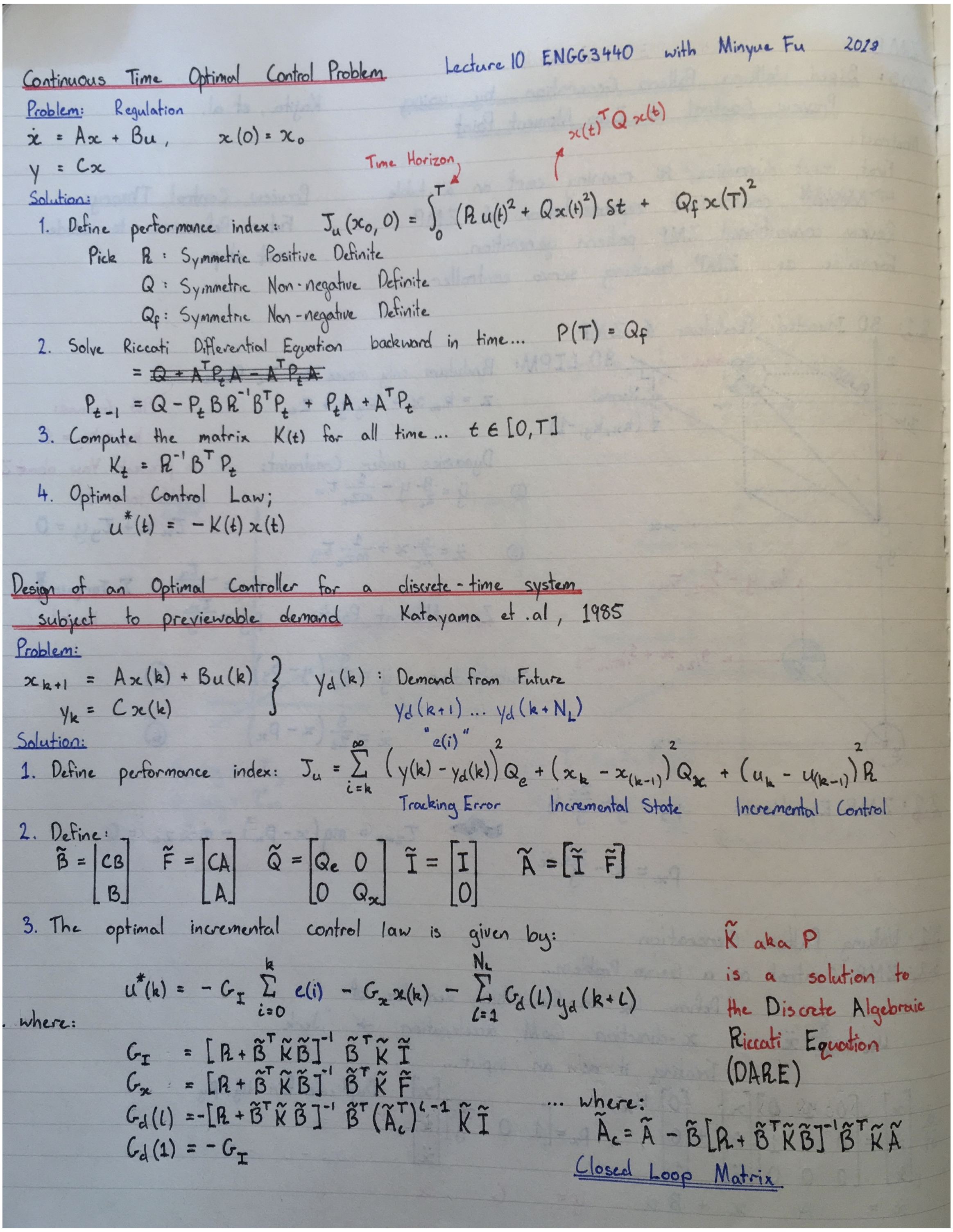


Figure 14: ZMP4.



Continuous Time Optimal Control Problem — Lecture 10 ENGG3440 with Minyue Fu 2018

Problem: Regulation

$\dot{x} = Ax + Bu, \quad x(0) = x_0$

$y = Cx$

Solution:

1. Define performance index: $J_u(x_0, 0) = \int_0^{\overset{\text{Time Horizon}}{T}} \left(R u(t)^2 + \overbrace{Q x(t)^2}^{x(t)^T Q x(t)}\right) \delta t + Q_f x(T)^2$

   Pick R : Symmetric Positive Definite

   Q : Symmetric Non-negative Definite

   Q_f : Symmetric Non-negative Definite

2. Solve Riccati Differential Equation backward in time... $P(T) = Q_f$

   $= \cancel{Q + A^T P_t A - A^T P_t A}$

   $P_{t-1} = Q - P_t B R^{-1} B^T P_t + P_t A + A^T P_t$

3. Compute the matrix $K(t)$ for all time ... $t \in [0, T]$

   $K_t = R^{-1} B^T P_t$

4. Optimal Control Law;

   $u^*(t) = -K(t)\,x(t)$

Design of an Optimal Controller for a discrete-time system subject to previewable demand — Katayama et .al, 1985

Problem:

$$\left.\begin{aligned} x_{k+1} &= Ax(k) + Bu(k) \\ y_k &= Cx(k) \end{aligned}\right\} \quad y_d(k) : \text{Demand from Future} \quad y_d(k+1) \dots y_d(k+N_L)$$

Solution:

1. Define performance index: $J_u = \sum_{i=k}^{\infty} \underbrace{\overbrace{(y(k) - y_d(k))}^{\text{"}e(i)\text{"}}{}^2 Q_e}_{\text{Tracking Error}} + \underbrace{(x_k - x_{(k-1)})^2 Q_x}_{\text{Incremental State}} + \underbrace{(u_k - u_{(k-1)})^2 R}_{\text{Incremental Control}}$

2. Define:

$$\tilde{B} = \begin{bmatrix} CB \\ B \end{bmatrix} \quad \tilde{F} = \begin{bmatrix} CA \\ A \end{bmatrix} \quad \tilde{Q} = \begin{bmatrix} Q_e & 0 \\ 0 & Q_x \end{bmatrix} \quad \tilde{I} = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad \tilde{A} = [\tilde{I} \;\; \tilde{F}]$$

3. The optimal incremental control law is given by:

$$u^*(k) = -G_I \sum_{i=0}^{k} e(i) - G_x x(k) - \sum_{l=1}^{N_L} G_d(l)\, y_d(k+l)$$

where:

$G_I = [R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T \tilde{K} \tilde{I}$

$G_x = [R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T \tilde{K} \tilde{F}$

$G_d(l) = -[R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T (\tilde{A}_c^T)^{l-1} \tilde{K} \tilde{I}$

$G_d(1) = -G_I$

... where:

$\tilde{A}_c = \tilde{A} - \tilde{B}[R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T \tilde{K} \tilde{A}$

Closed Loop Matrix

$\tilde{K}$ aka P is a solution to the Discrete Algebraic Riccati Equation (DARE)

Figure 15: Preview Control.