

國立交通大學

電控工程研究所

碩士論文

雙足步行機器人於未知斜面之姿態控制設計

Pose-Control Design for Bipedal Walking on
Unknown Slope Surfaces

研究生：謝章宏

指導教授：宋開泰 博士

中華民國一百零二年五月

雙足步行機器人於未知斜面之姿態控制設計

Pose-Control Design for Bipedal Walking on
Unknown Slope Surfaces

研究生：謝章宏

Student: Chang-Hung Hsieh

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song

國立交通大學

電控工程研究所

碩士論文

A Thesis

Submitted to Institute of Electrical Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical Control Engineering

May 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年五月

雙足步行機器人於未知斜面之姿態控制設計

學生:謝章宏

指導教授:宋開泰 博士

國立交通大學電控工程研究所

摘要

本論文之目的為設計一雙足機器人之步態控制器，使其於步行中遇到地面坡度改變時能即時進行姿態補償控制，以避免傾倒。本設計藉由安裝於機器人之陀螺儀與加速規取得上半身姿態資訊，使雙足機器人得以在變化的坡度地形，產生適當的姿態調整動作以完成平衡步行。本論文以 Central Pattern Generator(CPG) 為主要理論基礎來設計機器人行走時的完整步態軌跡。在平衡行走系統設計上，本論文發展出一套補償控制器演算法，在雙足步行中藉由機器人上半身姿態的感測值，判斷地形坡度與上半身姿態兩者關係，即時產生對機器人上半身姿態調整之補償動作。此補償動作分為預測補償(Predictive Compensation)和即時補償(Immediate Compensation)，兩種控制法則結合使用，使機器人得以克服斜坡變化順利完成連續步行。本論文以雙足機器人 NAO 進行模擬和實驗，驗證所提出之平衡行走系統的可行性。經過模擬與實驗驗證，此系統可達成雙足機器人於未知斜面間進行平穩的行走。

Pose-Control Design for Bipedal Walking on Unknown Slope Surfaces

Student: Chang-Hung Hsieh

Advisor: Dr. Kai-Tai Song

Institute of Electrical Control Engineering

National Chiao Tung University

Abstract

The objective of this study is to design a walking pattern generator and a balance control system such that a bipedal robot can handle an unknown slope in bipedal walking. The robot uses onboard gyro and accelerometer sensors to detect the pose information of upper-body. A controller is proposed for the robot to walk on unknown slope with suitable adjustment of the tilt angle of the upper-body pose. The theory of central pattern generator(CPG) is applied to generate the walking trajectory. By the pose information of upper-body, we develop a method to determine the relationship of the slope of the terrain and the upper-body posture and generate the compensation motion to adjust the tilt angle of the upper-body. The compensation control consists of the predictive compensation and the immediate compensation. The integrated controller adapts to the unknown slope while robot walking. Using the bipedal robot NAO, the simulation and the experimental results show that the biped robot can walk successfully on unknown slopes.

誌謝

謹向我的指導教授宋開泰博士致上感謝之意，感謝他三年來給予我在專業研究上的指導上和生活上待人處事的教導，並以他豐富的學識與經驗，配合理論的應用，使得本論文得以順利完成。

感謝學長孟儒、嘉豪、格豪、信毅、允智、育萱、仕晟、建宏、上峻及學姊巧敏在理論及實驗過程中的指導，以及感謝同學家昌和 Carlos 在學習過程中互相勉勵與成長，同時感謝學弟昭宇、俊儒、京叡、奕夫、瑋哲、政輝、佑霖、明翰以及學妹依穎在生活與課業帶來的樂趣。

最後，特別感謝我的父母，由於他們時刻的關心與辛苦的栽培，以及在生活與精神上的支持與溫暖，並在研究過程中給了我許多的鼓勵，是促使我向前的動力，使我才能順利完成論文，在此我願以此論文獻給我最摯愛的父母。



Symbols and Abbreviations

Symbols

N_1, N_2	Neuron
u_1, u_2	Membrane potential
v_1, v_2	Variable of self-inhibition of the neuron
τ_1, τ_2	Time constant
c	Tonic excitation
β	Adaptation constant
γ	Control constant
g_j	Number j of inputs
h_j	Scaled gain of input g_j
q	Output of the whole oscillator
$[x]^+$	The maximum positive value of 0 and x
$[x]^-$	The minimum negative value of 0 and x
p_{Lz}	Z-direction length from left foot to shoulder
p_{Rz}	Z-direction length from right foot to shoulder
x_0	Offset of X-direction between two foot sole in DSP
p_{Lx}	Difference value between x_0 and left foot position
p_{Rx}	Difference value between x_0 and right foot position
p_{Sy}	Coordinate position of shoulder center in Y-direction
p_{Sx}	Coordinate position of shoulder center in X-direction
p_{Ly}	Coordinate position of left foot sole in Y-direction
p_{Ry}	Coordinate position of right foot sole in Y-direction
t_s	Current time in DSP
$T_{Two_Support}$	Total time of DSP
k_{Sx}	Scaled gain of shoulder center moving in X-direction
k_T	Adjust the curve of shoulder center moving in X-direction
z_0	Initial length of Z-direction
q_z	Output of oscillator for Z-direction
A_z	Scaled factor of the amplitude of q_z
q_x	Output of oscillator for X-direction
A_x	Scaled factor of the amplitude of q_x
$\theta_{HipPitch_Swing}$	Control upper-body to swing of forward-backward
θ_{pitch_comp01}	Predictive compensation value for pitch plane
θ_{pitch_comp02}	Real-time compensation value for pitch plane
$\theta_{HipPitch_All}$	Control the angle value of Hip-Pitch joint

$\theta_{HipRoll_Swing}$	Control upper-body to swing of left-right
θ_{roll_comp}	Real-time compensation value for roll plane
$\theta_{HipRoll_All}$	Control the angle value of Hip-Roll joint
q_{hip_pitch}	Oscillator output for pitch plane
k_{hip_pitch}	Scaled factor of q_{hip_pitch}
q_{hip_roll}	Oscillator output for roll plane
k_{hip_roll}	Scaled factor of q_{hip_roll}
$[x\ y\ z]^T$	Absolute coordinate frame of joints
$[k]$	Representing the timing which is last time of the joint angles output to control robot in DSP
$AngY_{DSP}[k]$	Angular value of pitching angle of upper-body at k time
$\omega_AngY_{DSP}[k]$	Angular velocity value of pitching angle of upper-body at k time
$AngY$	Sensor data of pitching angle of upper-body
t	Timer
(S)	Step of No.S
$AngY_{Torso}$	Angular value of pitching angle of upper-body in DSP
k_{θ}	Scaled factor of $AngY_{DSP}$
k_{ω}	Scaled factor of ω_AngY_{DSP}
t_c	Average execution time of program
g_i	The input of oscillator for pitch plane
k_{g_i}	Scaled factor of $AngY_{DSP}[k]$
k_{pcp02}	Scaled factor of $AngY_{DSP}[k]$
$AngX$	Sensor data of rolling angle of upper-body
k_{rcp}	Scaled factor of $AngY_{DSP}[k]$
$\theta_i, i=1\sim 5$	The angle value of joint i
$\begin{bmatrix} absolute_x \\ absolute_y \\ absolute_z \end{bmatrix}$	Absolute coordinate frame of joints
$\begin{bmatrix} world_x \\ world_y \\ world_z \end{bmatrix}$	World coordinate frame of robot
θ_{ij}	The angle value of solution j of joint i
$u_{z1}, v_{z1}, u_{z2}, v_{z2}, \tau_{z1}$	Parameter of oscillator for Z-direction
$u_{x1}, v_{x1}, u_{x2}, v_{x2}, \tau_{x1}$	Parameter of oscillator for X-direction
$u_{hp1}, v_{hp1}, u_{hp2}, v_{hp2}, \tau_{hp1}$	Parameter of oscillator for pitch plane
$u_{hr1}, v_{hr1}, u_{hr2}, v_{hr2}, \tau_{hr1}$	Parameter of oscillator for roll plane

P	The position of vector of Zero Moment Point
f_j	The measured pressure value of the j-th force sensor
s_j	The position of vector of the j-th force sensor
b_{j0}	The output value of vacant of the j-th force sensor
b_j	The output value of the j-th force sensor
b_{js}	The parameter of calibration

Abbreviations

ZMP	Zero Moment Point
CoM	Center of Mass
IMU	Inertial Measurement Unit
CPG	Central Pattern Generator
DSP	Double Support Phase
SSP	Single Support Phase
D-H	Denavit-Hartenberg
FSRs	Foot Sensing Resistors
SDK	Software Development Kit



目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
Symbols and Abbreviations	iv
目錄.....	vii
圖例.....	ix
表例.....	xiii
第一章 緒論.....	1
1.1 研究動機與目的.....	1
1.2 神經振盪元 CPG 原理.....	1
1.3 相關研究回顧.....	11
1.4 問題描述.....	16
1.5 系統架構.....	17
1.6 章節說明.....	17
第二章 雙足步行步態規劃.....	18
2.1 行走步態之規劃.....	18
2.2 單腳支撐相.....	19
2.3 雙腳支撐相.....	24
第三章 在不平地面行走之補償設計.....	26
3.1 系統設計理念.....	26
3.2 行走的預測補償設計.....	28
3.3 行走的即時補償設計.....	31
3.4 補償控制的使用時機.....	35
3.5 補償方式討論.....	36
第四章 機器人實驗平台.....	38
4.1 NAO 之硬體介紹.....	38
4.2 NAO 之運動學計算.....	40
4.3 NAO 之感測器.....	45
4.4 NAO 之軟體介紹.....	48
4.5 NAO 之電腦模擬系統.....	49
第五章 模擬與實驗結果.....	52

5.1	模擬和實驗之參數設定.....	52
5.2	平衡行走之模擬結果.....	56
5.2.1	平衡行走於4°斜坡之模擬結果.....	56
5.2.2	平衡行走於5°斜坡之模擬結果.....	60
5.2.3	平衡行走於7°斜坡之模擬結果.....	64
5.2.4	模擬結果之比較.....	68
5.3	平衡行走之實驗結果.....	68
5.3.1	步態驗證之平地行走.....	69
5.3.2	斜坡行走實驗結果之 ZMP 驗證.....	71
5.3.2.1	平地→4°上坡之 ZMP 驗證.....	71
5.3.2.2	平地→4°下坡之 ZMP 驗證.....	75
5.3.2.3	平地→7°上坡之 ZMP 驗證.....	77
5.3.2.4	7°下坡→平地之 ZMP 驗證.....	79
5.3.3	斜坡行走實驗結果之控制器效果驗證.....	81
5.3.3.1	平地→4°上坡→平地之行走實驗結果.....	81
5.3.3.2	平地→4°下坡→平地之行走實驗結果.....	85
5.3.3.3	平地→7°上坡之行走實驗結果.....	90
5.3.3.4	7°下坡→平地之行走實驗結果.....	93
5.3.4	實驗結果之討論.....	96
5.4	討論.....	96
第六章	結論與未來展望.....	99
6.1	結論.....	99
6.2	未來展望.....	100
參考文獻	101

圖例

圖 1.1 在 IEEE 探索資料庫利用詞彙 Central Pattern Generator 或是 CPG 與 Robot 一起進行搜尋時所得到的相關文章數.....	3
圖 1.2 振盪器的模型圖.....	5
圖 1.3 對於有無滿足公式(1.7)的 CPG 輸出曲線變化圖(a)無滿足公式因此不會振盪(b)有滿足公式因此保持週期性振盪.....	7
圖 1.4 對於有無滿足公式(1.8)的 CPG 輸出曲線變化圖(a)無滿足公式因此振盪時振幅依次縮減至 0 (b)有滿足公式因此保持同振幅的週期性振盪.....	7
圖 1.5 CPG 輸出曲線的自然頻率與 $1/\tau_1$ 的關係曲線圖.....	8
圖 1.6 在輸入振幅改變時，CPG 輸出的對應變化關係圖.....	9
圖 1.7 當輸入為固定常數時，CPG 輸出的對應變化關係圖.....	10
圖 1.8 說明運動產生方式，藉由(a)神經系統(b)傳統機器人系統.....	11
圖 1.9 機器人姿態修正示意圖各別在(a)前向平面 (b)側向平面.....	12
圖 1.10 (a)地面感測(b)從雙足位移及正交方向估測出新的表面梯度.....	13
圖 1.11 機器人跌倒情況分析(a)下坡情況(b)上坡情況.....	13
圖 1.12 崎嶇地面行走時的整體控制流程.....	14
圖 1.13 行走控制之架構圖[16].....	14
圖 1.14 (a)整體 CPG 網絡 (b)與人相對的關節比較示意圖.....	15
圖 1.15 利用 CPG 的整體控制架構.....	16
圖 1.16 系統流程圖.....	17
圖 2.1 整體的步態規劃.....	19
圖 2.2 四個振盪器的控制安排.....	21
圖 2.3 (a)前向平面上的軀幹搖擺示意圖(b)側向平面上的軀幹搖擺示意圖.....	23
圖 3.1 整體之平衡行走控制架構圖.....	28
圖 3.2 雙足行走處於雙腳支撐相時期之整體行走架構圖.....	29

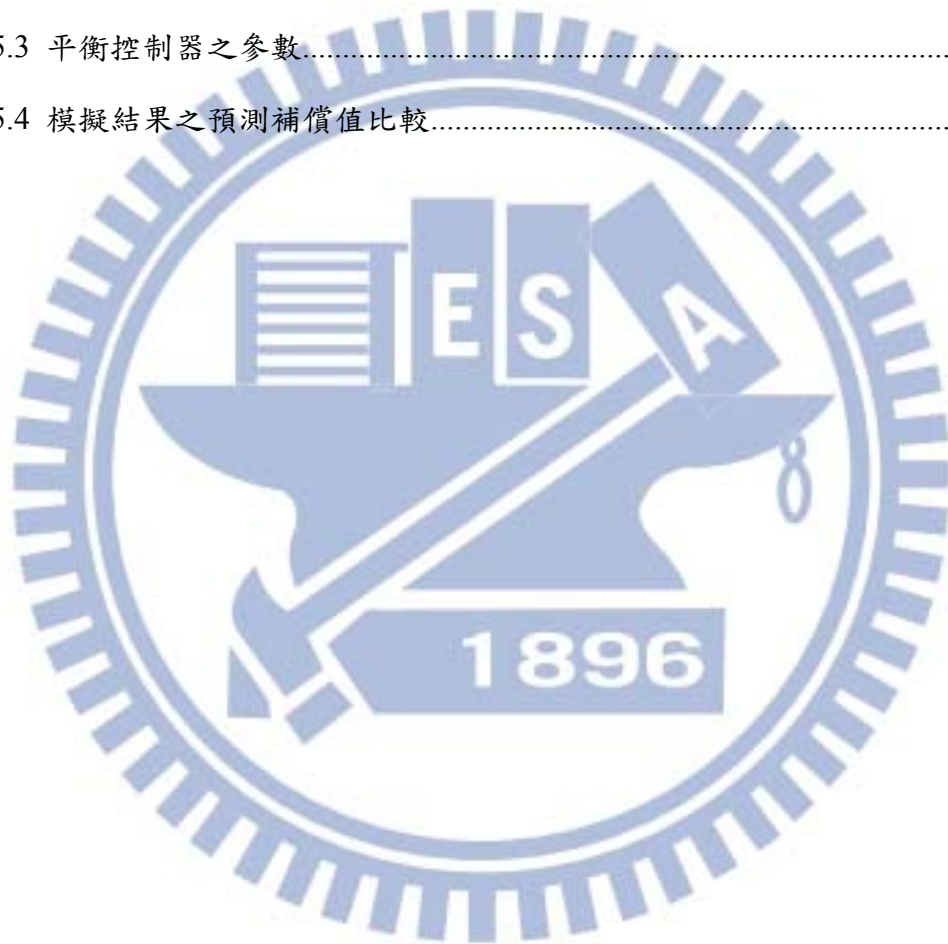
圖 3.3 雙足行走處於單腳支撐相時期之整體行走架構圖.....	32
圖 3.4 控制器的使用時機示意圖.....	35
圖 4.1 機器人結構上桿件間長度說明(單位為公厘).....	39
圖 4.2 雙腳自由度資訊與運動學使用上之命名.....	39
圖 4.3 NAO 雙腳之 D-H model.....	41
圖 4.4 一組位置可解出之所有角度解.....	45
圖 4.5 NAO 足部壓力計分布圖.....	47
圖 4.6 Python SDK 之程式介面.....	49
圖 4.7 NAO 之程式與硬體之間的傳輸方式.....	49
圖 4.8 虛擬 NAO 的模擬環境.....	50
圖 4.9 虛擬 NAO 站立時之 Gyro 感測器數值模擬圖.....	51
圖 5.1 平衡行走於 4° 斜坡之模擬過程.....	57
圖 5.2 行走過程中之 Y 方向旋轉角度曲線.....	57
圖 5.3 直接補償於關節 Hip-Pitch 之預測補償角度值.....	58
圖 5.4 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	58
圖 5.5 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	59
圖 5.6 行走過程中之 X 方向旋轉角度曲線.....	59
圖 5.7 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	60
圖 5.8 平衡行走於 5° 斜坡之模擬過程.....	61
圖 5.9 行走過程中之 Y 方向旋轉角度曲線.....	61
圖 5.10 直接補償於關節 Hip-Pitch 之預測補償角度值.....	62
圖 5.11 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	62
圖 5.12 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	63
圖 5.13 行走過程中之 X 方向旋轉角度曲線.....	63
圖 5.14 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	63
圖 5.15 平衡行走於 7° 斜坡之模擬過程.....	64

圖 5.16 行走過程中之 Y 方向旋轉角度曲線.....	65
圖 5.17 直接補償於關節 Hip-Pitch 之預測補償角度值.....	65
圖 5.18 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	66
圖 5.19 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	66
圖 5.20 行走過程中之 X 方向旋轉角度曲線.....	67
圖 5.21 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	67
圖 5.22 (a)4°之斜坡環境(b) 7°之斜坡環境.....	69
圖 5.23 穩定行走之步態切片圖.....	69
圖 5.24 步態行走時，四個 CPG 之個別振盪圖.....	70
圖 5.25 ZMP 軌跡.....	72
圖 5.26 行走過程中之 Y 方向旋轉角度曲線.....	72
圖 5.27 行走過程中之 X 方向旋轉角度曲線.....	73
圖 5.28 直接補償於關節 Hip-Pitch 之預測補償角度值.....	73
圖 5.29 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	74
圖 5.30 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	74
圖 5.31 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	75
圖 5.32 ZMP 軌跡.....	76
圖 5.33 行走過程中之 Y 方向旋轉角度曲線.....	76
圖 5.34 行走過程中之 X 方向旋轉角度曲線.....	77
圖 5.35 ZMP 軌跡.....	78
圖 5.36 行走過程中之 Y 方向旋轉角度曲線.....	78
圖 5.37 行走過程中之 X 方向旋轉角度曲線.....	79
圖 5.38 ZMP 軌跡.....	80
圖 5.39 行走過程中之 Y 方向旋轉角度曲線.....	80
圖 5.40 行走過程中之 X 方向旋轉角度曲線.....	81
圖 5.41 平地→4°上坡→平地之行走實驗過程.....	82

圖 5.42 行走過程中之 Y 方向旋轉角度曲線.....	82
圖 5.43 直接補償於關節 Hip-Pitch 之預測補償角度值.....	83
圖 5.44 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	84
圖 5.45 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	84
圖 5.46 行走過程中之 X 方向旋轉角度曲線.....	85
圖 5.47 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	85
圖 5.48 平地→4°下坡→平地之行走實驗過程.....	86
圖 5.49 行走過程中之 Y 方向旋轉角度曲線.....	87
圖 5.50 直接補償於關節 Hip-Pitch 之預測補償角度值.....	87
圖 5.51 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	88
圖 5.52 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	88
圖 5.53 行走過程中之 X 方向旋轉角度曲線.....	89
圖 5.54 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	89
圖 5.55 行走過程中之 Y 方向旋轉角度曲線.....	90
圖 5.56 直接補償於關節 Hip-Pitch 之預測補償角度值.....	91
圖 5.57 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	91
圖 5.58 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	91
圖 5.59 行走過程中之 X 方向旋轉角度曲線.....	92
圖 5.60 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	92
圖 5.61 行走過程中之 Y 方向旋轉角度曲線.....	94
圖 5.62 直接補償於關節 Hip-Pitch 之預測補償角度值.....	94
圖 5.63 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係.....	94
圖 5.64 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值.....	95
圖 5.65 行走過程中之 X 方向旋轉角度曲線.....	95
圖 5.66 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值.....	96

表例

表 4.1 雙腳各個馬達的旋轉角度之限制.....	38
表 4.2 NAO 之雙腳 D-H model 參數.....	41
表 4.3 壓力感測器量測值.....	48
表 5.1 CPG 振盪器之參數，模擬與實驗皆用此組參數.....	53
表 5.2 步態規劃之參數.....	54
表 5.3 平衡控制器之參數.....	56
表 5.4 模擬結果之預測補償值比較.....	68



第一章 緒論

1.1 研究動機與目的

在現今機器人的應用上，除了工業用機器人以外，在其他領域的機器人應用也是蓬勃的發展中，例如：家用機器人、醫療用機器人...等等。對於這些機器人的研究開發與應用，除了要代替人類勞力與使人類可以不用在有危險的場合工作之外，許多機器人的應用是需要直接與人類進行互動已提供服務，因此在機器人的設計上，較友善的擬人化機器人近幾年來受到很多重視。

雙足機器人與人們互動較為友善且人性化，它的多關節與高自由度使其靈活度高，使得動作近似人類，並且在環境中的移動可以較為自由。在雙足機器人的研究方面中，除了希望機器人整體可以更加的擬人化之外，更希望它能發揮出足型機器人的功能優點，例如克服不同的地形障礙，與人們進行移動式的互動。因此對於雙足機器人在不同環境中如何即時因應環境變化來達成穩定平衡，需要進行深入的探討與研究。

本研究的目的是為了使雙足機器人在面對了不同的地形環境之下，都可以保持穩定行走，不會跌倒，所以，我們希望結合機器人具備之感測器，可以即時的偵測出機器人目前的狀態及環境的狀況，進行即時的姿態控制，致使機器人可以持續平衡的進行穩定的行走，克服環境變化所帶來的不穩定因素。

1.2 神經振盪元 CPG 原理

為了使機器人行走控制的整體思路可以更近似於人類，我們在對於二足機器人平衡行走的整體設計上希望以生物學的觀點來達成行走步態的設計和感測回授的控制。我們透過對生物神經學的研究[1-3]發現，生物體的運動在神經系統反應上分為四種：第一種為反射(Reflex)(如敲膝時小腿彈)，此運動特徵為自發且定型，在對感知訊號的響應是沒有閾值限制的；第二種為確定行為(Determine Behavior)模式(如打噴嚏)，它也是自發且定型，但此運動需達到一定的刺激閾值才能被激發；第三種為節律運動(Rhythmic movement)(如行走、呼吸、心跳)，此

運動特點為定型並非自覺的循環，卻又受到連續自覺的控制；第四種為受控運動 (Controller movement)(如伸手動作)，屬於自覺的，通常是不定型也沒重複性。從上述得知，對於雙足機器人的行走設計屬於生物體運動的第三種-節律運動。

另一方面，研究發現，脊椎動物的節律性運動，例如：呼吸、心跳、行走...等，並不是從大腦產生，也不用透過外部傳感器輸入來產生運動，而是由位於神經中樞系統中的腦幹及脊髓中的神經網絡組織所控制達成的[2]。更詳盡的解釋為整個節律性的週期運動是利用脊椎當中脊髓軸突神經產生的週期性生物電流放電刺激肌肉運動因而產生，不是透過大腦另外下達需要有週期性的循環才會產生此節律性運動。而大腦的命令只是用來當節律性運動面對到環境改變的情況時，再透過感測資訊當輸入，經過判斷進行此時運動行為所需的因應變化的控制。

研究者將脊髓在不用接收節奏性輸入的情況下，可以產生有節奏的神經運動的此功能稱之為 Central Pattern Generator(CPG)[1]，CPG 屬於一種複雜的神經迴路，但一般是由類似神經軸突一類的可興奮細胞所構成，這種神經細胞受到刺激後狀態(通常用細胞膜內外電位差來做為狀態標誌，稱之為膜電位)可發生顯著變化，利用此變化可以產生複雜的自發性節奏行為，甚至透過感知系統對於外界環境所得的感測資訊當成輸入刺激進行節奏行為上的反射控制(如步態改變)。

對於 CPG 應用在機器人移動的控制系統上已經越來越常見[4]，當我們在 IEEE 探索資料庫利用詞彙 Central Pattern Generator 或是 CPG 與 Robot 一起進行搜尋時所得到的相關文章統計，從 2001 年到 2012 年數量有持續增加的趨勢，如圖 1.1 所顯示。CPG 在機器人學(Robotics)研究上，主要研究有：各種多足機器人的移動步態改變；模擬實驗水陸兩棲之機器人，如：鮡(Salamander)，研究如何直接用 CPG 在沒有輸入改變的曲線狀況下，進行水中游泳、陸上行走的狀態改變；或是透過與生物的研究搭配，如：生物先行測試出當大腦的訊號輸入被阻斷或是不完整訊息時，生物只使用 CPG 所產生的移動模式為何，或是測試生物當一隻腳受傷時，生物使用 CPG 產生的移動模式發生改變的狀態...等生物研究結果，並再把這些生物實驗實現在仿生物的機器人中，透過機器人的行為進行實驗，

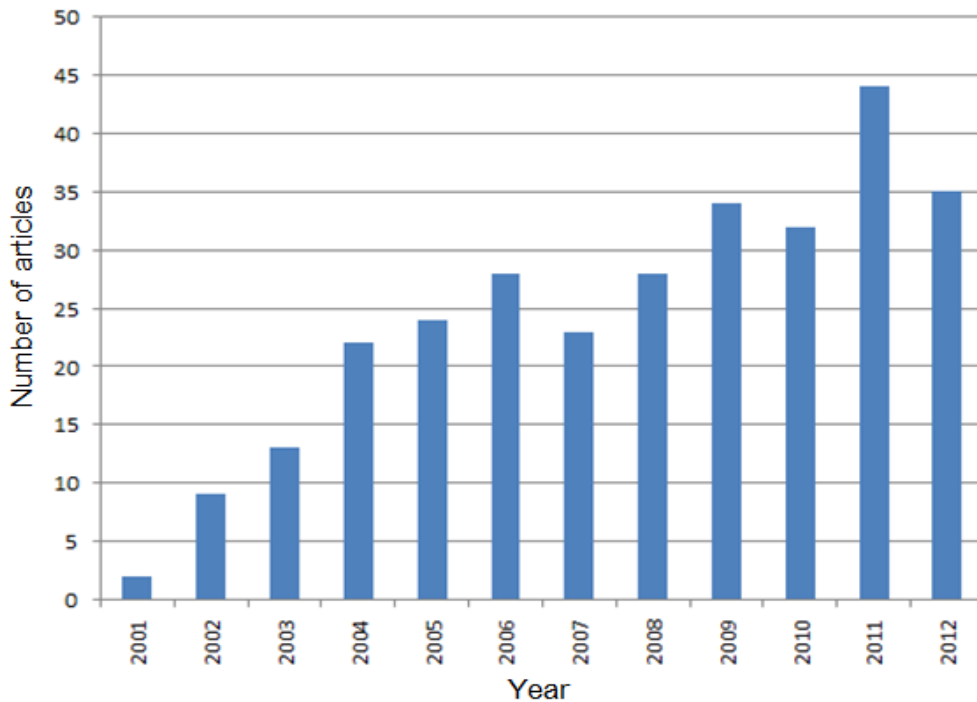


圖 1.1 在 IEEE 探索資料庫利用詞彙 Central Pattern Generator 或是 CPG 與 Robot 一起進行搜尋時所得到的相關文章數

驗證有何因素會導致 CPG 的輸出曲線即整體行為模式產生改變。而這所有 CPG 的應用，一方面為影響到對於機器人產生步態的方式改變，可以有越來越生物化的移動方式產生；另一方面為透過與生物的行為模式搭配，希望可以對於 CPG 有更深入的了解，並應用在復健機器人(如:腦中風、腳受傷時的復健應用)上，使得透過刺激訊號的改變來達到運動行為的改善。

CPG 神經元有以下幾個優點[4]:

- 在沒有外來刺激要求使得系統振盪的情況下，CPG 系統仍然會自動地開始產生週期性振盪。
- 當利用基於環境的感測輸入來產生適當的運動，此時可以達成即時的自適應學習和調整。
- CPG 是快速回授控制的迴路，對於輸入的改變輸出能即時的應對。
- 在輸入和輸出之間有自動地同步功能，且 CPGs 可以被即時的加入或移除並且此時系統仍然繼續運作。

從生物神經學與控制應用研究上，我們得知 CPG 的優點，因此我們希望將 CPG 應用在雙足機器人的行走控制來實現自發性的週期性的單腳支撐相行走步態，並且再透過感測資訊直接對 CPG 神經元進行外界輸入的刺激來改變 CPG 輸出曲線，使得單腳支撐相時的行走曲線對於環境變化產生即時自適應調整，達成持續的穩定行走狀態。

為了將生物學上的 CPG 理念實現在控制系統上，CPG 大多被設計為一種神經網路模型。在此類神經網路模型中，一個神經網路可以透過多個神經元之間的互相抑制激發，使得神經網路模型產生振盪，也有人稱此為振盪器。在此我們以兩個神經元 N_1 和 N_2 互相連結為例，進行一個神經網路振盪器是如何產生振盪，首先 N_1 先進行放射在此同時 N_2 受到 N_1 的放射所抑制住，在此過程中 N_2 漸漸去極 (depolarization)，當達成一定閾值時， N_2 完成去極化從 N_1 中逃脫，接著 N_2 開始產生放射並同時對 N_1 進行抑制，而在此過程中 N_1 也是會漸漸去極，當達成一定閾值時， N_1 完成去極化從 N_2 的抑制中逃脫再繼續開始放射，藉此兩個神經元狀態達成週期性的互換，因而產生出有循環振盪的神經網路模型。

在許多 CPG 神經網路的數學模型中，例如:Hopf 振盪器[5][6]、McMillen 振盪器[7]或是 Matsuoka 振盪器[8][9]...等數學模型都可以產生出有循環振盪的神經網路模型，因為 Matsuoka 振盪器相對於其它的振盪器模型較為簡單且也被廣泛的利用在機器人的應用上，所以我們在此利用 Matsuoka 振盪器提出的 CPG 神經網路的數學模型設計行走步態。如圖 1.2 所示，對於 Matsuoka 振盪器的數學模型，利用方程式描述如下:

$$\tau_1 \dot{u}_1 = c - u_1 - \beta v_1 - \gamma [u_2]^+ + \sum h_j [g_j]^+ \quad (1.1)$$

$$\tau_2 \dot{v}_1 = [u_1]^+ - v_1 \quad (1.2)$$

$$\tau_1 \dot{u}_2 = c - u_2 - \beta v_2 - \gamma [u_1]^+ - \sum h_j [g_j]^- \quad (1.3)$$

$$\tau_2 \dot{v}_2 = [u_2]^+ - v_2 \quad (1.4)$$

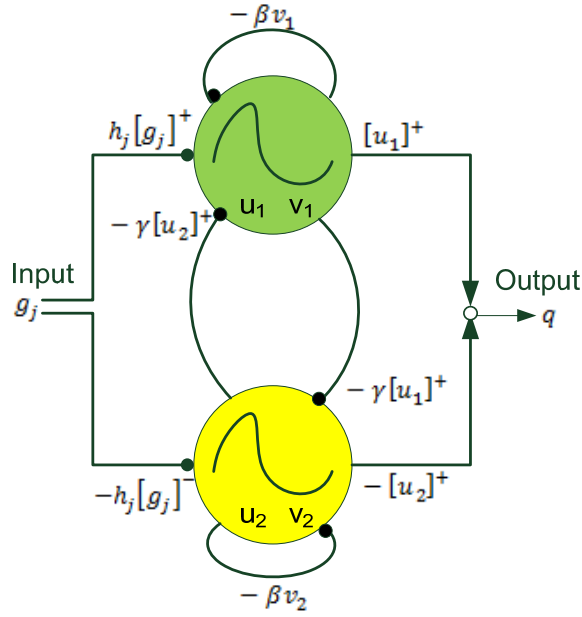


圖 1.2 振盪器的模型圖[18]

$$q = [u_1]^+ - [u_2]^+ \quad (1.5)$$

$$[x]^+ \stackrel{\text{def}}{=} \max(0, x), \quad [x]^- \stackrel{\text{def}}{=} \min(0, x) \quad (1.6)$$

其中 u_1 和 u_2 屬於膜電位， v_1 和 v_2 屬於自我抑制效果的變量，以上四個變量都屬於振盪器的內部狀態， u_1 和 v_1 為一個神經元之狀態變數； τ_1 , τ_2 , β , c 和 γ 為常數，其中 τ_1 , τ_2 , β 和 γ 值的關係會影響到振盪器是否振盪和波形[8][10]， c 的大小影響輸出的振幅； g_j 為輸入， h_j 為輸入的比例因子； q 為輸出。公式(1.6)表示符號 $[x]^+$ 和 $[x]^-$ 個別表示取 0 和 x 中最大正值和 0 和 x 中取最負值。

公式(1.1)和(1.3)屬於同一功能之公式，以公式(1.1)為例，此式屬於計算神經元之膜電位 u_1 的值，在式中先將 u_1 移到左邊，此時此式將呈現出膜電位之位置與乘上一時間常數 τ_1 之膜電位的變化速率將被得到，透過利用右邊參數的相加，其中 c 用來測定 CPG 輸出的振幅，而乘上一適應常數 β 之 v_1 是為了使 u_1 可以產生自我抑制，另外乘上一控制常數 γ 之 u_2 是為了使 u_1 與 u_2 之間存在著相互抑制，最後對於外來的輸入 g_j 被安排在式(1.1)中當作刺激神經元(取正部分)的輸入、在式(1.3)中則當作抑制神經元(取負部分)的輸入。

公式(1.2)和(1.4)屬於同一功能之公式，以公式(1.2)為例，此式屬於計算要產

生自我抑制神經元的微分公式，在式中先將 v_1 移到左邊，此時可以看出自我抑制神經元 v_1 與乘上一時間常數 τ_1 之 v_1 微分將透過同一神經元之膜電位 u_1 的正部分產生變化而獲得此時之自我抑制神經元 v_1 ，再藉此值回授到(1.1)式中讓膜電位 u_1 產生抑制的效果。

公式(1.5)則為此神經振盪器中，兩個神經元相互抑制之間膜電位的關係輸出，在此式中皆取膜電位 u_1 和 u_2 為正值之部分，藉由相加得到最後由兩個神經元所產生之整體神經振盪器的輸出曲線。

在此我們對 Matsuoka 神經振盪器模型的輸出特性[8][10]進行探討，其中模型中的參數和振盪器的輸入與振盪器的輸出之間的影响關係為：

第一、 在外界的輸入為零時($g_j=0$)，為了使得 CPG 可以保持持續得振盪，參數的調整皆須滿足公式(1.7)和(1.8)；在滿足公式(1.8)的情況下，圖 1.3(a)為未滿足公式(1.7)、圖 1.3 (b)為有滿足公式(1.7)；在滿足公式(1.7)的情況下，圖 1.4(a)為未滿足公式(1.8)、圖 1.4(b)圖為有滿足公式(1.8)。圖 1.3(b)和圖 1.4(b)雖都滿足公式(1.7)和(1.8)，但因為參數設定上仍然有所不同，因此 CPG 輸出之曲線並不相同。另外，公式由於公式(1.7)和(1.8)皆須滿足才能讓 CPG 保持持續振盪，因此對於兩個公式皆不滿足的情況，我們在此不作討論。

$$\gamma/(1+\beta) < 1 \quad (1.7)$$

$$\gamma > 1+(\tau_1/\tau_2) \quad (1.8)$$

在[8]的研究中，證明出公式(1.1)和(1.3)的微分方程式在一範圍內至少存在一穩態解，而當方程式之解處於穩態解的狀態下時，振盪器之輸出不會產生振盪，因此透過公式(1.7)和(1.8)的參數值設定限制，能使公式(1.1)和(1.3)在穩態範圍外，處於非穩定靜止狀態，並且因公式(1.1)和(1.3)為有界性(Bounded)，因而產生的非穩態解會在一定範圍內持續振盪。

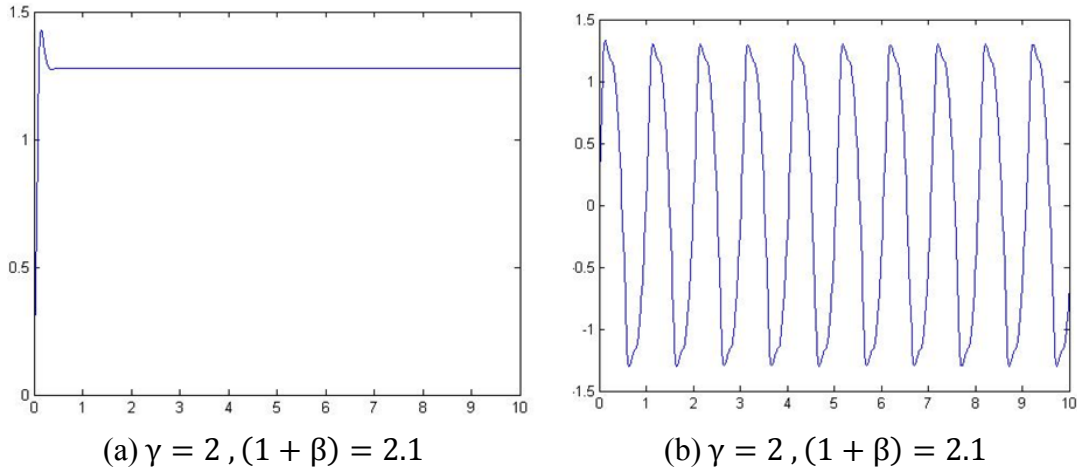


圖 1.3 對於有無滿足公式(1.7)的 CPG 輸出曲線變化圖(a)無滿足公式

因此不會振盪(b)有滿足公式因此保持週期性振盪

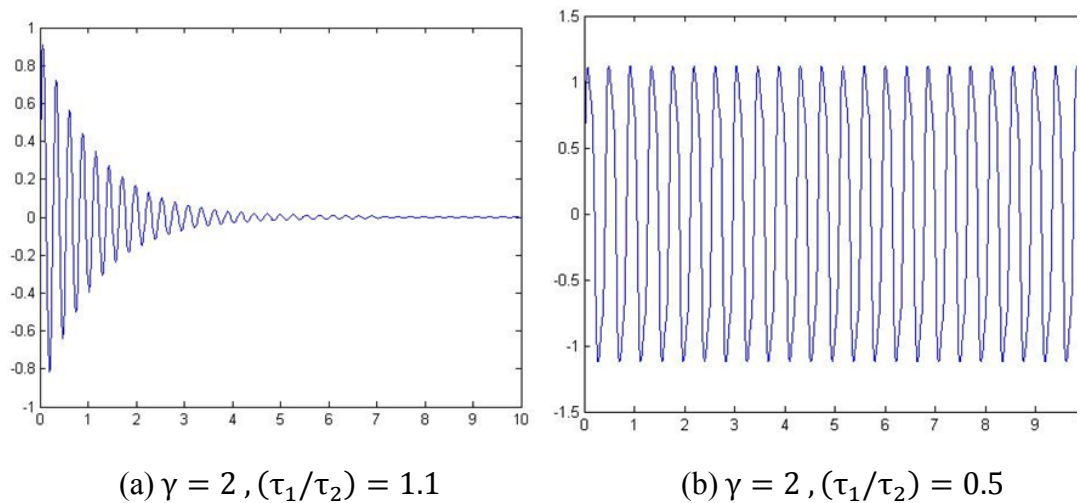
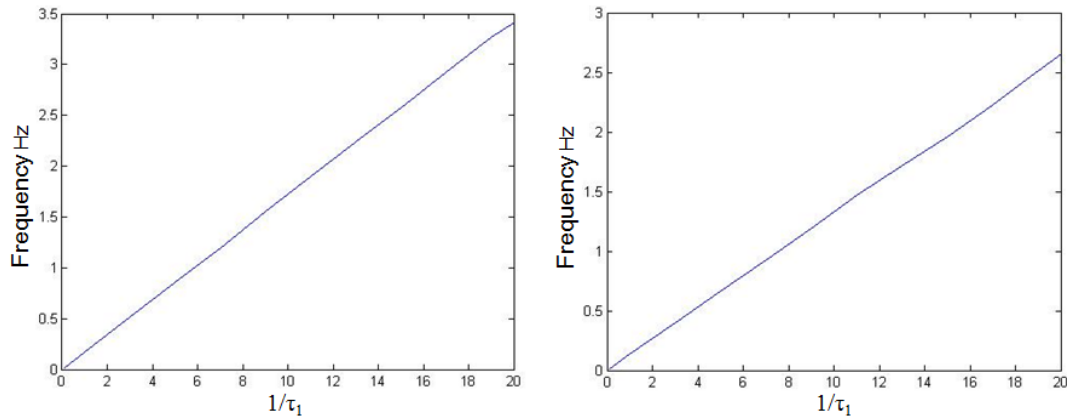


圖 1.4 對於有無滿足公式(1.8)的 CPG 輸出曲線變化圖(a)無滿足公式因此

振盪時振幅依次縮減至 0 (b)有滿足公式因此保持同振幅的週期性振盪

第二、當外界的輸入為零時($g_j=0$)，CPG 此時若滿足振盪條件產生持續的振盪，那麼它本身的自然頻率與 $1/\tau_1$ 成一定比例，如圖 1.5 所示，在此圖中我們利用兩個不同比例的 τ_1/τ_2 來進行驗證，因為 τ_1/τ_2 的不同比例值會直接影響到 CPG 輸出波形的振幅大小與自然頻率的高低，所以由圖中顯示出在不同的 τ_1/τ_2 的比例值中，CPG 輸出波形的自然頻率皆與 $1/\tau_1$ 成一定比例，只是其中的比例值會有所變化。



(a) $\tau_1/\tau_2=0.8$, slope=0.17065

(b) $\tau_1/\tau_2=0.5$, slope=0.13265

圖 1.5 CPG 輸出曲線的自然頻率與 $1/\tau_1$ 的關係曲線圖

第三、 CPG 神經振盪器在有輸入時，輸入與輸出之間存在著牽引作用。當輸入的頻率與自然頻率有相位差時，且輸入振幅大到可以完全影響 CPG 時，振盪器的牽引作用可以使得 CPG 的輸出頻率與輸入頻率可以追到幾乎相同，只是在輸入和輸出之間會有一點相位延遲，如圖 1.6(c) 所示；但是如果是在輸入的訊號振幅太小時，牽引作用就無法作用，因而不影響到 CPG 的自然頻率，這即屬於一種抗干擾的特性，如圖 1.6(a) 所示；另外若是輸入訊號振幅大小介於是否影響 CPG 之間時，自然頻率整體上不會受到太大影響，但是每個週期之間仍會有頻率變得較快或較慢產生，只是輸出振幅會受到影響隨著輸入放大縮小，如圖 1.6(b) 所示。

第四、 當輸入為一固定常數，此時輸入常數的大小會立刻影響到輸出振幅的大小，我們即可利用此特性進行 CPG 曲線的輸出曲線控制，但由於輸入對輸出的影響不是呈現一個有比例性的放大或縮小，由圖 1.7(a)(b) 比較所知，且當輸入值有一個較大的改變時，此時輸出的第一週期振盪的峰值會有一個超越量產生，如圖 1.7(b) 圖上說明所示，

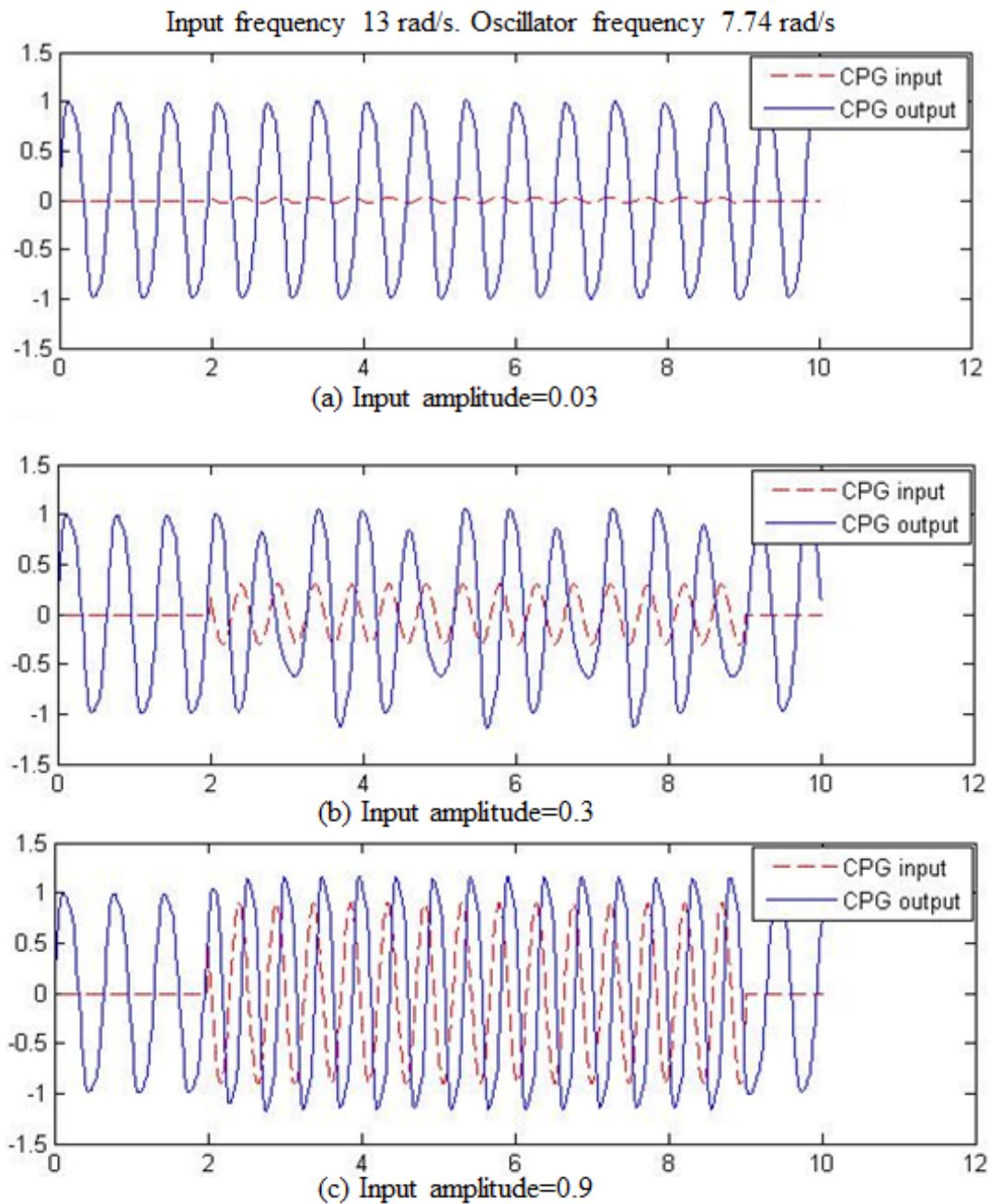


圖 1.6 在輸入振幅改變時，CPG 輸出的對應變化關係圖

因此由於有以上兩點的關係，在控制應用上的對於輸入利用的參數上的考量將有一定的限制。另外，如果輸入振幅太大會立刻使得輸出曲線不再繼續振盪，輸出曲線成一直線，此性質稱為抑制作用，如圖 1.7(c) 所示，此一特性使得我們在輸入值上須加以限制使得維持 CPG 曲線的週期性，但若是善加利用此特性，可以產生將週期性任務中斷或暫停的功能。

經由上述四點對於 Matsuoka 神經振盪器模型內的參數和不一樣的輸入訊號對於振盪器輸出曲線的影響，以及在兩者之間相關聯的特性說明之後，讓我們在將 CPG 振盪器應用到雙足機器人步態行走的控制設計上，能利用 CPG 的特性“牽引特性”達成感測器回授，或是利用 CPG 參數的調整使滿足 CPG 的振盪公式且同時能避開 CPG 的限制“抑制作用”、“超越量的產生”，也讓我們知道在整體的控制設計上需要再加入何種控制策略或是補償工具，達成與 CPG 振盪器應用在雙足步態行走上有兩者相輔相成之效果。

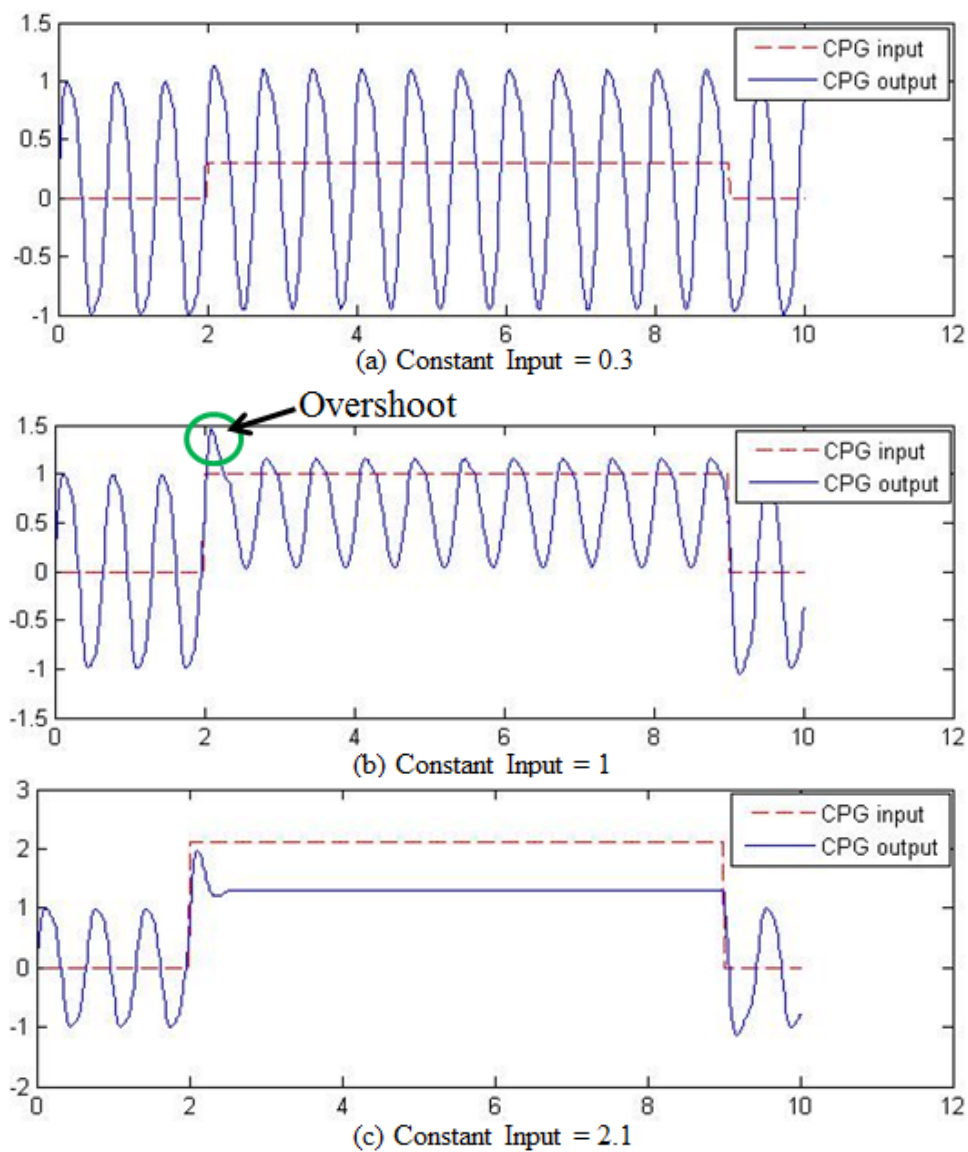


圖 1.7 當輸入為固定常數時，CPG 輸出的對應變化關係圖

1.3 相關研究回顧

機器人基於環境偵測的回授控制設計上，在 Fukuoka 等人發表的文獻[11]表示目前大致上分為兩大類，第一類是基於生物觀點的運動生成方式，雖然控制運動的神經系統和機器人的機械系統各別屬於一個非線性動態系統，但在此方式中將兩個系統進行耦合，使得神經系統可以直接由機械系統取得環境資訊對環境進行即時的適應，如圖 1.8(a)所示，而此方式被稱為耦合動態系統，在此方式中建構神經系統需要很縝密才能同時滿足所需的理想運動曲線和達成環境變化下的因應對策，所以對於其中的參數設定就會變得過於複雜且較難達成完美估測。第二類則屬於較傳統的控制系統，例如系統中有一個阻抗控制，與機器人的機械系統進行耦合成為整體的耦合動態系統，但在此最大的不同是系統的運動產生器不包含在整體的耦合動態系統中，如圖 1.8(b)所示，作者表示這樣會發生當耦合動態系統根據環境狀況產生出的自主規劃，會在耦合外的理想運動曲線和目前的實際運動之間有互相抵觸的嚴重問題產生，但在此研究者大多只要加入適當的控制器進行整合即可避免此問題的產生。

上述當中對於機器人產生運動方式之分類介紹，在雙足步行機器人的平衡控制上亦是如此，因此我們將步態規劃以此兩大類-傳統機器人系統和神經系統，分別依序介紹，並且對它們各別如何控制平衡穩定行走，進行不同方法的探討。

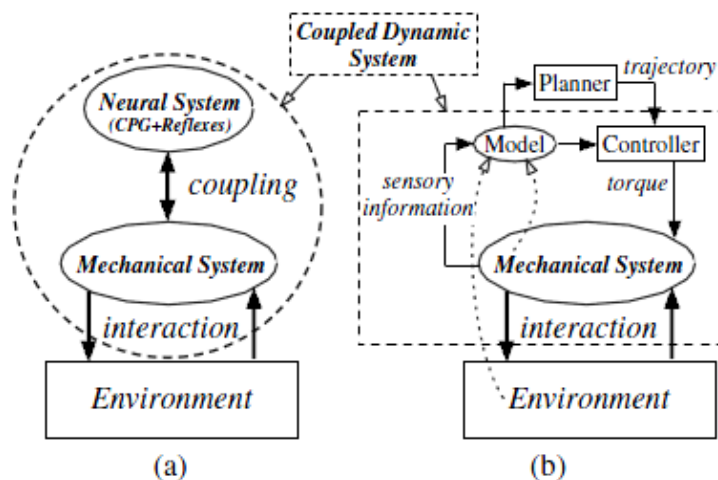


圖 1.8 說明運動產生方式，藉由(a)神經系統(b)傳統機器人系統[11]

在 Ali 等人發表的文獻[14]中，提出一方法乃是完全利用基於逆運動學演算法的座標位置和座標之間的相對旋轉角度，來使雙足機器人可以行走在有傾斜的表面上，此控制方法利用全身姿態改變來達成對地形的適應，並非只改變一個關節馬達的角度(如:踝關節)達成適應地面。首先，這策略是把位置和定向分開來看，但都包括在運動學計算中，且預先定義當在平地時旋轉的定向角設為0度，接著，利用旋轉定相角得到合適的角度來行走在傾斜的平面上。那他們所使用的運動模式是利用零力矩和點(Zero Moment Point, ZMP)[13]來控制機器人整體 CoM(Center of Mass) 的位置和擺盪腿的位置，而當雙足機器人走在與定義的旋轉的定向角 0 度不一樣時，即代表旋轉定向角發生改變，此時 CoM 的曲線會因為定向角的改變而透過逆運動學的運用，使得 CoM 曲線也跟著進行修正，因而得到一個新的位置根據旋轉定向角。而對於傾斜地面的機器人的姿態修正分為雙足機器人的兩個平面:前向平面(Frontal Plane)和側向平面(Lateral Plane)，如圖 1.9 顯示，分別進行各自方向的矩陣運算來達成修正。

在 Yi 等人發表的文獻[12]中，機器人行走時上半身是利用 ZMP 進行控制移動，而在行走時的即時控制，利用了腳底的壓力感測器進行每一步的步伐落地時間調整和對於落地時的衝擊力進行緩衝的補償控制，在最主要針對地形環境變化的控制部份為透過腳位置方向的變化來得知，首先利用每個關節自由度的編碼器

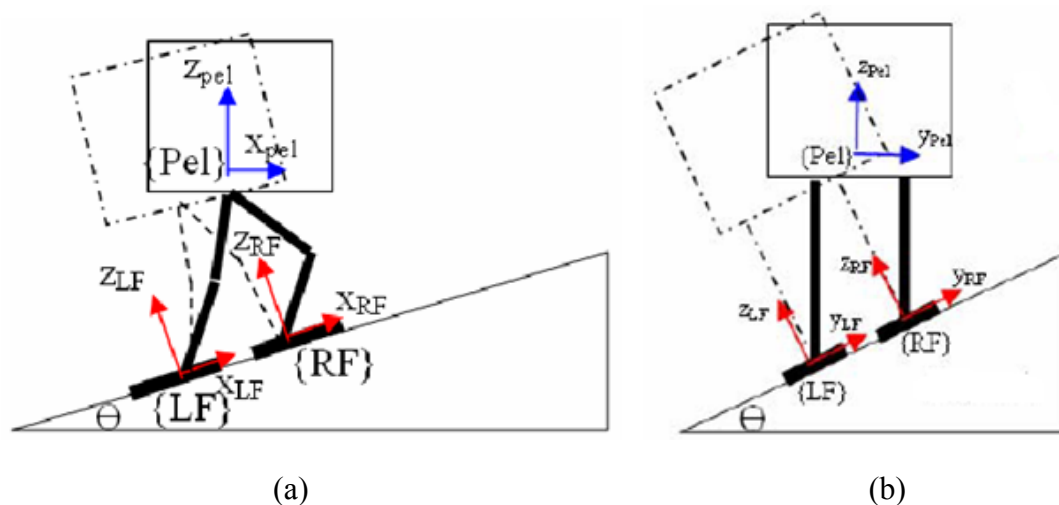


圖 1.9 機器人姿態修正示意圖各別在(a)前向平面 (b)側向平面[14]

得知目前角度，並透過運動學運算得知雙腳底的姿態再搭配上半身的姿態感測得到 inertial measurement unit(IMU) 機器人上半身的角度方向，來算出兩腳底位置之間的移位方向，以及下一步支撐腳目前對於地面的正交方向，以此來偵測出地形是否有產生變化，如圖 1.10(a)顯示，再利用文中提出的學習算法減少雜訊影響估測出較精確的新地形表面梯度，如圖 1.10(b)顯示，而得到成功的機器人移動控制平衡。

在 Ogino 等人發表的文獻[15]中，作者先對上下坡機器人會跌倒的情況進行分析，如圖 1.11 所示。

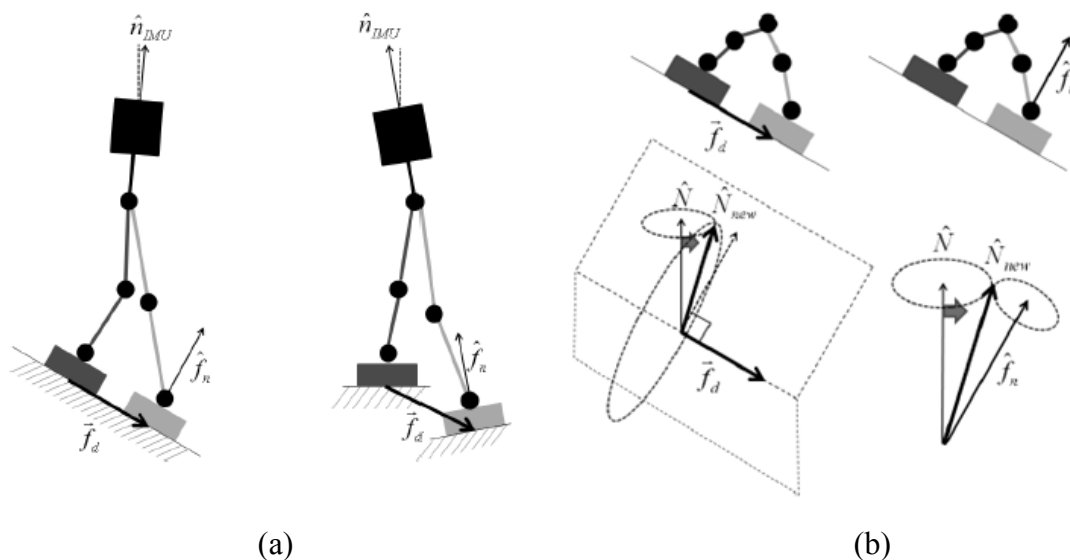


圖 1.10 (a)地面感測(b)從雙足位移及正交方向估測出新的表面梯度[12]

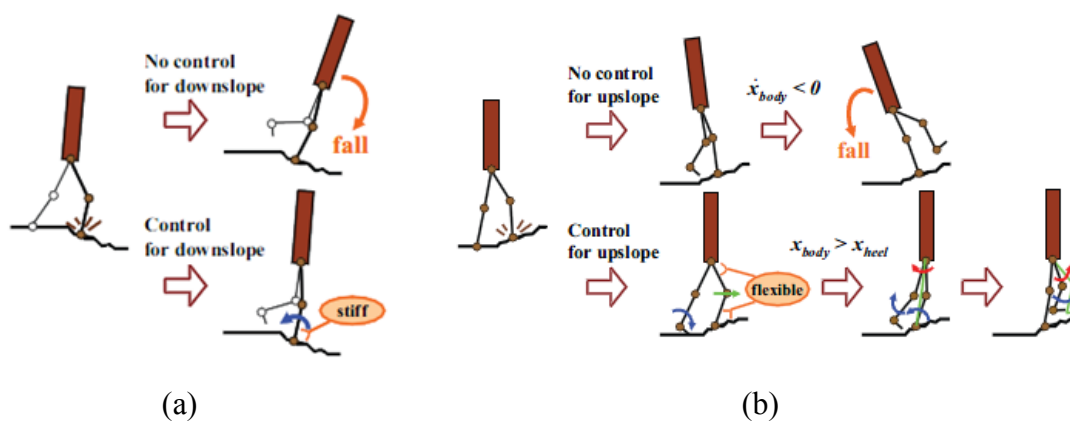


圖 1.11 機器人跌倒情況分析(a)下坡情況(b)上坡情況[15]

接著，作者提出的行走控制器在不用感測地形狀態之下使機器人行走在崎嶇地形上，此行走控制器的應用方式為，分析對於機器人軀幹的速度、機器人各部位相對位置、腳底得到的作用力和行走步態在哪一階段，判斷此時須因應的解決方法，如圖 1.12 所示，而此方法利用的是控制上半身的速度和踝關節馬達的鎖馬達程度(stiffness)，最後控制每個關節自由度的力矩來達成。

在 Li 等人發表的文獻[16]中，行走步態是利用被動倒單擺產生，面對地形改變的情況下，利用上半身姿態感測資訊和腳的力與力矩感測資訊，進行身體姿態控制、理想 ZMP 控制和非線性著地控制，最後輸出成每個關節自由度的位置向量來達成，圖 1.13 為綜合上述之行走控制架構圖。

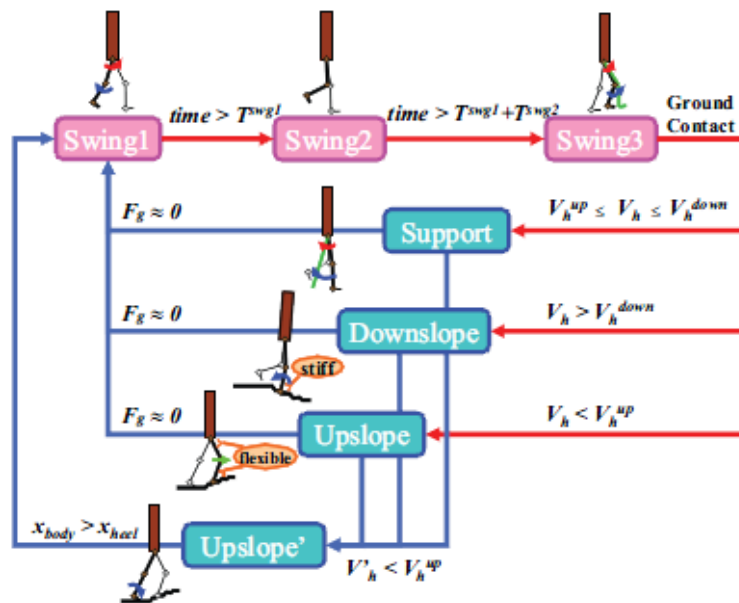


圖 1.12 崎嶇地面行走時的整體控制流程[15]

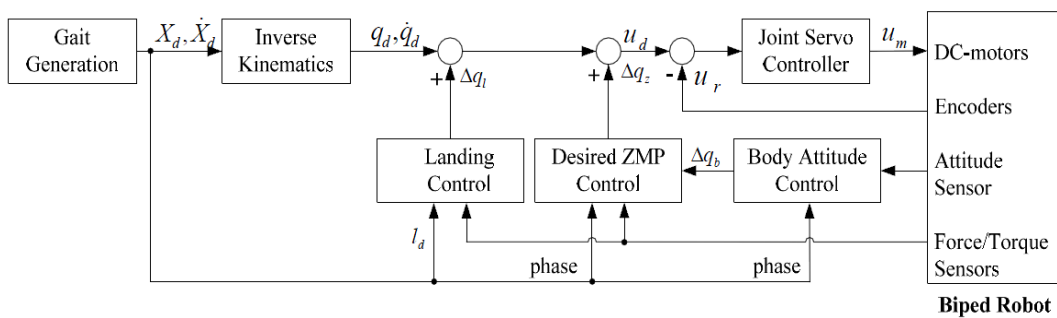


圖 1.13 行走控制之架構圖[16]

在 Inada 等人發表的文獻[17]中，整體的行走曲線是由 CPG 的輸出安排得出，方法為把八個 CPG 輸出安排在八個關節直接控制所需轉的角度，且每個關節中皆有權重互相牽制影響，如圖 1.14(a)所示，而對於各個關節所需的 CPG 輸出曲線在此則是先擷取人類的行走曲線換算成該關節所需的角度形成連串曲線，如圖 1.14(b)所示，再利用基因演算法計算出 CPG 所需的參數，藉此得到關節所需的 CPG 輸出曲線，完成雙足機器人行走。

利用文獻[17]此類把 CPG 安排在各個關節上所需的 CPG 參數就有 76 個，而為了得到這些參數所使用的基因演算法就又要 271 個參數，參數實在是使用過多，雖然有些參數值是利用演算法自動生成，但仍會導致運算太過複雜，對於實際上要即時利用感測資訊，進行感測回授運算，時間運算上可能會來不及。

在 Endo 等人發表的文獻[18]中，作者減少大量的 CPG 的數量和所需調整的參數，他們將 CPG 安排來產生行走曲線時所需跨的長度和高度，整體使用了兩個 CPG 且之間也沒有權重連結，大大的降低 CPG 參數的數量，使得回授運算可以即時的運算，並且在行走時即時的把感測資訊回授到 CPG 中，對於往前跨長度的 CPG 用上半身在 Roll 方向的角度來控制，而跨步高度則是用上半身在 Roll 方向的角度和腳底的壓力感測值來控制，如圖 1.15 所示。

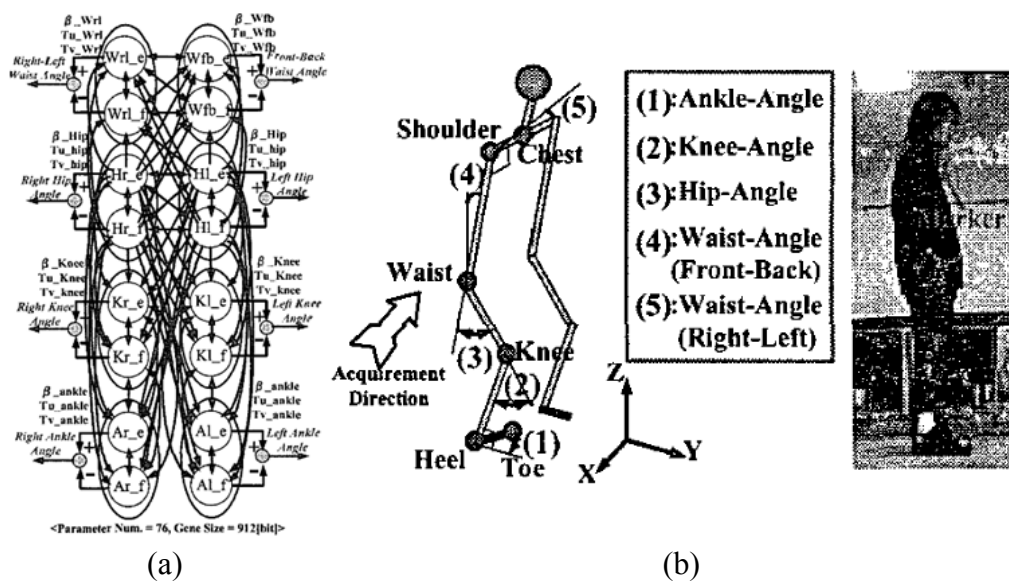


圖 1.14 (a)整體 CPG 網絡 (b)與人相對的關節比較示意圖[17]

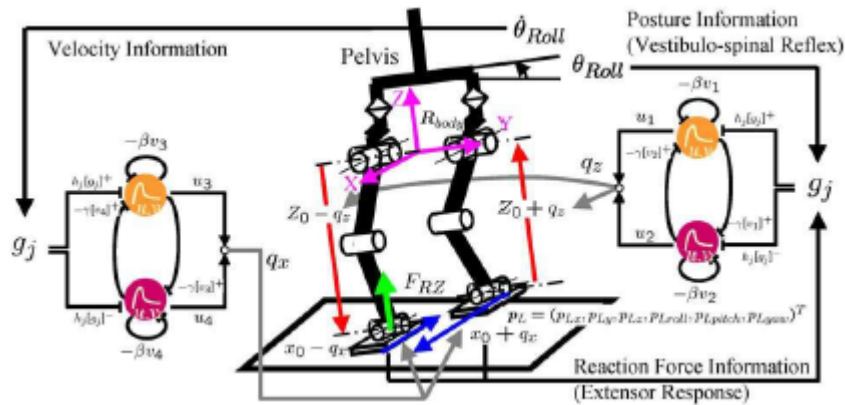


圖 1.15 利用 CPG 的整體控制架構[18]

1.4 問題描述

對於雙足步行功能的最大課題之一就是面對不同的環境地形，機器人都能成功地行走或跨越。對於雙足步行上要保持穩定平衡所需面臨的問題有：第一、對於雙足機器人有較為重的上半身時，步態需如何規劃才得以維持平衡的行走；第二、當外界地形環境發生改變時(例如：上下坡與平地之間的轉換)，該如何進行應對；第三，如何利用自身的感測系統來察覺環境情報和機器人本身的姿態狀況，並利用即時的回授控制，保持行走的穩定平衡。我們希望經由克服這三類的問題，使得雙足機器人可以穩定的在環境中行走。

首先，在設計行走步態上，我們希望能透過可自我產生連續週期的 CPG 來規劃理想的行走步態，並利用感測資訊回授改變 CPG 曲線來控制行走時上半身的擺盪控制，而當中對於較重的上半身我們將加入雙腳支撐相時的上半身重心移動和單腳支撐相時的上半身的擺盪，藉此保持平衡。

接著，在對於地形資訊和機器人本身的姿態資訊偵測上，我們希望可以用最一般化的感測器來取得，使得我們提出的方法可以用在商業化的機器人上，而機器人不需要將感測器特殊加裝在特定位置上來得到環境或機器人本身的資訊。

最後，對於整體的行走平衡控制上，除了將感測資訊直接回授到相關的 CPG 中，完成直接曲線改變之外，為了因應 CPG 的限制，我們將加入補償控制器，並針對不同時機的回授控制策略來達成連續行走的平衡穩定控制。

在論文中，我們希望完成一套方法使機器人能因應地板斜面改變而產生即時姿態調整的行走控制。

1.5 系統架構

圖 1.16 為本論文的系統流程圖，我們主要設計的區塊為以下兩個區塊:步行軌跡產生器(Walking Pattern Generator)和補償控制器(Compensation Controller)。藉由上半身姿態資訊與使用者所給的命令，在步行軌跡產生器的區塊中計算出目前所需的步行軌跡，再與透過上半身姿態資訊所得到在補償控制器中生成的補償動作，進行合成，得到雙足機器人行走時的每個自由度所需到達的目標角度，使雙足機器人能成功的在未知的斜面行走。

1.6 章節說明

本論文一共分為六章，第一章介紹研究動機、CPG 神經振盪元原理及其相關回顧並說明整體的系統架構。第二章介紹雙足步行步態的規劃設計。第三章則介紹平衡行走的整體系統設計。第四章為針對本論文實驗用之機器人 NAO 的介紹。第五章為模擬與實驗結果，驗證整體系統設計的可行性與正確性。第六章為結論與未來工作。

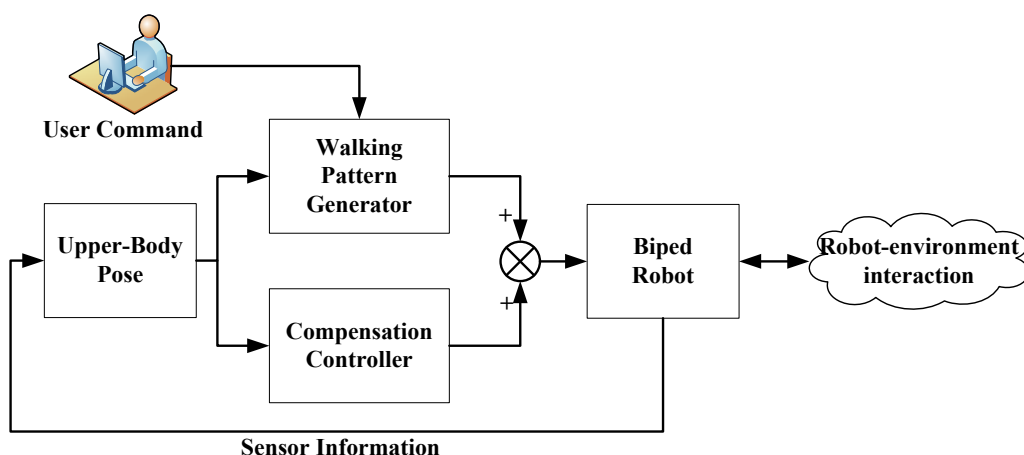


圖 1.16 系統流程圖

第二章 雙足步行步態規劃

為了使雙足機器人可以完成穩定的平衡行走，我們先建立可以在平地上行走的步態軌跡。在相關研究調查中，對於預先離線的理想行走曲線規劃理論基本上分為利用 ZMP[13]、倒單擺模型[19]、非線性模型(如拉格朗日動態模型(Lagrange dynamic model)[20])...等來達成，而這些理論現在也已被延伸研究出更多元的方法例如 Fictitious ZMP[13]、Eulerian ZMP[21]、被動倒單擺模型[16]...等，來達成行走控制。

對於上述行走曲線規劃的理論方法，在分析上可分為運動學(Kinematics)以及動力學(Dynamics)兩部分。以一個多連桿系統為例，在運動學的分析中，是計算各連桿旋轉角度或角速度所產生在機器人空間中的位移量，或是計算機器人在空間中位移量所產生在各連桿之旋轉角度或角速度的改變，此方法較常利用在產生靜態步行設計，即是考慮將機器人的重心投影全落在腳部的支撐多邊形區域內即可。在動力學的分析上則是須利用實際且精確的系統動態模型來得到連桿之速度、角速度、加速度、慣性力或扭力等等，而此方法在分析上較為精準，因此常利用在產生動態步行設計上，在行走時機器人的重心可以落在腳部的支撐多邊形區域外，而此時是利用機器人的運動慣量達成平衡，但動力學的分析過程也較為複雜[20]。

2.1 行走步態之規劃

在我們規劃的運動軌跡中，沒有利用複雜的控制系統來精確計算出步行速度、加速度及作用力矩等精確的物理模型使得利用運動慣量達成平衡，所以此行走軌跡是屬於靜態步行的設計，因此在設計理想的行走軌跡時，必須考慮要將機器人的重心投影完全落在行走時支撐腳的支撐多邊形區域內，在論文的實驗中也將透過 ZMP 的驗證，來證實行走的步態規劃是符合靜態步行的設計。

對於整體的行走步態規劃上，由於連續的行走是由左右腳個別的向前跨步循環而成，屬於週期性的對稱循環，因此我們將利用能自動產生出節律性週期運動

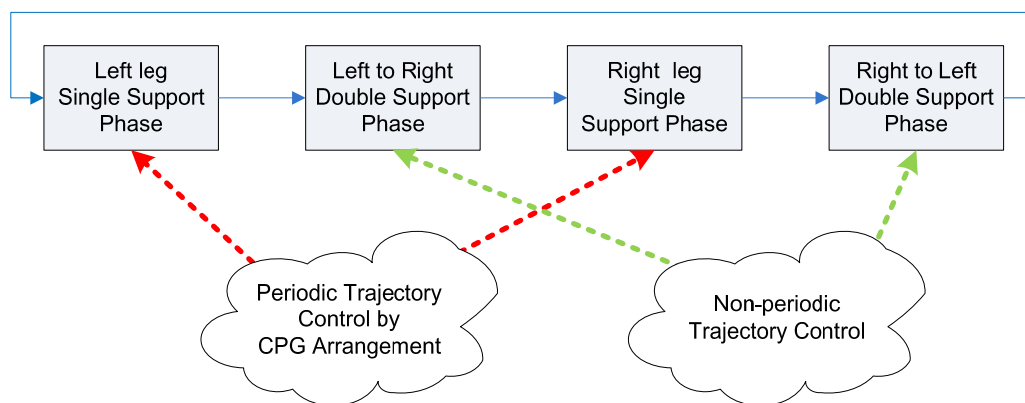


圖 2.1 整體的步態規劃

的 CPG 來設計行走曲線。但因為我們的設計行走屬於靜態步行的設計，對於機器人的重心投影必須落在腳步的支撐多邊形區域內，因此對於上半身的重心移動上，無法使用運動慣性力的方式達成重心的快速移動，會導致會有拖步的情況產生，因此在雙腳支撐相時我們加入非週期性的上半身重心移動曲線規劃，使得在跨步之前上半身重心可以已移動到支撐腳的多邊形區域中，達成穩定的行走軌跡。我們把整體步態規劃分成雙腳支撐相(Double Support Phase, DSP)與單腳支撐相(Single Support Phase, SSP)這兩部分來分開設計，如圖 2.1 所示，以完成機器人理想的連續穩態行走，在論文中我們定義一步(step)為 DSP 加上 SSP，一跨步(stride)則只表示單一 SSP，而行走了一週期則表示完成了兩步(two steps)的行走。

2.2 單腳支撐相

雙足機器人在行走中的單腳支撐相，左右腳的曲線擺盪是屬於週期性的對稱循環，使得雙腳是呈現左右對稱的方式進行向前擺動，接著再搭配雙腳支撐相的重心移動來達成行走的目的。

對於單腳支撐相時的左右腳曲線擺盪的週期性對稱循環控制，我們希望可以透過對 CPG 此方法的功能模擬實現出每一跨步，藉此得到週期性的理想行走步態，並且再透過感測器對於外界環境刺激的感測進行行走時的平衡控制。

在文獻中，無論是何種型態的機器人對於 CPG 應用在行走上時，大多都是利用一個振盪器分配給一個關節或是一個馬達位置[17][23][24]；或是多足機器人，

對於振盪器的分配為一隻腳一個振盪器，再藉由彼此的相位合作來完成行走[25]。以上這些方法，證明出對於他們可以藉由多個振盪器之間自我協調產生出所想要的步態，但是這些方法所使用的參數很多，對於參數的調整是一大挑戰，另外若是要用在實作上，在運算量及時間上也是一大考量。最後，因為此振盪器的分配方式在機器人自由度很多的情況下，對於外界環境所感測到的資訊是否有合理的應用在所需的回授控制上，達到移動也是一大課題。

基於上述的想法，我們希望可以用較少的 CPG 亦或是較少的參數調整來完成機器人的穩定行走，因此我們需要對 CPG 的輸出曲線進行功能安排設計，來產生出一個穩定的行走步態。Endo 等人提出 CPG 安排的設計方法[18] [26] [27]。以[18]為例，基本上是利用兩組振盪器的輸出來得到行走控制，一個用來產生雙腳在 Z 軸方向的位置控制來進行支撐腳和擺盪腳的轉換，另一個則是用來產生雙腳前端在 X 軸方向的位置控制來產生往前行走的跨步曲線，利用此種 CPG 安排方式得到可以穩定行走的曲線，並且此種方法所需的參數調整減少很多。

利用上述的概念，機器人平衡行走在單腳支撐相時所需的 CPG 安排設計被提出如下；我們設計四組神經振盪器來完成平衡行走，在此設計中我們把控制方式分成整體行走曲線轉換及單一關節角度兩種，分別各以兩組神經振盪器，來達成我們想要的控制目的。如圖 2.2 所示，雖然在我們對於 CPG 的安排設計中，相較於[18]所提出的 CPG 安排的設計方法多了兩組的神經振盪器，參數多了兩倍，看似整體的參數調整會變得困難複雜，但由於我們 CPG 所產生的曲線皆是要產生相同的圓滑曲線，並不因不同的控制目的來產生不同的曲線，其中當要調整神經振盪器曲線值時，會由感測資訊當輸入來完成曲線控制，另外當振盪器的自然頻率要改變時，由 1.2 節中對 CPG 神經振盪元原理分析中得知，只要在特定參數 $1/\tau_1$ 做等比例調整即可產生新頻率的 CPG 曲線圖，而此 CPG 所產生的圓滑曲線的振幅不會改變，所以在此只要調整出一組我們所需的圓滑曲線的參數即可，並不會增加參數上的調整的麻煩。

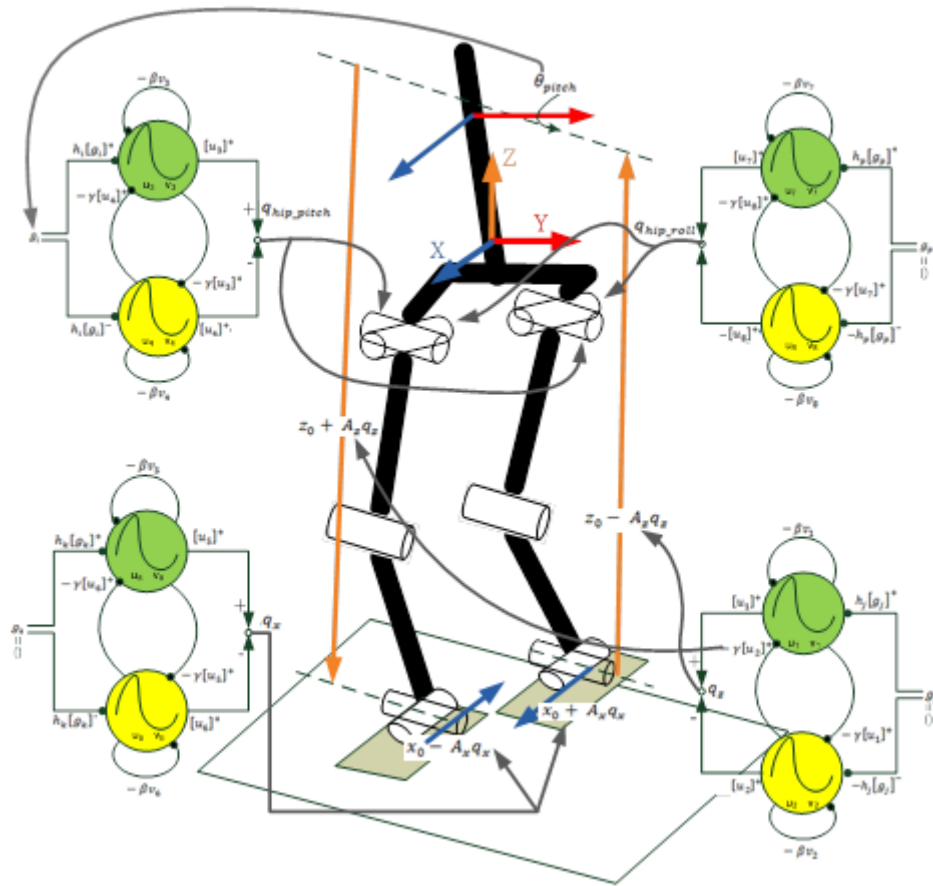


圖 2.2 四個振盪器的控制安排

接下來我們要對四個振盪器所要達成的目的及功能進行更詳盡的說明：

第一個振盪器顯示在圖 2.2 的右下方，被分配來控制機器人支撐腳的轉換及擺盪腳的抬腳高度，所以振盪輸出用來轉換成行走曲線在 Z 方向的位置座標移動，完成控制雙腳在 Z 軸垂直方向的位置高度，而此轉換公式為(2.1)、(2.2)

$$p_{Lz} = z_0 - A_z q_z \quad (2.1)$$

$$p_{Rz} = z_0 + A_z q_z \quad (2.2)$$

其中 p_{Lz} 、 p_{Rz} 各別為 Z 方向左右腳到肩部的長度， z_0 為 Z 方向被設定的初始長度， A_z 為比例因子對振盪器的輸出 q_z 進行等比例的放大成跨步時所需的高度，透過兩式之間有+-的相差，產生出左右腳個別行走時的抬腳高度。

第二個振盪器如圖 2.2 左下方所顯示，被分配來控制機器人向前行走時所邁開的步伐長度，所以振盪輸出用來轉換成行走曲線在 X 方向的位置座標移動，

完成控制雙腳在 X 軸前行方向的跨步長度，而此轉換公式為(2.3)、(2.4)

$$p_{Lx} = x_0 - A_x q_x \quad (2.3)$$

$$p_{Rx} = x_0 + A_x q_x \quad (2.4)$$

其中 x_0 為 X 方向被設定的兩腳皆著地時的步距， A_x 為比例因子對振盪器的輸出 q_x 進行等比例的放大成跨步時所需的長度的一半值， p_{Lx} 、 p_{Rx} 為行走時左右腳個別在 X 方向對於雙腳支撐相時的步距 x_0 的相差長度，因而產生出左右腳相對座標上的變化，再利用建立絕對座標進行相對座標的累積，產生出向前的跨步。

第三個振盪器如圖 2.2 左上方所示，我們利用它來使得在單腳支撐相時機器人上半身可以進行前後的擺動，其振盪輸出用來直接轉換成上半身前後搖擺的角度，控制位置在髖關節 Pitch 方向上的關節旋轉角度，此轉換公式為(2.5)

$$\theta_{HipPitch_Swing} = k_{hip_pitch} \times q_{hip_pitch} \quad (2.5)$$

其中 k_{hip_pitch} 為比例常數對振盪器的輸出 q_{hip_pitch} 進行等比例的放大成行走時上半身前後擺盪所需的的角度， $\theta_{HipPitch_Swing}$ 為直接相加到控制髖關節 Pitch 方向之馬達的擺盪角度值。

第四個振盪器如圖 2.2 右上方所示，我們利用它來使得在單腳支撐相時機器人上半身可以進行左右的擺動，其振盪輸出用來直接轉換成上半身左右搖擺的角度，控制位置在髖關節 Roll 方向上的關節旋轉角度，此轉換公式為(2.6)

$$\theta_{HipRoll_Swing} = k_{hip_roll} \times q_{hip_roll} \quad (2.6)$$

其中 k_{hip_roll} 為比例常數對振盪器的輸出 q_{hip_roll} 進行等比例的放大成行走時上半身左右擺盪所需的的角度， $\theta_{HipRoll_Swing}$ 為直接相加到控制髖關節 Roll 方向之馬達的擺盪角度值。

安排前兩個振盪器的輸出經由公式轉換成行走時，在單腳支撐相時擺盪腳所需的跨步高度與跨步長度，及支撐腳所需的支撐高度。以左腳為支撐腳為例，此時座標位置的基準點在左腳腳底，利用 p_{Lz} 得知肩膀的高度，利用 p_{Rz} 再得知右腳擺盪腳腳底的擺盪高度($p_{Lz} - p_{Rz}$)，並透過 p_{Lx} 、 p_{Rx} 的相差距離，得雙腳在 X 方向的相關距離，並透過累加運算得到擺盪腳往前跨步時的相關位置($p_{Rx} - p_{Lx}$)，

而此時座標位置的 Y 方向我們將設定為固定常數。因此得到左右腳和肩部的世界座標(world_x, world_y, world_z)，假設左腳底的基準點位置為(0,-50,0)，肩部位置則為(0,-50, p_{Lz})，右腳腳底位置是($p_{Rx} - p_{Lx}$,50, $p_{Lz} - p_{Rz}$)，將此三個座標位置代入逆運動學中，即可得到行走跨步時每個關節所需到達的角度。

另外，當單腳支撐相開始時，雖然我們已經加入了雙腳支撐相來使得機器人軀幹的重心已移動到支撐腳上，但是由於當擺盪腳抬起時，整體機器人的平衡又會改變，導致整體機器人重心會往抬起的擺盪腳移動，發生行走時不穩或拖步的情況。

為了克服上述情況，我們希望在單腳支撐相時加入機器人上半身的擺動，使得當擺盪腳舉起時可以持續的保持穩定平衡的行走。在相關研究[28]中作者提出一個控制概念，即當有較重軀幹的雙足機器人在行走時，在雙腳著地時軀幹往前傾、接著在單腳支撐相搖擺的腿逐漸由後往前時軀幹逐漸往後傾。我們將[28]的控制概念進行修改，提出一個控制策略為，在雙腳著地時軀幹為直立的、接著在擺盪腳逐漸由後往前時，軀幹此時有兩個運動同時發生，其一為在側向平面(Lateral Plane)的方向上，軀幹往非擺盪腳的方向進行轉動，接著在轉回為直立，如圖 2.3(a)所示；另一為在前向平面(Frontal Plane)的方向上，當擺盪腳由後到與支撐腳齊平時，軀幹此時先由直立往前轉動再轉回為直立，接著當擺盪腳從齊平到向前落地時，軀幹再接著從直立往後轉動再轉回為直立，如圖 2.3(b)所示。

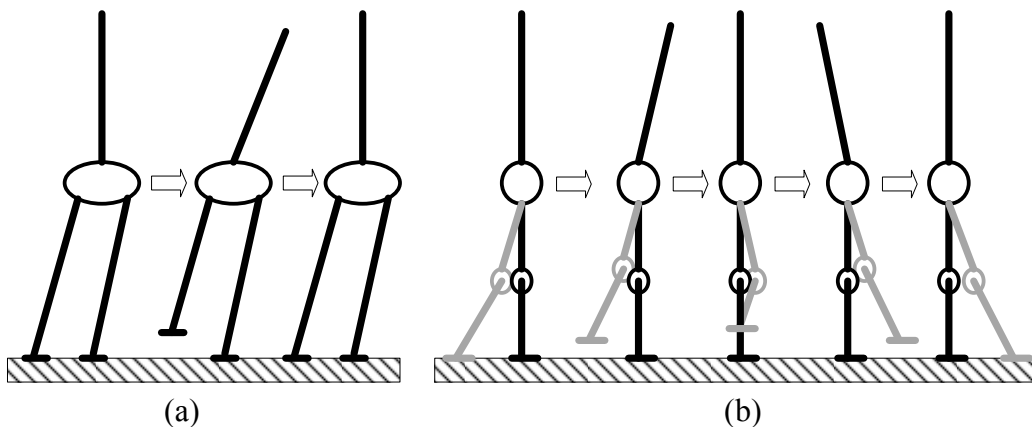


圖 2.3 (a)前向平面上的軀幹搖擺示意圖(b)側向平面上的軀幹搖擺示意圖

運用以上的方法，希望可以在單腳支撐相，當擺盪腳抬起時，減少重心的不穩定移動，讓重心可以持續保持在支撐腳上，使得行走時不會產生不穩或拖步的情況，完成在平地上理想的平衡穩定行走。所以我們在設計上再加入後兩個振盪器，產生輸出可以各別直接在髖關節的 Roll 和 Pitch 方向的關節旋轉角度上，加上我們想要讓上半身擺盪的控制角度，來完成此一控制策略。

2.3 雙腳支撐相

我們規劃的理想行走曲線為靜態步行，所以無法只靠單腳支撐相的運動慣量移動來完成瞬時的重心移動，因此我們在支撐腳要在左右腳中進行轉換時，加入雙腳支撐相先進行上半身的重心移動，當重心移動到下一跨步的支撐腳上，再繼續執行單腳支撐相，完成連續行走的週期循環。

在雙腳支撐相中，如之前所述就是要將上半身的重心移動到下一步行走時的支撐腳上，此時雙腳腳底座標位置皆不變，只針對肩部座標位置進行移動，因此在肩部座標移動曲線設計上，將機器人 X 方向(前後)和 Y 方向(左右)兩部份利用個別的理想曲線移動，完成機器人的上半身重心移動，Z 方向(高度)部分則在雙腳支撐相中不做改變。

上半身重心曲線在 Y 方向移動的公式(2.7)，以上半身重心要從右腳移到左腳為例，此公式目的為讓上半身軀幹可以平滑移動到設定在下一支撐腳的理想目標點上，達成平穩的重心移動。

$$p_{Sy} = 2(p_{Ly} - p_{Ry})t_s + p_{Ry}, 0 \leq t_s \leq T_{Two_Support} \quad (2.7)$$

其中 p_{Sy} 為機器人肩部中心在 Y 方向的座標位置， p_{Ly} 為左腳腳底中心在 Y 方向的座標位置， p_{Ry} 為右腳腳底中心在 Y 方向的座標位置， $T_{Two_Support}$ 為雙腳支撐相時的整體時間， t_s 為雙腳支撐相中目前的即時時間。

$$p_{Sx} = k_{Sx}(1 - e^{-\frac{t_s}{k_T}}), 0 \leq t_s \leq T_{Two_Support} \quad (2.8)$$

其中 p_{Sx} 為機器人肩部中心在 X 方向的座標位置，時間常數 k_T 為控制輸出曲線的波形，藉此控制上半身重心在 X 方向移動的快慢，常數 k_{Sx} 用來調整輸出曲線的

終點位置，使得機器人肩部中心在 X 方向的座標位置最終會停在前支撐腳的支撐範圍中。

上半身重心曲線在 X 方向移動的公式為(2.8)，利用此公式是希望在移動到目標點時，上半身可以在最後在 X 方向移動較為緩慢，使得感測器所得到的感測資訊可以更加精準，進而讓回授的補償值更加準確。會有此設計概念是，當機器人行走在平地上，此時在雙腳支撐相時機器人上半身的理想姿態應為直立的，但因為在單腳支撐相時此時機器人剛從後仰姿態變回直立，此時若使用與公式(2.7)相同的方式來移動，整體姿態會保持等速緩慢的修正，導致整體姿態會因為後仰的重心影響沒有修正到理想化，加上感測資訊在傳回時會有所延遲，使得得到的即時感測資訊會有所偏差，影響到平衡行走系統的設計，所以我們在此設計利用公式(2.8)來讓機器人在雙腳支撐相的剛開始可以有一加速度的移動，導致此時整體姿態能進行快速的修正，克服後仰時所帶來的重心影響，到最後在準備要量測下一步的感測資訊時，也因為此時的移動較為緩慢許多，減少了因快速移動而產生的多餘雜訊，使我們得到更準確的感測資訊，完成我們為了平衡行走的控制設計。

由上述得知，在雙腳支撐相時利用兩個肩部座標移動曲線，我們得到機器人肩部 X 方向和 Y 方向兩部份利用個別的理想曲線移動，Z 方向部分則為固定常數，完成機器人的上半身重心移動，而此時兩腳腳底位置皆是固定在同一位置上。以上半身重心要從右腳移到左腳為例，在左右腳底和肩部的世界座標位置 $(world_x, world_y, world_z)$ ，此時先假設後腳-右腳底座標位置在 $(0,50,0)$ ，因此在前腳-左腳底位置被假設為 $(50,-50,0)$ ，則肩部位置則為 $(p_{sx}, p_{sy}, 415)$ ，將此三個座標位置代入逆運動學中，即可得到上半身重心移動時每個關節所需到達的角度。

第三章 在不平地行走之補償設計

在完成了整體理想行走曲線的設計之後，我們必須再利用感測器所得到的感測資訊來加入回授控制，進行即時地調整行走的運動軌跡，使得機器人可以在上下坡行走時，仍可以保持穩定的平衡行走。

3.1 系統設計理念

人們在閉眼走路時，由於不知道下一步地形的狀態為何，會將重心保持在支撐腳上，直到擺盪腳落地得知地形為何後，此時上半身往下一步支撐腳前進時，立即已經針對此時預測出的地形狀況進行平衡修正，而人們對於平衡上的控制，大體上為調整上半身的姿態來達成，且當在單腳支撐相時若仍然發生不穩的情況，依然可以即時的透過上半身的姿態修正，防止跌倒。因此我們整體的補償控制器，將以此人們閉眼行走的概念來進行設計。

在多種雙足機器人行走的研究當中，對於在進行回授控制時，用何種感測器取得感測資訊，再轉換成目前環境的資訊，有著多元的設計方法。所用的感測器，有裝在腳底板上的壓力感測器、裝在機器人軀幹中心的陀螺儀與加速規、或是裝在各個自由度馬達上的力感測器或編碼器...等，來得知目前機器人本身與環境的感測資訊。

在上述的感測裝置上，我們採用的感測器為裝在機器人軀幹中心的陀螺儀與加速規，利用此兩個感測裝置得到上半身目前在 Pitch 和 Roll 方向上個別的旋轉角度為多少，藉此得知目前地面環境的狀況為何，以此進行行走曲線上的補償修正。目前只用上半身的感測資訊來進行回授控制，因為我們認為由於雙足機器人在有一個較有重量的上半身的同時，上半身是否保持在可以維持平衡的穩定區間中，往往會大大的影響到機器人是否會跌倒，所以我們打算控制機器人的上半身，讓機器人在行走時不管遇到何種地形，都可以利用調整上半身的姿態，使得上半身重心投影能落在更加安全的腳部支撐多邊形區域內，達到在不同地形中皆能平衡行走的效果。

接著，對於雙足機器人行走在環境中的平衡分析上，我們認為當機器人處於在雙腳支撐相時，機器人整體的平衡穩定度是較高的，但當機器人在單腳支撐相時，整體的平衡會因抬腳擺動的關係產生改變，因此最需要注意此時是否能平穩完成前跨步伐的動作，所以我們需要在單腳支撐相開始前，就預測此單腳支撐相的跨步所會面臨的地形為何，在一開始就對地形的變化進行機器人的姿態對應，使得行走的下一步可以因而保持平穩的完成。

因此在平衡行走控制設計上，我們假設機器人在雙腳支撐相時，必為平穩的站在環境中，且假設機器人不會受到外力的干擾，例如：機器人軀幹被推動。在以上兩個假設的情況下，雙腳支撐相時期的重心移動由目前支撐腳移動到下一步支撐腳上時，此時對雙足機器人唯一的環境變化為地形上的改變，所以我們認為當雙足機器人行走時，即可利用此時所得到的感測資訊，得知下一步支撐腳的地形狀況為何，在下一步單腳支撐相開始時，立即完成在此次單腳支撐相中，需對理想行走曲線進行調整的補償控制，而此部分的補償控制設計，我們稱之為行走的預測補償設計，詳細內容會在 3.2 節中進行說明。

在前段當中我們已經提出了對於下一步單腳支撐相時的預測補償設計，但是此設計在補償控制中仍然存在不足之處，會導致機器人在單腳支撐相時發生跌倒，而此情況有以下兩點：

1. 在單腳支撐相時，此時機器人可能會發生自身擾動的狀態產生，導致對於行走曲線的執行上產生有些偏差，若當偏差過大時機器人的行走平衡穩定狀態就會被破壞，致使機器人跌倒。
2. 當機器人在雙腳支撐相已先對下一步的地形環境進行所需的控制補償了，但是在機器人開始單腳支撐相之後，卻因為上半身重心為了平衡搖擺移動的關係，讓本來與地面接觸的腳掌產生與重心有極大相關的轉向，因而導致機器人突然失去平衡導致機器人跌倒。

為了防止上述兩點導致跌倒的事情發生，在單腳支撐相時，除了利用在雙腳支撐相時所測得的補償控制值之外，我們再加入即時對機器人上半身的姿態感

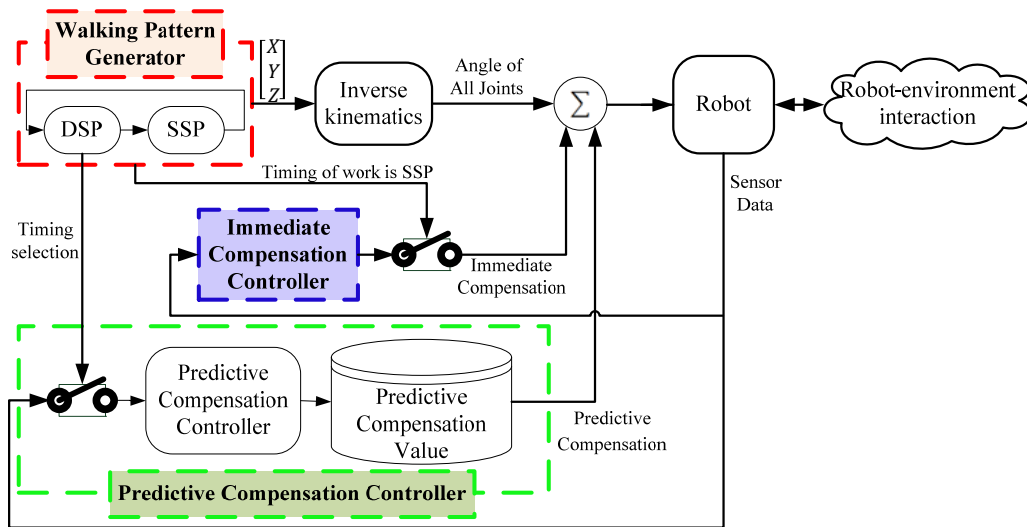


圖 3.1 整體之平衡行走控制架構圖

測，來得到機器人目前對地形環境的姿態，重新再判斷上半身所需補償的姿態角度，使得機器人可以即時的進行姿態調整，讓機器人可以持續地保持平衡行走，而此部分的補償控制設計，我們稱之為行走的即時補償設計，詳細內容會在 3.3 節中進行說明。

因此我們希望透過預測補償設計與即時補償設計的結合之下，讓機器人能因應地板斜面改變而產生即時姿態調整的行走控制，整體之平衡行走控制架構圖，如圖 3.1 所示。所以對於感測資訊應用的時機與兩個補償控制器各別的補償時機，我們將在 3.4 節中做詳細的說明。

3.2 行走的預測補償設計

我們認為在雙腳支撐相重心已移動到下一步支撐腳時，就必須已經能判斷出目前的地形，並且做出在下一步時需要的預測補償控制，才能即時地完成單腳支撐相時的行走曲線調整，防止機器人發生跌倒。因此，在預測補償設計中所需取得的補償值，我們是設計在機器人行走週期中的雙腳支撐相時期來取得此補償值，因此對於行走的預測補償設計，我們將在雙足行走處於雙腳支撐相時期來進行完整的補償取得設計，對於預測補償值的物理意義為修正上半身的姿態在 DSP 時，一直維持絕對於平地之直立，也就是希望 SSP 上半身的擺盪能與平地行走時相

同，此值的好壞判斷取決於是否與因應於地形變化所改變的理想補償值接近。圖 3.2 為雙足行走處於雙腳支撐相時的整體行走架構圖，預測補償值的取得設計也整合於其中。

對於圖 3.2 雙足行走處於雙腳支撐相時期之整體行走架構圖的方塊流程進行說明，首先，在左上方的紅色虛線大方塊為重心移動曲線規劃器，用來產生一連串機器人從左右腳到上半身所要到達目標點的相關座標。此相關座標傳入逆運動學方塊，得到所有關節馬達所需旋轉的角度。在右下方的綠色虛線大方塊為預測補償控制器，先利用感測器取得在 Pitch 方向目前機器人上半身的姿態角度為何當作感測資訊，並利用重心移動的時間當作選擇點，選擇是否進行預測補償控制器的工作，時間點選擇在雙腳支撐相要結束前，取用最後兩筆感測資訊來估測出

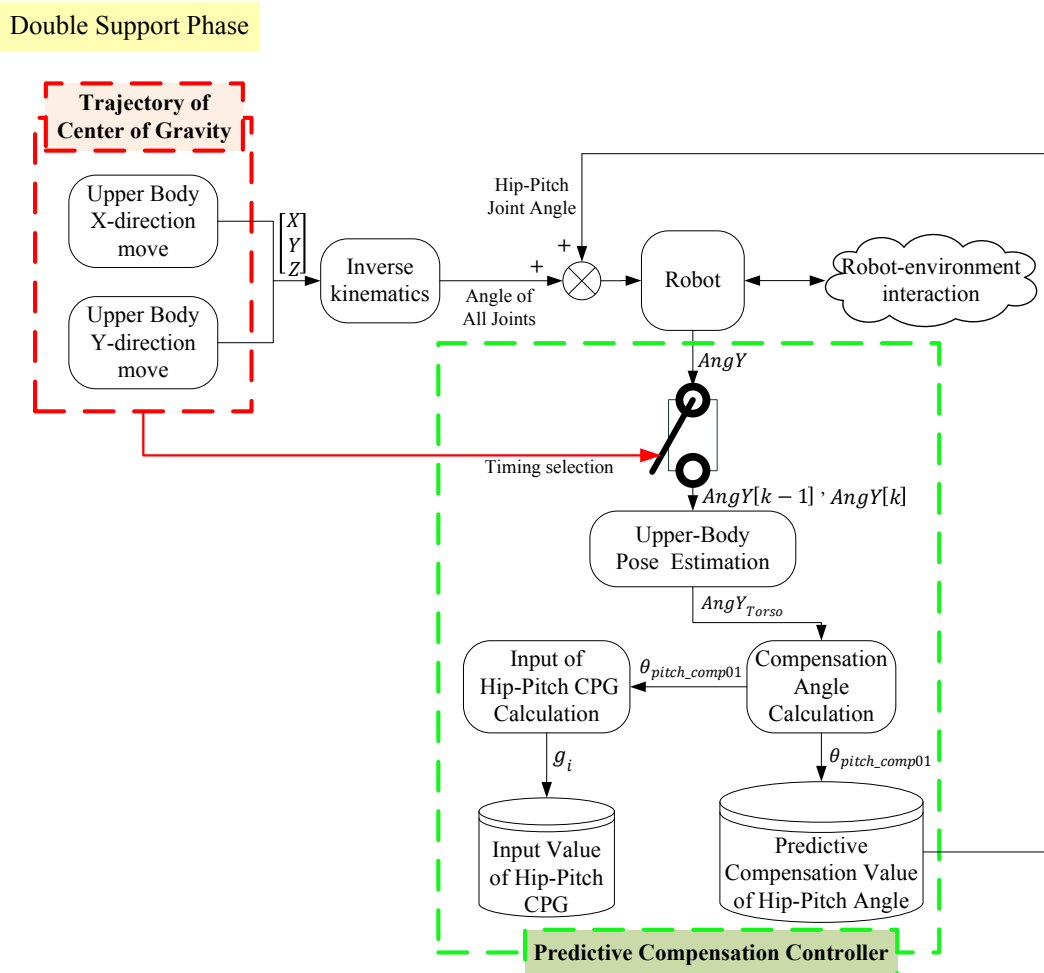


圖 3.2 雙足行走處於雙腳支撐相時期之整體行走架構圖

目前的上半身姿態，並運算出在下一步單腳支撐相時，機器人上半身姿態須調整多少角度來因應地形傾斜的變化的預測補償值，而此值除了會應用在下一步的單腳支撐相中，在下一步的雙腳支撐相中也會將需要補償到 Hip-Pitch 馬達上的角度值與從逆運動學方塊中的 Hip-Pitch 馬達的理想角度值進行相加，完成在下一步雙腳支撐相所需的上半身姿態修正。

在紅色虛線大方塊的重心移動曲線公式已在 2.3 節中進行說明，接著以下為在綠色虛線大方塊中的功能實現的公式說明。

對於上半身姿態的估測，首先取用最後兩筆 Pitch 方向的旋轉角度值進行平均，如公式(3.1)所示，再利用此兩筆 Pitch 方向的旋轉角度值計算出目前在 Pitch 方向的旋轉角速度，如公式(3.2)所示。

$$AngY_{DSP}[k] = (AngY[k - 1] + AngY[k])/2 \quad (3.1)$$

$$\omega_{AngY_{DSP}}[k] = (AngY[k] - AngY[k - 1])/(t[k] - t[k - 1]) \quad (3.2)$$

其中 $[k]$ 和 $[k - 1]$ 個別代表為雙腳支撐相時之最後一次和倒數第二次的輸出角度控制的次數， $AngY$ 為感測資訊在 y 方向所得的旋轉角度值， t 為程式之計時時間， $AngY_{DSP}$ 為利用兩時刻平均計算出較準之 y 方向旋轉角度值， $\omega_{AngY_{DSP}}$ 為計算出目前 y 方向旋轉角速度值。

若是在理想的感測估測和機器人移動下，其實根據我們的重心移動方式，兩筆被感測出的 Pitch 方向的旋轉角度值應為相同的，想當然爾此時的旋轉角速度應為零，但因為機器人本身移動即會產生自身的擾動和移動的慣性力影響，且感測器所得到的感測值也會有所偏差，因而導致感測出的資訊會有所誤差產生，感測角度不為相同，且角速度當然也不為零。

所以，為了減少誤差地取得最接近目前姿態的姿態角度值，我們在利用(3.1)式和(3.2)式所得到的上半身在 Pitch 方向目前姿態的角度和角速度時，其中個別乘上一個比例調整來減少誤差值的影響，藉此得到更加精準的目前姿態的姿態角度值，使得機器人目前的上半身能更精準的反映出地面斜度，如公式(3.3)所示。

$$AngY_{Torso} = k_{\theta} \times AngY_{DSP} + k_{\omega} \times \omega_{AngY_{DSP}} \times t_c \quad (3.3)$$

其中， k_θ 為比例常數、 k_ω 為速度常數，主要是以 trial and error 的方式取得， t_c 為程式在近兩個迴圈的平均執行時間， $AngY_{Torso}$ 為經過計算後較精準之目前上半身姿態角度值。

在所需要補償角度的計算方塊中，我們利用(3.3)式所得到的值，進行計算得到直接補償的角度，公式為(3.4)，而此值就用來當作在臀部 Pitch 方向的關節所需要的補償角度值，直接控制上半身的傾斜角度。

$$\theta_{pitch_comp01}(S+1) = -AngY_{Torso} + \theta_{HipPitch_Swing} + \theta_{pitch_comp01}(S) \quad (3.4)$$

其中， $AngY_{Torso}$ 為計算後之目前上半身姿態角度值，我們要進行補償因此須把值乘上負號， $\theta_{HipPitch_Swing}$ 為在公式(2.5)中之上半身理想前後擺盪角度， $\theta_{pitch_comp01}(S)$ 表示為此步中目前的預測補償角度值， $\theta_{pitch_comp01}(S+1)$ 表示為下一步開始將更新的預測補償角度值

對於臀部 Pitch 方向的 CPG 所需的輸入值的計算方塊，我們利用(3.3)式所得到的補償角度，進行等比例縮小的運算得到在下一步單腳支撐相在臀部 Pitch 方向的 CPG 所需的輸入值，如公式(3.5)所示。

$$g_i = k_{gi} \times \theta_{pitch_comp01} \quad (3.5)$$

其中 k_{gi} 為比例常數，針對參數的調整是以 CPG 的輸出不產生太大的 overshoot 的情況下，達成我們所需的上半身前後擺盪之調整效果。

3.3 行走的即時補償設計

在 3.1 節提及，當雙足行走處於單腳支撐相時，除了利用在雙腳支撐相時所測得的預測補償控制值之外，為了讓機器人能克服單腳支撐時可能會面臨的平衡問題，我們將再加入即時對機器人上半身的姿態感測，來得到機器人目前姿態，重新再判斷上半身所需補償的姿態角度，使得機器人可以即時的進行姿態調整，讓機器人可以持續地保持平衡行走，而此部分的補償控制設計，我們稱之為行走的即時補償設計，而對於雙足行走處於單腳支撐相時期之整體行走架構圖，如圖 3.3 所示，接著我們將對於圖中的每個大方塊進行說明。

首先，在圖 3.3 左上方的綠色虛線大方塊為預測補償控制器，我們將在雙腳支撐相計算出的兩個預測補償值在此進行單腳支撐相的回授，一個預測補償值當作控制臀部 Pitch 方向的 CPG 的新輸入，來改變 CPG 的輸出曲線；另一個在臀部 Pitch 方向的關節所需要的補償角度值，直接回授到關節角度整合方塊中，與其它臀部 Pitch 方向的角度進行相加。

接著，在圖 3.3 左中紅色虛線的為直接關節控制的大方塊是利用兩個 CPG 來產生輸出曲線再經過 CPG 計算成角度值來得到要讓機器人上半身搖擺的控制值，以上的運算公式如(2.5)、(2.6)所示，而在其中對於控制臀部 Pitch 方向的

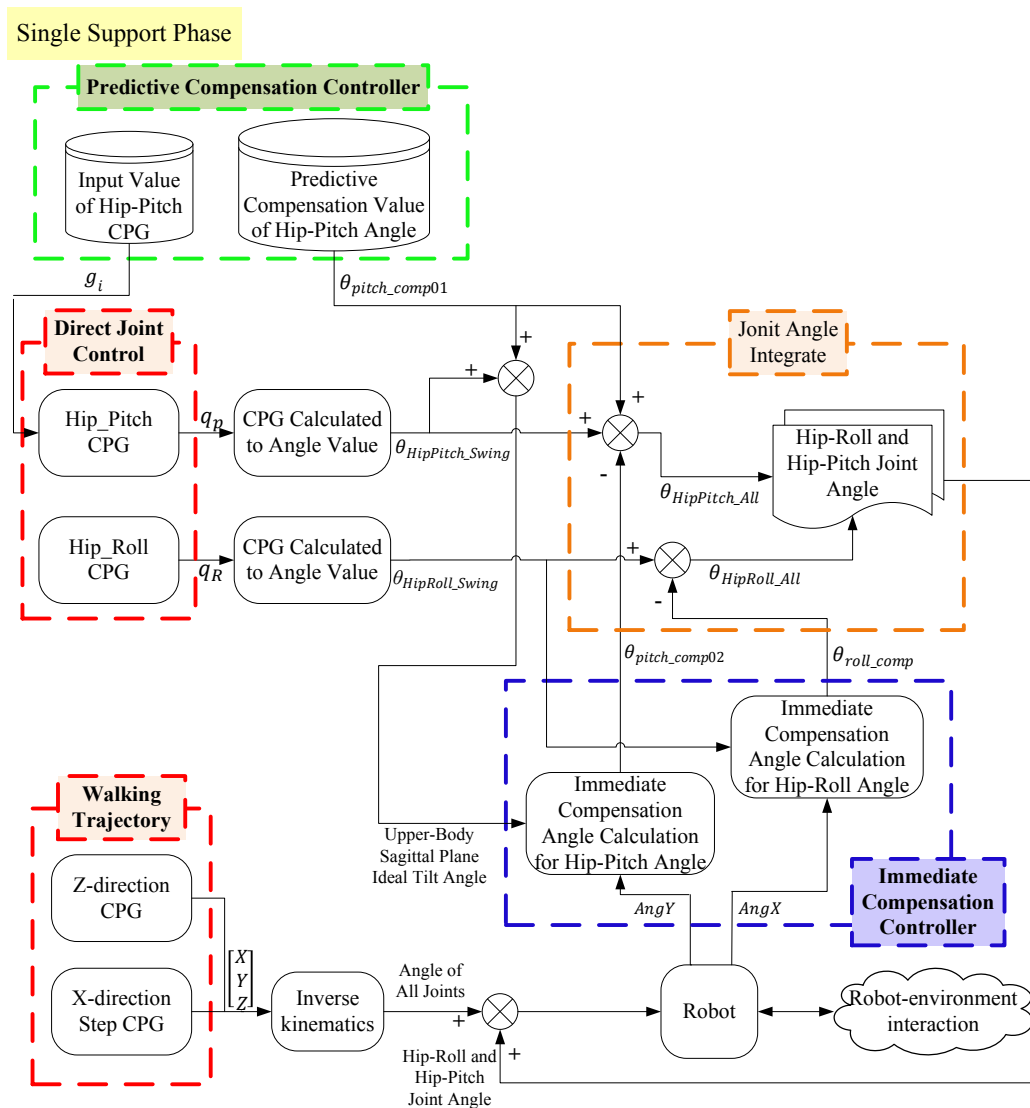


圖 3.3 雙足行走處於單腳支撐相時期之整體行走架構圖

CPG，我們利用了雙腳支撐相的預測補償控制器，得到新輸入進行 CPG 輸出曲線的調整，此目的為我們希望機器人上半身重心對於非在腳部支撐多邊形區域內的重心分力可以大大的減少，也就是要機器人走到上坡時，能增加往前的擺動且減少往後的擺動，相對走到下坡時，能減少往前的擺動且增加往後的擺動，使得機器人在進行擺盪腳往前擺盪時可以更加穩定。

然後，在圖 3.3 右上橘色虛線的關節角度整合大方塊中，主要就是要將所有透過感測資訊所得到的補償控制值，經過整理直接補償到臀部在 Pitch 方向的關節馬達角度和臀部在 Roll 方向的關節馬達角度，而在此大方塊中除了直接關節控制讓機器人上半身搖擺的控制值，和雙腳支撐相時所要補償的值要相加之外，當此一單腳支撐相在運行時，機器人持續會透過感測器得到機器人目前上半身在 Pitch 方向的角度和 Roll 方向的角度，再經由補償角度的計算回傳到關節角度控制的大方塊中，進行統一的整理補償。

那麼，在對於運算上半身在 Pitch 方向的角度補償角度計算方塊，所利用的公式為(3.6)式，此補償角度計算的目的為把機器人目前上半身姿態可以一直調整成與理想的姿態相同，這補償方塊位於即時的補償控制器圖 3.2 中右下方的藍色大區塊中。

$$\theta_{pitch_comp02} = k_{pcp02} \times (AngY - (\theta_{HipPitch_Swing} + \theta_{pitch_comp01})) \quad (3.6)$$

其中 θ_{pitch_comp02} 為計算出在 Hip-Pitch 關節上之即時補償角度值； k_{pcp02} 為比例常數為了減少補償誤差用來將即時補償值比例上減少，達成漸進補償； $AngY$ 為感測資訊在 y 方向所得的旋轉角度值， $\theta_{HipPitch_Swing}$ 為在公式(2.5)中之上半身理想前後擺盪角度， θ_{pitch_comp01} 為此步目前的預測補償角度值。

另外，對於運算上半身在 Roll 方向的角度補償角度計算方塊，所利用的公式為(3.7)式，此公式的設計目的為，由於當擺盪腳要抬起的一開始，在上半身左右擺盪的方向需要有一個較大的擺動慣性力，來達成成功的抬腳，因此設計的上半身左右擺動的曲線值，在腳成功抬起之後會有過度往支撐腳的方向傾斜情況導致跌倒，因此此公式設計就是希望當上半身目前姿態傾斜角度大於希望角度時

能及時修正，使得左右搖晃角度可以不要過大；而此公式的另一目的為因為我們沒有設計針對機器人發生旋轉時在行走上下坡時會發生面對在左右高度上有進行補償的預測補償控制器，因此此即時補償控制器也是希望當行走旋轉而導致左右高度上有所偏差時，機器人不會有傾斜過大而往左右兩邊倒的情況發生。此補償方塊也是位於即時的補償控制器圖 3.3 中右下方的藍色大區塊中。

$$\theta_{roll_comp} = k_{rcp} \times (-AngX - \theta_{HipRoll_Swing}) \quad \text{if } |\theta_{HipRoll_Swing}| < |AngX| \quad (3.7)$$

其中 θ_{roll_comp} 為計算出在 Hip-Roll 關節上之即時補償角度值； k_{rcp} 為比例常數為了減少補償誤差用來將即時補償值比例上減少，達成漸進補償； $-AngX$ 為感測資訊在 x 方向所得的旋轉角度值， $\theta_{HipRoll_Swing}$ 為在公式(2.6)中之上半身左右理想擺盪角度。

得到了上述在單腳支撐相時的即時補償角度之後，我們就可以得到準備在臀部在 Pitch 方向的關節馬達角度和臀部在 Roll 方向的關節馬達角度的所有補償角度，計算公式為(3.8)、(3.9)。

$$\theta_{HipPitch_All} = \theta_{HipPitch_Swing} + \theta_{pitch_comp01} - \theta_{pitch_comp02} \quad (3.8)$$

$$\theta_{HipRoll_All} = \theta_{HipRoll_Swing} - \theta_{roll_comp} \quad (3.9)$$

其中 $\theta_{HipPitch_All}$ 和 $\theta_{HipRoll_All}$ 皆為將非步態曲線透過逆運動學產生之關節角度的其他角度進行整合，整合的有要讓上半身擺盪的理想角度、預期補償控制值和即時補償控制值。

而整個單腳支撐相的基本跨步曲線為利用圖 3.3 中左下方紅色虛線理想走路曲線大方塊來實現，兩個 CPG 產生在 Z 方向(兩腳高度)與 X 方向(前進距離)兩方向的座標位置，以上對於座標的運算公式如(2.1)~(2.4)式所示，此時 Y 方向(左右)先假設為固定值，因而得到左右腳和軀幹的座標位置，接著透過逆運動學運算出各個關節馬達所要到達的理想角度，再加入行走時透過感測資訊計算後所得的所有補償角度，即是把關節角度整合大方塊得到的 Hip-Pitch 和 Hip-Roll 角度與透過逆運動學運算出的 Hip-Pitch 和 Hip-Roll 馬達的理想角度值進行相加，即

可把所有馬達要到達的角度資訊輸出給機器人，完成機器人的平衡行走控制。

3.4 補償控制的使用時機

兩個補償控制器作用的時機如圖 3.4 所示，紅色為在雙腳支撐相結束前，產生預測補償值的區域；綠色為在行走一整個週期中，預測補償值回授利用的部分；藍色為在單腳支撐相時，即時補償控制器運作的部分。

由於預測補償控制器的目的為要在雙腳支撐相重心從上一步支撐腳移到下一步支撐腳之後，即能在雙腳支撐相時偵測出目前機器人姿態與地面斜度的關係，藉此運算出預測補償值(紅色區域)，而得到了預測補償值之後，此補償值將回授應用到機器人的姿態平衡上，修正上半身在 DSP 時保持絕對於平地的直立姿態，因此在下一步的 DSP 和 SSP 皆用同一值，若只有 SSP 用，但 DSP 不用，會讓上半身因為此時的坡度狀況產生上半身恢復傾斜，增加跌倒風險，所以必須直至下一個預測補償值被偵測出才再進行更新已回應下一步的地形變化，因此在偵測出預測補償值之後，下一步的單腳與雙腳支撐相都將回授此預測補償值(綠色區域)，達成此一步中機器人姿態因應於地面斜度的關係達到平衡。

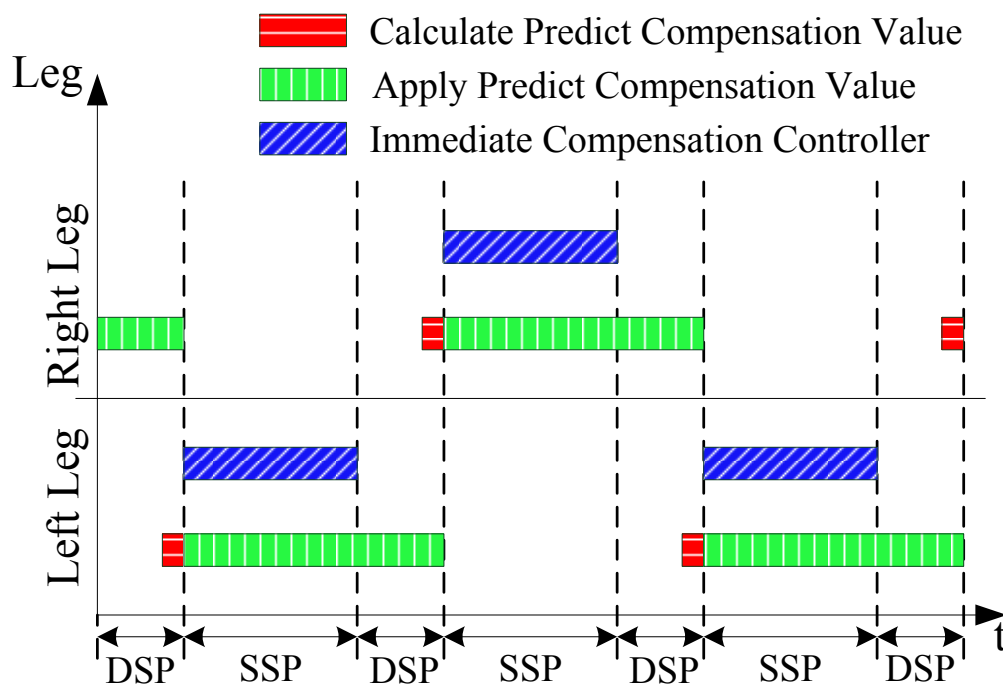


圖 3.4 控制器的使用時機示意圖

對於即時補償控制器的目的是在機器人進行擺盪腳的跨步時，機器人姿態可以持續保持在平穩的狀態，因此作用範圍主要是在整個單腳支撐相中，在此期間控制器會一直針對目前機器人的姿態與理想機器人的姿態進行比較，藉此來修正機器人的姿態，使得機器人的姿態可以符合預測補償值和單腳支撐相的步態規劃來移動，不會因為擺盪腳的抬起或是機器人本身的擾動，影響到機器人的平衡。而當單腳支撐相結束，雙腳支撐相開始之後，即時補償控制器的作用就逐漸歸零，因為我們假設當機器人兩隻腳著地且地形斜度還是在可以持續行走的範圍內時，機器人是可一直保持平衡站立的，所以這時不需要即時補償控制器的作用，只需要預測補償控制器的作用，即可應付雙腳支撐相時的上半身重心移動。

3.5 補償方式討論

在整體的補償控制中，我們著重於調整上半身的姿態使其達到平衡行走，因為我們認為對於上半身重量較重的機器人，調整上半身的姿態即可達到平衡控制。另外，由於我們單純利用裝於上半身的姿態感測器進行地形變化的感測，因此調整上半身姿態來達到行走平衡控制，為較直觀與感測資訊相對應的回授控制方法。

至於對於上半身的姿態控制，我們利用補償一髖關節角度值使得下一步在 DSP 和 SSP 時，初始姿態都可以先因應地形有變化產生，接著，再利用感測資訊當作 CPG 輸入調整 CPG 輸出曲線來達成在下一步的 SSP 上半身可以隨著斜坡變化進行擺盪的調整，在此沒有利用乘在 CPG 輸出上的參數來達到曲線調整，因為我們認為此參數為控制擺盪角度的比例因子，調整擺盪角度應該利用 CPG 輸出曲線的調整即可。

對於整體的補償控制，為何不是完全利用改變 CPG 曲線來進行控制調整，原因之一為在 1.2 節中所描述的 CPG 特性的限制，我們利用感測資訊當作輸入，進行調整 CPG 輸出時，必須避開 CPG 的限制“抑制作用”、“超越量的產生”才能使機器人繼續保持平衡，在此條件限制下，往往補償的角度會有些許的不足，

因此需要再另加直接補償的方式來達成補償目的。原因之二為在感測資訊作為 CPG 輸入利用 CPG 特性影響了 CPG 輸出曲線的變化，CPG 輸出曲線改變影響了行走模式改變之時，此時的行走模式可能無法使得行走趨於更加穩定。

總結的說，就是對於感測資訊直接的用來當作 CPG 輸入的回授應用上，我們認為相對難以促使機器人達到平穩控制，因此對於兩者之間的相互應用關係，我們尚未得到合宜的應用方式，因此在控制中對於無法直接利用 CPG 輸出曲線達到平衡的補償控制，我們皆先行利用加入額外的控制器來達成。

最後，對於其他補償控制方法的發想，我們認為如果單純改變步態長度與高度，要適應地形變化是較不可行的，因為當機器人在行走時需要有即時姿態調整的平衡控制才得以成功，若只靠改變步態長度和高度不能有效地應對發生傾斜的姿態，除非有利用視覺系統在踏下每一步之前，皆已計算出地形的改變為何，且步態高度的改變有加上對於傾斜面的姿態修正才得以成功。但若是在有了即時姿態調整的平衡控制之下，進行步態長度與高度的改變，確實可以增加機器人的行走穩定度與減少行走能量上的消耗，只不過所用的感測資訊就必須再增多以至於能準確的偵測地形狀態。

第四章 機器人實驗平台

本論文提出的雙足平衡行走控制架構將透過外購之雙足機器人 NAO 的硬體與軟體平台[29]進行實現與模擬和實驗的驗證。

4.1 NAO 之硬體介紹

NAO 機器人高 57.32 公分，重 4.7 公斤，足長 24.79 公分，圖 4.1 為 NAO 機器人的整體長度及每個桿件(link)之間的長度資訊。在雙足上共 11 組驅動器與馬達，左右腳各為六個自由度，在單腳的自由度分配上，以腳踝兩個、膝關節一個、臀部三個，如圖 4.2 所示，腳踝上的自由度分別為在 Roll 方向(紅)和 Pitch 方向(綠)，膝關節在 Pitch 方向(綠)，臀部三個自由度分別在 Roll 方向(紅)、Pitch 方向(綠)與 Yaw-Pitch 平面上 45 度的方向上(黃)的自由度 HipYawPitch，而此在臀部的自由度 HipYawPitch 為左右兩者是共用一個馬達進行控制；圖 4.2 除了雙足的自由度資訊介紹外，對於我們在 4.2 節順逆運動學要用到的每個自由度在 4.2 圖中也先加以命名(如 joint1)，因為我們在運動學中是由左腳底到肩部和右腳底到肩部，雙腳運動學計算屬於對稱操作，所以左右腳對應同位置自由度的命名為相同的。

表 4.1 為雙腳腳部相關馬達的旋轉角度限制，在此 LHipYawPitch 和 RHipYawPitch 我們不列為行走規劃所需的運動學的應用，因此在機器人行走時我們把 LHipYawPitch 和 RHipYawPitch 皆設定為 0 度不變，故不用考慮此兩自由度的角度限制。

表 4.1 雙腳各個馬達的旋轉角度之限制[29]

Joint name	Range(degrees)	Joint name	Range(degrees)
RHipRoll	-42.30 to 23.76	LHipRoll	-21.74 to 45.29
RHipPitch	-101.54 to 27.82	LHipPitch	-101.63 to 27.73
RKneePitch	-5.90 to 121.47	LKneePitch	-5.29 to 121.04
RAnklePitch	-67.97 to 53.40	LAnklePitch	-68.15 to 52.68
RAnkleRoll	-22.27 to 45.03	LAnkleRoll	-44.06 to 22.79

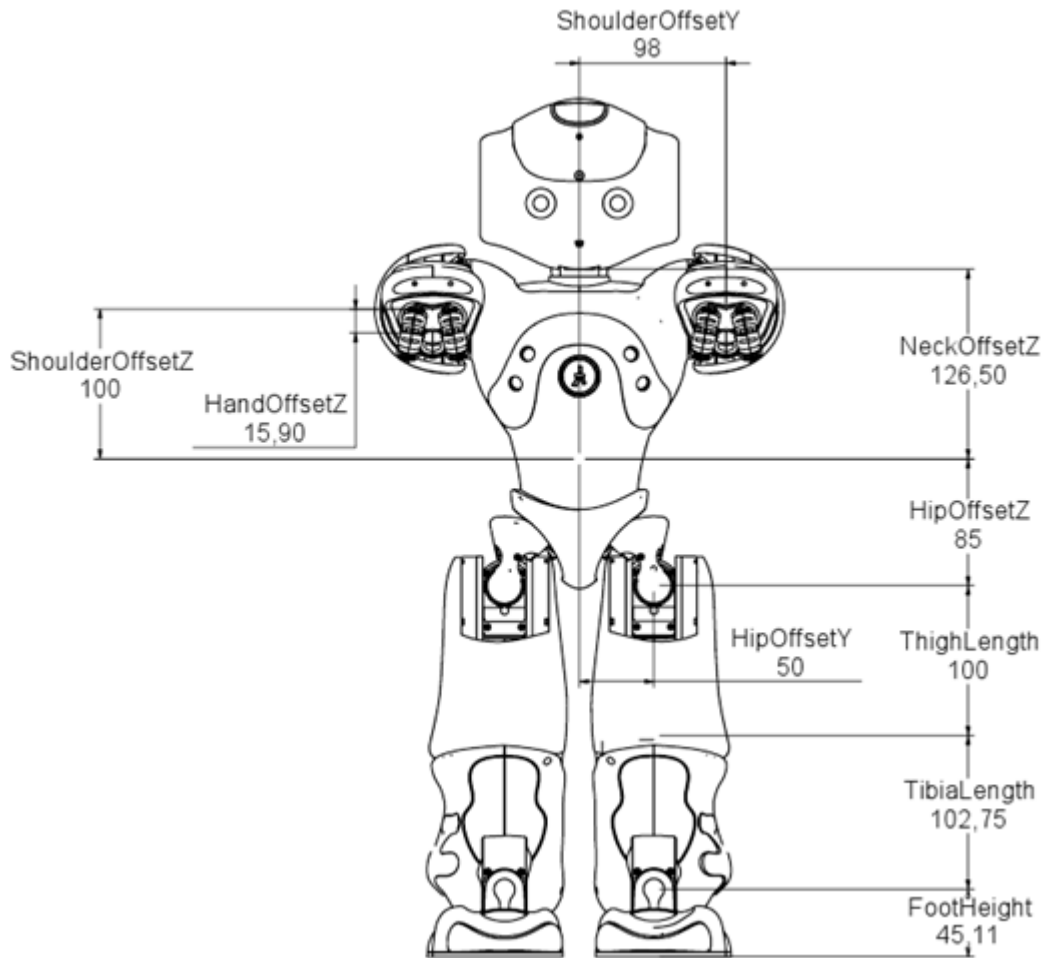


圖 4.1 機器人結構上桿件間長度說明(單位為公厘)[29]

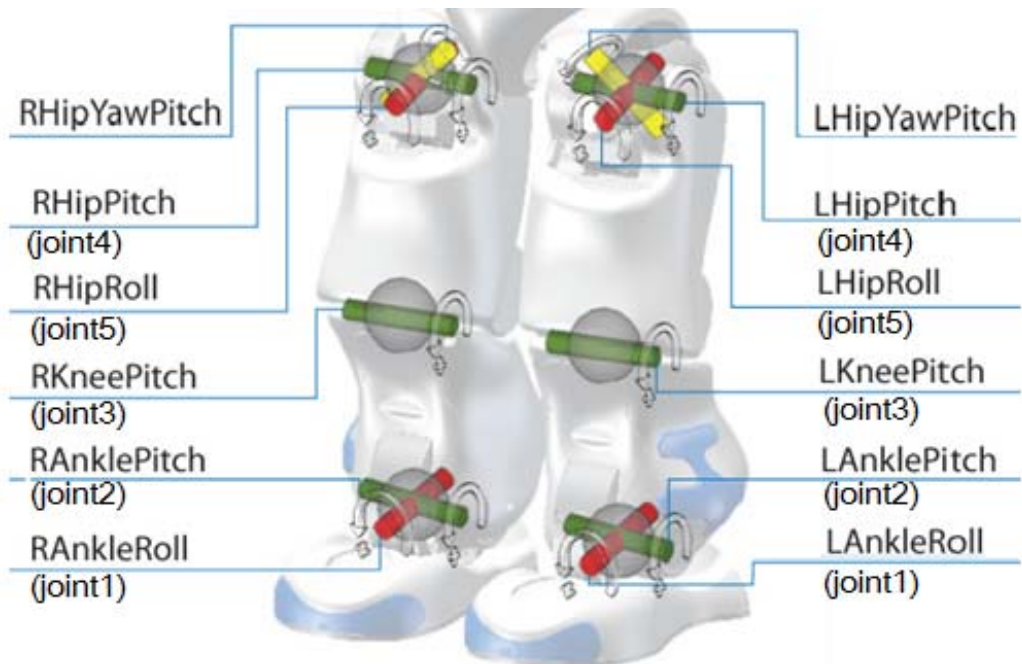


圖 4.2 雙腳自由度資訊與運動學使用上之命名[29]

4.2 NAO 之運動學計算

在第二章的雙足步行步態規劃中，我們提出了行走時的胸部軌跡以及雙腳腳部軌跡，我們將其相減算出應該產生之由腳部至胸部的末端點向量軌跡，此位置向量即是從雙腳腳底指向胸部末端點位置，因此我們將其帶入逆向運動學進行角度求解，在此由於左右腳到肩部的運動學為獨立運算，且沒有考慮到單一馬達控制兩自由度的關係，因此逆運動學計算出的各個自由度角度不會有衝突產生。

機器人運動學分為順(正)向運動學(Forward-Kinematics)及逆(反)向運動學(Inverse-Kinematics)兩部分。順向運動學是已給定各關節軸的轉角數據，進而求得末端桿件在絕對座標系中的空間位置；而逆向運動學則是已給定兩端在絕對座標系的位置，反過來求解各關節軸的旋轉角度值。

在本論文中，我們將採用 D-H model 來建立機器人之運動學模型，用以進行順逆向運動學的計算，建立的順序為從兩腳底到肩部，以往雙足機器人大多只考慮到腰部[22]，是因為之前用來研究雙足行走議題的雙足機器人大多無上半身，但是現今雙足機器人大多有上半身，且上半身反而成為機器人行走時要保持穩定的一大變因，因此我們在此把運動學模型考慮到的位置由腰部往上提高到肩部，也因為我們所用的姿態感測器裝在機器人的胸部位置，上半身軌跡曲線從肩部位位置開始計算，能使得感測資訊與上半身的軌跡曲線之間的關係更加的密切。

D-H model 是一種針對連桿運動設計的運動學分析法，可以對連桿在 3 度空間的移動與轉動有著完整的描述。進行分析之前，本論文先做下列假設：

- 各桿件為剛體。
- 桿件間旋轉接頭處視為一點。
- 擺盪中的腳底板的姿態會與初始狀態相同，亦即維持水平且方向不變。
- 不考慮雙足機器人受地面衝擊之振動行為。
- 假設雙足機器人支撐腳底部與地面不會產生滑動。

接下來我們利用 D-H model 來建立運動學模型，建立順序為由腳底到肩部，

如圖 4.7 所示，圖中為建立 D-H model 時，所設置在各自由度的三軸座標方向。

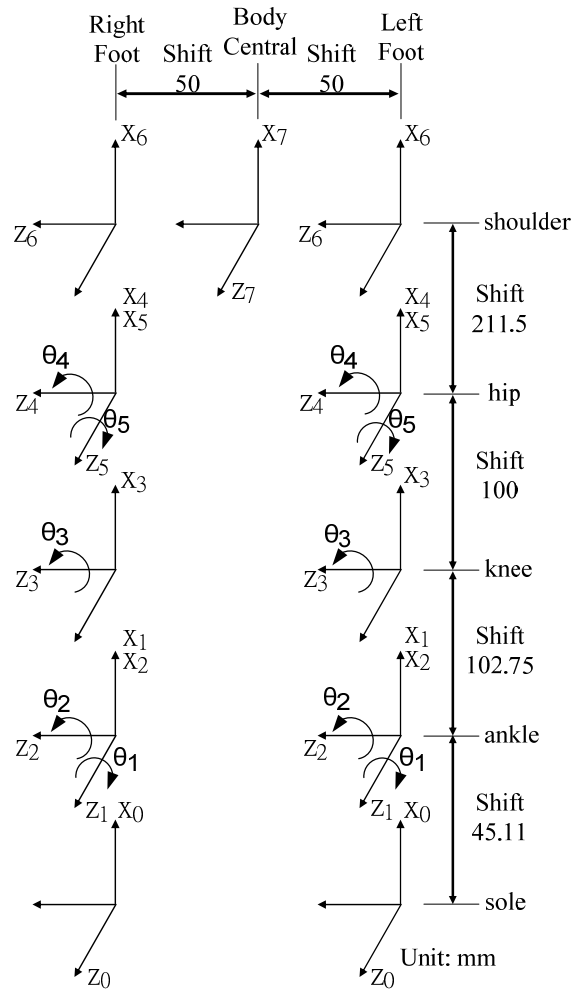


圖 4.3 NAO 雙腳之 D-H model

表 4.2 NAO之雙腳D-H model參數

	$\theta(^{\circ})$		d(mm)		a(mm)		$\alpha(^{\circ})$	
foot sole to joint1	0		0		45.11		0	
joint1 to joint2	θ_1		0		0		-90	
joint2 to joint3	θ_2		0		102.75		0	
joint3 to joint4	θ_3		0		100		0	
joint4 to joint5	θ_4		0		0		90	
joint5 to hip	θ_5		0		0		0	
hip to shoulder	0		0		211.5		-90	
shoulder to middle(L R)	0	0	-50	50	0	0	90	90

透過圖 4.3 所建立的 D-H model，我們可以得到機器人 D-H model 的參數，如表 4.2 所示，其中肩部左右兩邊到胸中央的距離相同為 50mm，但在參數中差一正負號，因此我們將肩部到胸中央此部分分成左右(L|R)腳來表示；接著因為每 4 個 D-H 參數(θ 、 d 、 α 、 a) 就可以決定出兩個座標之間的轉換矩陣，其利用的式子如式(4.1)，式子當中的左上角 3×3 矩陣代表的是旋轉方向，而第四行前三個元素分別代表為 X、Y、Z 方向的位移。因此當我們已得知所有轉換矩陣的 D-H 參數時，就可把所有座標之間的轉換矩陣依順序乘積起來，即可得知雙足機器人從雙腳底到手臂中央之間的相關位置；反之，當我們已知雙足機器人從雙腳底到手臂中央之間的相關位置時，也就可反推出每個自由度(θ_i)應到達的位置。

$$A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

接著，我們進行對雙足機器人的逆向運動學推導的前置作業，因為在表二中除了 joint1 到 hip 之間為真正有自由度會改變之外，其餘的座標轉換矩陣皆只為平移效果，因此對於求逆向運動學時，可以將其平移效果先行忽略，直到最後需要順向運動學計算肩膀與腳底位置時再將平移效果加入即可。所以在此我們先把我們需要在逆向運動學中使用的五個轉換矩陣建立出來，如式(4.2-4.6)。

$$A_1 = \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 102.75 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 102.75 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$A_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 100 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 100 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$$A_4 = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 & 0 \\ \sin \theta_4 & 0 & -\cos \theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$A_5 = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

在進行逆向運動學求解之前，對於我們在第二章設計的步態規劃的軌跡曲線，屬於世界坐標系，亦即機器人的前後座標為 X 軸、左右座標為 Y 軸、上下座標為 Z 軸，而在圖 4.3 雙腳的 D-H model 中，為了使得逆運動學好求解，我們將關節中的絕對座標系設為前後座標為 Z 軸、左右座標為 Y 軸、上下座標為 X 軸，因此當我們要將步態規劃的軌跡曲線利用到逆向運動學來得到此時各個關節所需到達的自由度時，需先將軌跡曲線的世界座標系轉為 D-H model 的絕對座標系，利用式(4.7)來達成。

$$\begin{bmatrix} absolute_x \\ absolute_y \\ absolute_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} world_x \\ world_y \\ world_z \end{bmatrix} \quad (4.7)$$

逆向運動學求解方式如下：

在此我們從腳底座標系往上進行連桿帶動，直到髖部的最後一個自由度，根據此模型可知此結構為 wrist-partitioned 的類型，所以位置屬於前三軸來得到，後兩軸處理旋轉，因此我們先利用轉移矩陣乘積中之第四行來得到前三軸自由度的解。首先，我們假設整體的轉移函數為 T，如式(4.8)，並比較其中 $A_2 * \dots * A_5$ 和 $A_1^{-1} * T$ 的第四行，如式(4.9-4.11)。

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

$$p_x \cos \theta_1 + p_y \sin \theta_1 = \frac{411}{4} \cos \theta_2 + 100 \cos \theta_2 \cos \theta_3 - 100 \sin \theta_2 \sin \theta_3 \quad (4.9)$$

$$-p_z = \frac{411}{4} \sin \theta_2 + 100 \cos \theta_2 \sin \theta_3 + 100 \cos \theta_3 \sin \theta_2 \quad (4.10)$$

$$p_y \cos \theta_1 - p_x \sin \theta_1 = 0 \quad (4.11)$$

我們首先先解出 θ_1 ，在此三個方程式中，利用式(4.11)經過運算後，即可得：

$$\frac{p_y}{p_x} = \tan \theta_1 \Rightarrow \theta_1 = \text{Atan2}(p_y, p_x) \quad (4.12)$$

另外，為相位相差 180 度的時候， θ_1 具有以下的同值解：

$$\theta_1 = \text{Atan2}(-p_y, -p_x) \quad (4.13)$$

在已知 θ_1 後，就可利用式(4.9)和式(4.10)求 θ_2 ，將兩式分別平方相加整理後可解

得下式:

$$\cos \theta_3 = \frac{(p_x \cos \theta_1 + p_y \sin \theta_1)^2 + p_z^2 - 100^2 - (\frac{411}{4})^2}{50 \cdot 411} \quad (4.14)$$

$$\sin \theta_3 = \pm \sqrt{1 - \cos^2 \theta_3} \quad (4.15)$$

所以可得

$$\theta_3 = \text{Atan2}(\sin \theta_3, \cos \theta_3) \quad (4.16)$$

由於在此 \sin 有正負號的緣故，另一解為

$$\theta_3 = \text{Atan2}(-\sin \theta_3, \cos \theta_3) \quad (4.17)$$

接著，因為 θ_1 、 θ_3 皆已知，我們再比較其中 $A_3 \dots A_5$ 和 $A_2^{-1} \dots A_1^{-1} \dots T$ 的第四行，

可得下列二式，如式(4.18-4.19):

$$(P_x \cos \theta_1 + P_y \sin \theta_1) \sin \theta_2 + P_z \cos \theta_2 = -100 \sin \theta_3 \quad (4.18)$$

$$(P_x \cos \theta_1 + P_y \sin \theta_1) \cos \theta_2 - P_z \sin \theta_2 = \frac{411}{4} + 100 \cos \theta_3 \quad (4.19)$$

藉由式(4.18)與式(4.19)解 $\sin \theta_2$ 與 $\cos \theta_2$ 之聯立方程式，可得

$$\cos \theta_2 = \frac{(100 \cos \theta_3 + \frac{411}{4})(P_x \cos \theta_1 + P_y \sin \theta_1) - 100 P_z \sin \theta_3}{P_z^2 + (P_x \cos \theta_1 + P_y \sin \theta_1)^2} \quad (4.20)$$

$$\sin \theta_2 = \frac{-100 \sin \theta_3 - P_z \cos \theta_2}{P_x \cos \theta_1 + P_y \sin \theta_1} \quad (4.21)$$

所以可知

$$\theta_2 = \text{Atan2}(\sin \theta_2, \cos \theta_2) \quad (4.22)$$

到此已解出前三軸角度，接下來就要利用末端點之空間姿態方向來解出後兩軸角度，因此此時對於後兩軸角度的運算，可以利用到 n 、 o 、 a 來進行運算，所以我們經由比較 $A_n \dots A_5$ 和 $A_{n-1}^{-1} \dots A_1^{-1} \dots T$ 兩個矩陣其中的每項，從中得到我們可以用來得到後兩軸角度的方程式，如式(4.23-4.26)。

$$\begin{aligned} \sin \theta_4 \cos \theta_5 &= \sin \theta_3 [n_z \sin \theta_2 - \cos \theta_2 (n_x \cos \theta_1 + n_y \sin \theta_1)] \\ &\quad - \cos \theta_3 [n_z \cos \theta_2 + \sin \theta_2 (n_x \cos \theta_1 + n_y \sin \theta_1)] \quad (4.23) \\ \cos \theta_4 \cos \theta_5 &= -\cos \theta_3 [n_z \sin \theta_2 - \cos \theta_2 (n_x \cos \theta_1 + n_y \sin \theta_1)] \end{aligned}$$

$$-\sin \theta_3 [n_z \cos \theta_2 + \sin \theta_2 (n_x \cos \theta_1 + n_y \sin \theta_1)] \quad (4.24)$$

$$\sin \theta_5 = n_y \cos \theta_1 - n_x \sin \theta_1 \quad (4.25)$$

$$\cos \theta_5 = o_y \cos \theta_1 - o_x \sin \theta_1 \quad (4.26)$$

因此可知，角度範圍在 $0 \sim \pi$ 時有一組解為：

$$\theta_4 = \text{Atan2}(\sin \theta_4, \cos \theta_4) \quad (4.27)$$

$$\theta_5 = \text{Atan2}(\sin \theta_5, \cos \theta_5) \quad (4.28)$$

角度範圍在 $-\pi \sim 0$ 時有另一組解為：

$$\theta_4 = \text{Atan2}(-\sin \theta_4, -\cos \theta_4) \quad (4.29)$$

$$\theta_5 = \text{Atan2}(-\sin \theta_5, -\cos \theta_5) \quad (4.30)$$

由上述的求解方式我們可從一組位置解出每個關節角度的所有解，而所有解的組合如圖 4.4 所示，共有八組，而此八組解需搭配表 4.1 中雙腳各個自由度可活動的範圍來進行篩選，最後只留下一組可執行解。到此，我們已經可以將想要之末端座標經由逆向運動學轉換成為各自由度之旋轉角度，達到機器人行走時的步態規劃和姿態控制。

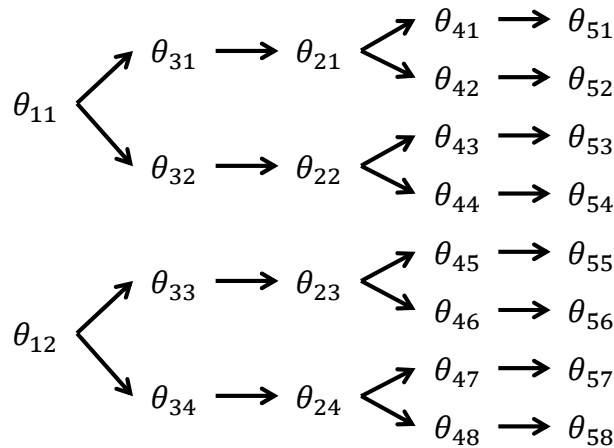


圖 4.4 一組位置可解出之所有角度解

其中角度值 θ_{ij} 的 i 表示第 i 個自由度， j 表示第 j 個解

4.3 NAO 之感測器

NAO 搭載多種感測器：

- 雙腳腳底安裝了八個腳底壓力感測計(FSRs)，此壓力傳感器利用測量電阻變化，運算出目前所施加的壓力，量測的力範圍為 0~25 牛頓力。
- 胸部內安裝一個兩軸陀螺儀(X 軸和 Y 軸)，偵測角速度為 $500^\circ/s$ ，精度為 5%；和一個三軸加速規，偵測加速度為 2G，精度為 1%。
- 胸前左右兩邊各有一組超音波感測系統，偵測距離為 0~70 公分，但在低於 15 公分時只能知道有物體擋住，無法偵測出準確的距離資訊。
- 在額頭和嘴巴上各有一個攝影機，分別拍攝前方和腳下，每秒可提供 30 張 640×480 解析度的影像。
- 頭的前後左右各有一個麥克風，左右耳朵各為一個擴音器。
- 在頭頂有三個觸摸感測器，以及兩腳掌前端各有一個碰觸感測器。

由於本論文提出之平衡行走系統設計，主要是改變上半身的姿態來達到行走平衡，因此，在此論文研究中只有用到裝在胸部內的兩軸陀螺儀和三軸加速規來進行平衡控制。對於以上兩種感測器，陀螺儀用來偵測角速度，其動態性能適於測量快速變化的訊息，但測量數據經積分處理後得到的傾斜角度會存在著累積誤差，所以長時間使用下，累積誤差將無限增大，導致控制系統無法正常工作；另一方面，當單獨使用加速規，讓機器人進行水平移動時，偵測出的角位置會發生旋轉，有受到相似低通濾波器特性的情況產生。

對於當利用單一感測器產生不足的地方，眾多研究文獻如[30-32]...等，提出利用陀螺儀和加速規的感測數據進行一起使用來得到傾斜角度，並將具有較強抑制干擾能力的卡曼濾波(Kalman filter)來對感測資訊進行前處理，使得可以得到精確的傾斜角度，因為卡曼濾波能使測量到的值近似於正確的值，不受隨機變化的雜訊和其他不準確性影響，使得控制方式具有良好的動態性能。

對於 NAO 上此兩個感測器的單獨取角度值依舊有此問題，所以我們一樣要利用兩個感測器的感測數據進行一起使用來得到傾斜角度，但在此我們將不用自行濾波，而是利用 NAO 官方以濾波過的軀幹在 X 方向和 Y 方向的傾斜角度值進行補償控制的應用，因為 NAO 官方 Aldebaran Robotics 公司已經利用一套算

法，透過陀螺儀和加速規的感測值，計算出軀幹的姿態角度。

NAO 官方表示此套算法利用了每個感測器的優良特性:當在靜態模式時，加速規是唯一的絕對參考，藉此得到準確的軀幹角度；當有運動被偵測到時，輸出角度計算將與陀螺儀一起運算，因為陀螺儀有好的動態性能。而對於上述的偏差值，此套算法也是透過加速規與陀螺儀的融合計算來達到消除此偏差值。

另外，為了驗證行走的步態規劃是符合靜態步行的設計，機器人的重心投影必須全落在腳部的支撐多邊形區域內，因此我們將再利用八個腳底壓力感測計 (FSRs)，測量機器人與地面之間的作用力，經由計算來得到 ZMP 軌跡。NAO 腳底壓力感測計配置位置如圖 4.5 所示，此圖中除了壓力感測器的相關位置之外，我們也標示了在行走時雙腳腳底在前後和左右方向的長度，來定義出 ZMP 的穩定區間。此感測訊號透過類比轉數位轉換後傳輸至控制系統中，並經過公式計算得到 ZMP，如式(4.31)，成為量測機器人平衡的環境資訊。在式(4.31)中， f_j 為第 j 個感測器所量到的壓力， s_j 為第 j 個感測器的位置向量， P 為 ZMP 的位置向量， n 為壓力感測器數量。因此，將機器人全部的 8 個感測器帶入所算出的 ZMP 為總 ZMP，而如果只將單一隻腳的 4 個感測器帶入計算則可得到對應此腳的 ZMP，而利用單腳的 ZMP 也可得知地面分別對左右腳的受力情形，而在單腳支撐相時，支撐腳之單腳 ZMP 即為總 ZMP，雙腳支撐相時，總 ZMP 必定會在左右腳單腳

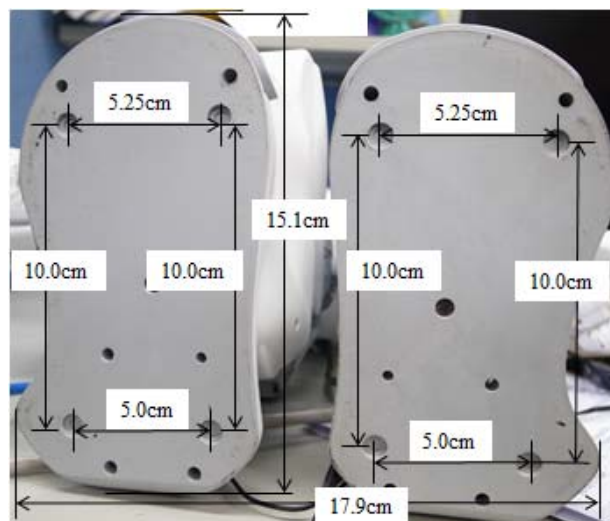


圖 4.5 NAO 足部壓力計分布圖

ZMP 之連線。透過這樣的計算方式，可以簡易且直覺的計算出雙足機器人於行走或是站立等其他姿態時的 ZMP 軌跡，以及兩隻腳分別的受力情況，由此可以了解到機器人的足部環境資訊。

$$P = \frac{\sum_{j=1}^n f_j \times s_j}{\sum_{j=1}^n f_j} \quad (4.31)$$

本機器人輸出的值為未經過處理的原始電壓輸出，所以我們先進行感測值的數據處理，使得它可以讓我們利用且得到目前機器人的狀態。腳底壓力感測器在完全無碰觸到地板時(即壓力為 0)，感測器得到的值皆不為 0，且 8 個腳底壓力感測器所得到的初始值都不一樣，所以我們先進行把取到的值做出統一歸零的動作，先測量懸空時壓力計所測量得值，再在感測器上放置不同重量之砝碼，如表 4.3 所示，由於每個壓力計所表現出的感測值和目前的重力是有成一定的比值，因此我們就可以使用公式(4.32)用來計算出目前機器人 ZMP 的位置，在式(4.32)中， f_j 為第 j 個感測器所量到的壓力， b_j 為感測器輸出值， b_{j0} 為感測器懸空時量測值， b_{js} 為校正用參數。

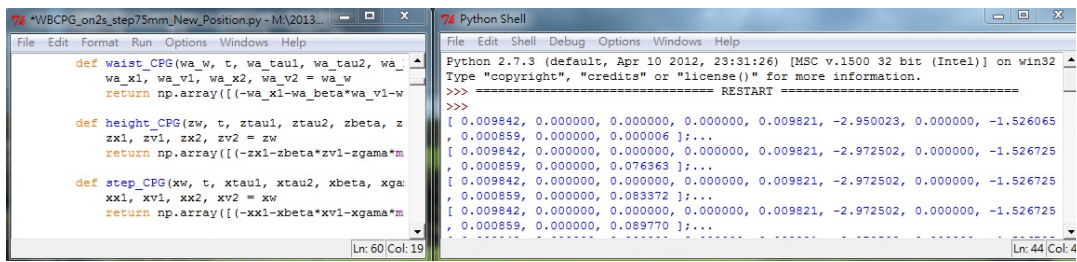
$$f_j = b_{j0} + b_j \times b_{js} \quad (4.32)$$

4.4 NAO 之軟體介紹

當我們使用者要控制 NAO 進行運動時，我們將使用 Python SDK 來達成程式的撰寫與應用，如圖 4.6 所示，圖 4.6(a)為程式撰寫介面，圖 4.6(b)為程式執行

表 4.3 壓力感測器量測值

Foot-Position	500g	200g	100g	0g
Right-RearRight	0.132	0.071	0.0478	0.03
Right-FrontRight	0.117	0.058	0.036	0.02
Right-RearLeft	0.095	0.038	0.02	0
Right-FrontLeft	0.071	0.0275	0.0073	0
Left-RearRight	0.14	0.057	0.0253	0.0068
Left-FrontRight	0.11	0.044	0.015	0
Left-RearLeft	0.394	0.23	0.17	0.12
Left-FrontLeft	0.145	0.06	0.03	0



(a) (b)

圖 4.6 Python SDK 之程式介面

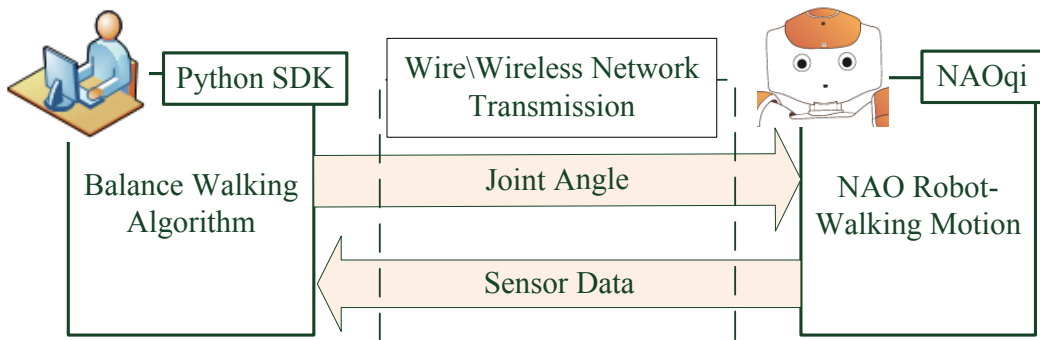


圖 4.7 NAO 之程式與硬體之間的傳輸方式

時將資訊回授顯示之介面。除了我們使用的方式之外，NAO 官方仍然提供了很多方法供使用者選擇，例如:NAO 官方自行開發的 AldebaranChoregraphe 圖形化軟體平台、C++ SDK、Matlab SDK...等。

對於電腦端 Python SDK 的程式要傳送馬達控制訊號給 NAO 機器人中使 NAO 運動並接收感測訊號時，我們將先透過設定，讓 NAO 自動得到目前的 IP 位置，我們再透過無線或是有線網路的方式，利用此 IP 位置連接到 NAO 機器人上的嵌入式軟體 NAOqi 中(NAOqi 為開發人員提供了一個全面的基礎，以提高、改進 NAO 的各項功能，通過此軟體，使用者將可選擇將模組在 NAO 上運行或是在電腦端遠程運行)，下達讓 NAO 各個關節運動的命令，並傳回感測資訊，再在電腦端此端進行平衡控制的運算，再把各個關節運動的命令傳到 NAOqi 中，控制機器人的運動產生，如圖 4.7 所示。

4.5 NAO 之電腦模擬系統

對於我們提出之平衡行走控制，我們希望在用到實體機器人之前可以先行利用電腦模擬的方式，測試控制系統的可行性，避免真實機器人因控制系統的錯誤而導致真實機器人硬體受到傷害，因此在我們進行真實的 NAO 測試之前，我們先利用 CyberBotics[33]所開發的 Webots for NAO 的模擬軟體，Webots for NAO 是 Webots7 的特定版本專門模擬 NAO，此模擬軟體提供了虛擬世界讓使用者可以使用控制器讓預定義模擬的 NAO 進行運動，測試控制器的可行性和成功性。

在此模擬器中我們可以先行選擇所需利用版本的虛擬 NAO，且加入所需的環境、地形或物品，如:上下坡、球、箱子...等。如圖 4.8 所示，左邊為環境中所有物件的資料，供我們從中改變物理性質(如:箱子的位置、大小、旋轉角度)，我們在此選擇了型號為 Nao_H21_V33 的虛擬 NAO，在圖中左邊上下兩小圖為 NAO 頭部兩個攝影機此時看出的環境狀態，由攝影機的環境得知我們在環境中透過箱子的旋轉加入了上下坡等地形。

對於控制虛擬 NAO 運動，我們一樣使用 Python SDK 來達成程式功能且輸出馬達所應到達的角度輸出。對於 Python SDK 的程式要傳送訊號給虛擬 NAO 中使 NAO 運動，用法與圖 4.7 我們連結到真實 NAO 的方式完全相同，而我們在

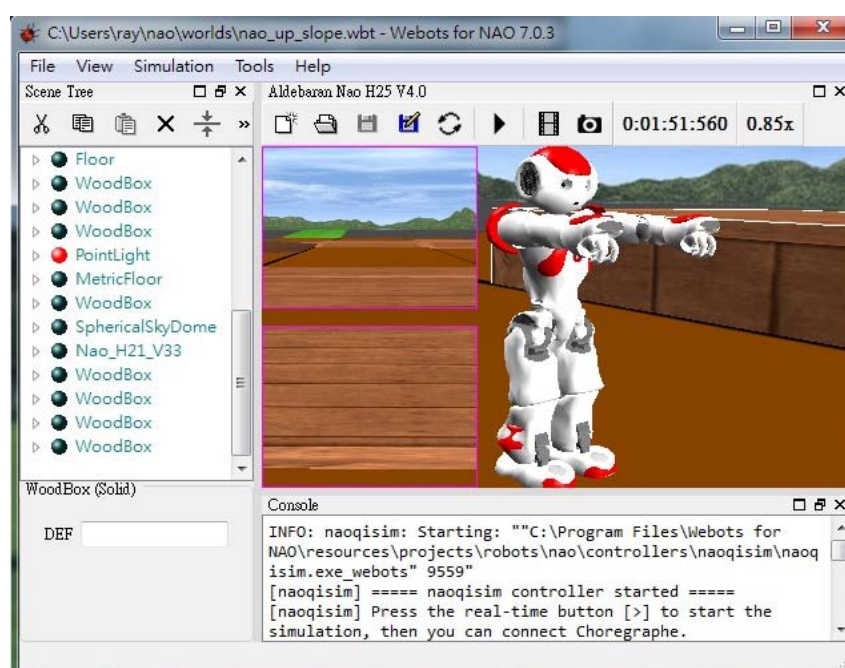


圖 4.8 虛擬 NAO 的模擬環境

此所需輸入的 IP 位置為目前使用的電腦的固定 IP 位置，達成連接到虛擬 NAO 機器人上的虛擬嵌入式軟體 NAOqi 中，如圖 4.8 右下角的文字顯示，此虛擬 NAO 在啟動時也有將虛擬的 NAOqi 控制器啟動，讓使用者進行使用。

在此模擬器中對於真實 NAO 有裝載的感測器，它皆有進行數值模擬，當我們點擊圖 4.8 中虛擬 NAO 的圖片位置時，會跑出各種感測器目前模擬出的狀態數值，如圖 4.9 所示，在上方可以切換不同感測器目前所提供的模擬狀態數值，如：加速規、攝影機、陀螺儀...等。而對於我們要用來應用的上半身姿態角度的感測值，在模擬環境中透過 NAOqi 的中對於陀螺儀和加速規的模擬數值的處理之後的上半身姿態角度值，與我們設定虛擬 NAO 目前上半身的姿態角度值是相似的誤差極小，於是此模擬感測器的模擬數值可以讓我們用來進行平衡行走控制的模擬驗證，且當我們一樣讓姿態為站立時，此模擬值與現實機器人 NAO 的感測值進行比較是符合且接近的，證明此模擬器對於感測器的數值模擬是符合真實情況的；另外對於腳底的壓力感測器，此模擬器中依然有進行模擬，但是在機器人行走時腳底對於地面的衝擊力影響，此模擬器中屬於理想的情況並非如真實世界相同會產生很大的衝擊力，此性質讓我們可以在不考慮衝擊力的影響下進行其他平衡效果的模擬與驗證，反之對於衝擊力的效果測試就會太過於理想化無法反映真實環境狀況。那麼由於我們要從行走步態規劃開始進行設計的驗證，需要先不考慮衝擊力的影響，因此我們將依然利用此模擬器來達到行走設計的驗證。

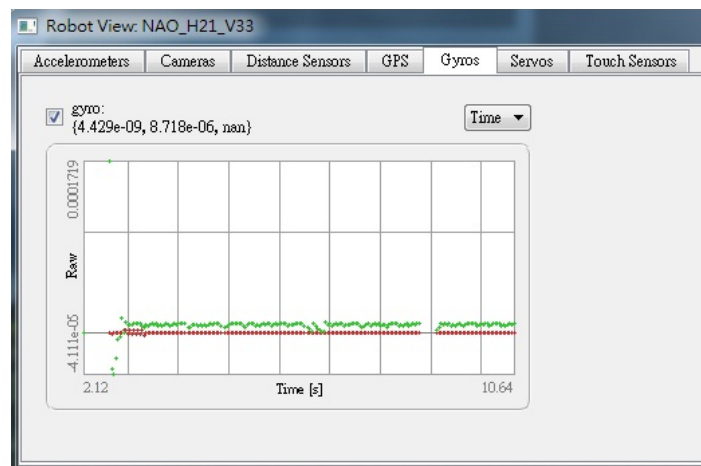


圖 4.9 虛擬 NAO 站立時之 Gyro 感測器數值模擬圖

第五章 模擬與實驗結果

本章主要的目的在驗證對於平衡行走系統設計的確可以使雙足機器人成功步行於不同斜度的地形之間，並驗證補償控制器的效果能針對不同的斜坡角度都能得到適當的補償控制修正。

首先，透過電腦模擬的方式進行驗證，模擬方式為利用 4.5 節所提到的模擬環境，此模擬的主要目的為在虛擬的環境、機器人和模擬的感測資訊下，進行對雙足平衡行走控制的驗證，而此模擬主要是希望在物理特性屬於理想且模擬的狀態下，先行驗證出我們提出的平衡行走控制是可成功的。最後，利用真實機器人在真實的環境中行走的實驗進行平衡行走設計的驗證，而此實驗的目的為希望在真實的物理特性與環境中，當有更多的不穩定因素之下(如:機器人本身的擾動)，我們的平衡行走控制設計是否仍然可以達成我們希望它達成的控制目標，藉此得證我們的控制理念與實現平衡行走的整體系統架構是可行的。

5.1 模擬和實驗之參數設定

由於模擬與真實環境中的地形狀態、機器人的物理特性、感測資訊和干擾...等性質皆存在著差異，因此對於每個公式的參數設定上，在模擬和實驗上會有所不同。此節就將針對在參數值設定上的同或異之意義，進行說明。

首先，對於雙足步行的步態規劃，在 2.2 節的單腳支撐相中，我們利用 CPG 來產生跨步時的行走曲線設計，如圖 2.2 所示，在圖中 4 個 CPG 振盪器所使用的公式皆為公式(1.1)~(1.6)，而對於這些公式的參數選擇，如表 5.1 所示。由於在此 CPG 主要是產生基本的連續振盪的曲線，使得我們可以進一步的設定跨步曲線的實際跨幅、跨高...等值，因此對於產生基本的連續振盪曲線的 CPG 參數，我們在模擬與實驗上皆用同一組參數值。

由於在 2.2 節中，我們安排了 4 個 CPG 來產生單腳支撐相時的行走行為，因此我們有 4 組 CPG 參數需要調整，但對於 4 組 CPG 參數中功能相同的參數選擇上有些將相同，但因為我們在 4 個 CPG 振盪曲線的設計上，仍然需要有不同

的曲線設計(在 2.2 節已有詳細說明 4 個振盪器的用途)，因此有些將選擇不一樣的值，所以對於相同之參數值的參數符號，將直接用 CPG 公式的符號表示，不另外標註所用的振盪器符號，而對於不同的參數值將在公式符號之右下角的標註各為： z 為控制跨步高度、 x 為控制跨步長度、 hr 為控制 Hip-Roll 的擺盪角度、 hp 為控制 Hip-Pitch 的擺盪角度，如表 5.1 所示，而同一功能之不同的參數值設定，我們利用雙直槓 $||$ 來將參數符號和所對應的參數值分開標示。我們利用的 CPG 輸出曲線的波形與[18]相同，因此大部分的參數選擇，我們是參考[18]的參數選擇，並再針對我們所要的功能進行週期和初始值上的參數改變。

透過表 5.1 的參數設定，因為時間常數(τ_1)的不同，控制跨步長度、跨步高度和 Hip-Roll 擺盪角度的 CPG 輸出週期為 Hip-Pitch 擺盪角度的 CPG 輸出週期的 0.5 倍，藉此才能達成圖 2.3 我們想要的擺盪控制-踏一步時，跨步曲線與左右擺盪只完成半個週期，而此時前後擺盪已完成一整個週期:接著，因為初始的內部變化數($u_1(0)$ 、 $u_2(0)$ 、 $v_1(0)$ 、 $v_2(0)$)不同，所以控制跨步高度、Hip-Roll 擺盪角度和 Hip-Pitch 擺盪角度的 CPG 輸出曲線為 $0 \rightarrow 1 \rightarrow 0 \rightarrow -1 \rightarrow 0$ ，而跨步長度的 CPG 輸出曲線為 $1 \rightarrow 0 \rightarrow -1 \rightarrow 0 \rightarrow 1$ ，會有如此區別是因為我們設定跨腳之前上半身為理想直立且雙腳碰地，所以振盪曲線應由 0 出發，而跨步的長度因為

表 5.1 CPG振盪器之參數，模擬與實驗皆用此組參數

Description	Symbol		Value	
time constant	τ_{z1} 、 τ_{x1} 、 τ_{hr1}	τ_{hp1}	1/2.918	1/5.848
time constant ratio	τ_1/τ_2		0.8	
mutual inhibition	γ		2.0	
adaptation constant	β		2.5	
tonic excitation	c		2.43	
scale gain of input	h		1	
initial internal variable	$u_{z1}(0)$ 、 $u_{hr1}(0)$ 、 $u_{hp1}(0)$	$u_{x1}(0)$	0.4432	0.9924
initial internal variable	$v_{z1}(0)$ 、 $v_{hr1}(0)$ 、 $v_{hp1}(0)$	$v_{x1}(0)$	0.1885	0.5834
initial internal variable	$u_{z2}(0)$ 、 $u_{hr2}(0)$ 、 $u_{hp2}(0)$	$u_{x2}(0)$	0.4334	-0.175
initial internal variable	$v_{z2}(0)$ 、 $v_{hr2}(0)$ 、 $v_{hp2}(0)$	$v_{x2}(0)$	0.69	0.3229

後腳跨到前腳初始位置上本來就有差距值，因此從 1 出發當到達 0 時也就是與支撐腳 x 方向齊平。

接著，對於步態規劃，2.2 節的單腳支撐相中我們已得知四個振盪器的參數值，因此我們再透過公式(2.1)~(2.6)，得到完整的跨步曲線，再加上 2.3 節雙腳支撐相中的公式(2.7)和(2.8)，得到上半身在雙腳支撐相時的重心移動曲線，因而得到一個完整的步態規劃，在模擬與實驗中對於公式(2.1)~(2.8)公式中的參數設定如表 5.2 所示，調整的方式為讓機器人重心維持在支撐腳的區間中。

表 5.2 內的參數設定會影響到跨步曲線的跨幅、跨高...等值和重心移動時的穩定度，因此在模擬和實驗中將用不同組參數，調整出符合各自環境的參數值。

將表 5.1 和表 5.2 中模擬用的參數值代入各個的公式，在模擬中所使用的完整步態曲線為：每一步的單腳支撐相為 1 秒、雙腳支撐相為 0.5 秒，因此左右腳各走完一步之完整週期為 3 秒，而此步的跨步長度(x 方向)為 15 公分、跨步高度(z 方向)為 2.5 公分，左與右腳 y 方向的中心位置各別固定在-5 公分與 5 公分的位置上，前後擺盪振幅為 8.5°，左右擺盪振幅為 19.5°，在雙腳支撐相時上半身重心移動到左右腳上各為-5 和 5 公分，與理想雙腳中心位置相同。而在實驗中所使用的完整步態曲線為：每一步的單腳支撐相依然為 1 秒，另外在雙腳支撐相時間

表 5.2 步態規劃之參數

Symbol	Simulation-Value	Experimental-Value	
		Non Feedback	Feedback
$z_0(\text{mm})$	415.0	415.0	
$A_z(\text{mm})$	12.5	7.5	
$x_0(\text{mm})$	75.0	50.0	
$A_x(\text{mm})$	75.0	50.0	
$k_{hip_pitch}(^\circ)$	8.5	2.0	
$k_{hip_roll}(^\circ)$	19.5	9.5	
$T_{Two_Support}(\text{s})$	0.5	0.5	1.5
$k_{Sx}(\text{mm})$	99.272	66.18325	52.6197848
k_T	0.355	0.355	0.5

上，當實驗為單純驗證步態規劃的可行性之下時間仍然為 1 秒，因此左右腳各走完一步之完整週期為 3 秒，但當實驗為要行走在上下坡進行補償控制時，為了減少腳觸地時產生衝擊力...等干擾因素，增加感測資訊的精準度，因此雙腳支撐相的時間增為 1.5 秒，因此左右腳各走完一步之完整週期為 5 秒，而此步的跨步長度(x 方向)縮短為 10 公分、跨步高度(z 方向)也縮短為 1.5 公分，左與右腳 y 方向的中心位置各別固定在 -5 公分與 5 公分的位置上，前後擺盪振幅為 2° ，左右擺盪振幅為 9.5° ，而在雙腳支撐相時上半身重心移動到左右腳上各為 -5 和 6 公分，為了平衡與理想雙腳中心位置相比右腳的重心有再向右偏。若要調整行走的速度，SSP 為調整 CPG 的自然頻率，DSP 為調整上半身重心移動的位置控制速度。

對於模擬和實驗上參數值設定的差別原因為，在實驗中增加雙腳支撐相的時間是為了增加此時刻的穩定度和削減降低腳落地時產生的振動，因此我們希望利用增加最短的時間將此雜訊對姿態感測的影響降到最低；另外，因為當抬腳的高度和長度設定較長時，會需要較大的扭矩來達成向前跨步，但這樣就會使得機器人不能平穩的持續行走，所以在實驗時對於跨步的長度和高度的大小都減少，藉此增加單純雙足步行時的穩定度，而這些值皆是根據機器人的平衡狀態進行調整，且行走曲線的設計仍然有將機器人的重心投影落在腳步的支撐多邊形區域內；最後對於在實驗時雙腳支撐相時上半身重心移動與理想雙腳中心位置有所不同，我們認為是硬體設計上的重心並非理想的左右對稱，因此當左腳抬起時，上半身所在的支撐腳右腳上須再往右移動一些距離，才能使得機器人保持平衡。

我們到目前為止已得到了模擬時所將使用完整步態曲線，因此接下來我們將再加入感測資訊的應用來達到平衡行走的控制，因此我們利用公式(3.1)~(3.9)來達到平衡行走時的感測資訊回授控制，對於在模擬和實驗時公式中的參數設定如表 5.3 所示。

由於表 5.3 內的參數設定影響到感測資訊，因此在此模擬和實驗的參數將用不同組參數，調整出符合各自環境的參數值。

在表 5.3 中，所有的參數皆是透過 trial and error 得出，對於預測補償控制器

表 5.3 平衡控制器之參數

Symbol	Simulation-Value	Experimental-Value
k_{θ}	1.0	0.24
k_w	0.0	0.3456
k_{gj}	0.04	0.04
k_{pcp02}	0.6	0.35
k_{rcp}	0.8	0.3

的參數值調整(k_{θ} , k_w), 目的為讓上半身的姿態能持續的保持與行走在平地時相同, 即為在 DSP 時上半身姿態與平地保持直立, 在模擬環境中, 機器人處於雙腳支撐相時, 模擬的感測資訊屬於穩定的值, 因此單靠感測資訊所得到的位置變化進行地形變化的判斷即可, 而在真實環境中, 感測資訊屬於跳動較不穩定的值, 所以需靠感測資訊得到的位置和速度變化進行地形變化的偵測; 另外對於控制 CPG 輸出曲線的輸入曲線參數值(k_{gj})的得到, 我們考慮了感測資訊的值大小幅度和不要讓輸出曲線有過大的 over-shoot 產生的情況進行參數的調整, 由於模擬和實驗中的感測資訊的大小範圍相同, 因此模擬和實驗皆用同值; 對於即時補償控制器的參數調整(k_{pcp02} , k_{rcp}), 因為感測資訊無論在模擬或實驗的單腳支撐相, 感測資訊跳動幅度都相對於雙腳支撐相時大, 因此對於感測所得到的資訊都進行比例上的縮小, 藉此得到較穩定的變化值, 使我們進行修正時不會有大幅度的動作修正, 而是以漸進的方式來進行平衡修正。

有了步態規劃和平衡行走控制設計在模擬與真實實驗中的參數值, 因此我們可以開始在模擬與真實環境中進行對於我們提出之雙足平衡行走控制的驗證。

5.2 平衡行走之模擬結果

在模擬中我們將讓機器人各別在 4° 、 5° 和 7° 的斜坡環境中, 進行“平地→上坡→平地→下坡→平地”此任務的行走測試, 目的為驗證我們設計的控制方法面對不同的斜坡角度的轉換都能得到適當的補償控制修正。

5.2.1 平衡行走於 4° 斜坡之模擬結果

平衡行走於 4° 斜坡之模擬過程如圖 5.1(a)~(o)所示，此時對於地形偵測的情況與補償控制器的作用效果說明如下：

行走時偵測到在 Y 方向的旋轉角度 $AngY$ ，如圖 5.2 所示，此感測值 $AngY$ 隨上半身的前後擺盪呈現擺盪曲線，從整體變化來看，在上坡時上半身有稍微的往前傾；另外，在下坡時上半身的傾斜角度為向後傾斜，此感測值表示補償控制器的作用使得上半身重心的移動行為，使得行走能持續保持平穩。

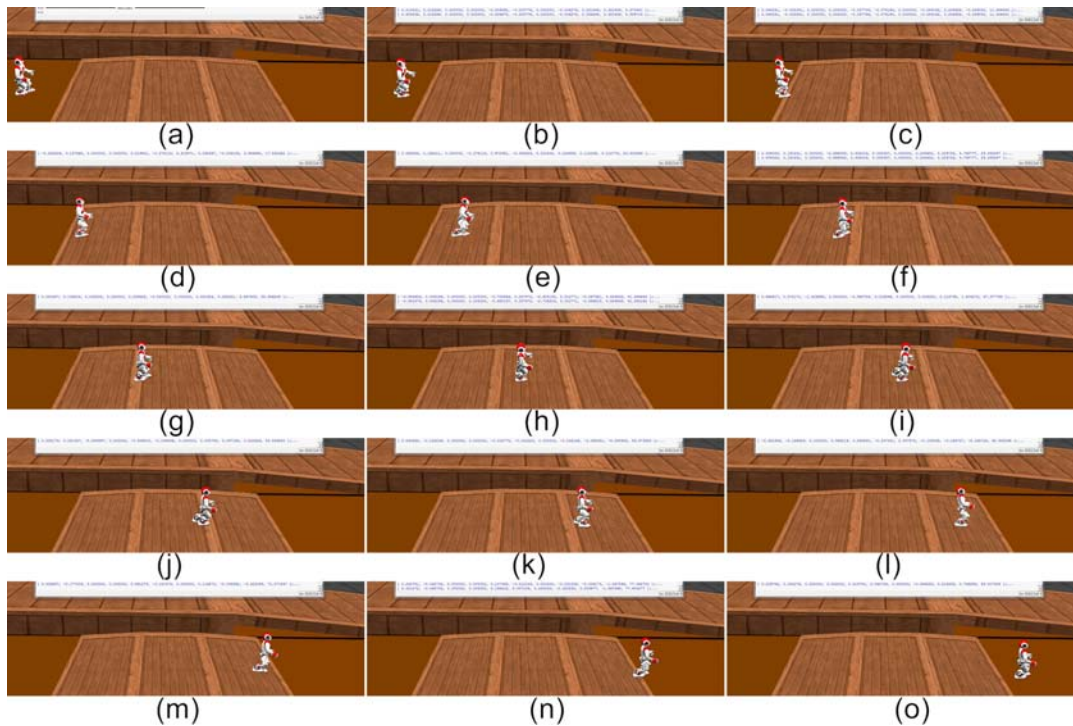


圖 5.1 平衡行走於 4° 斜坡之模擬過程

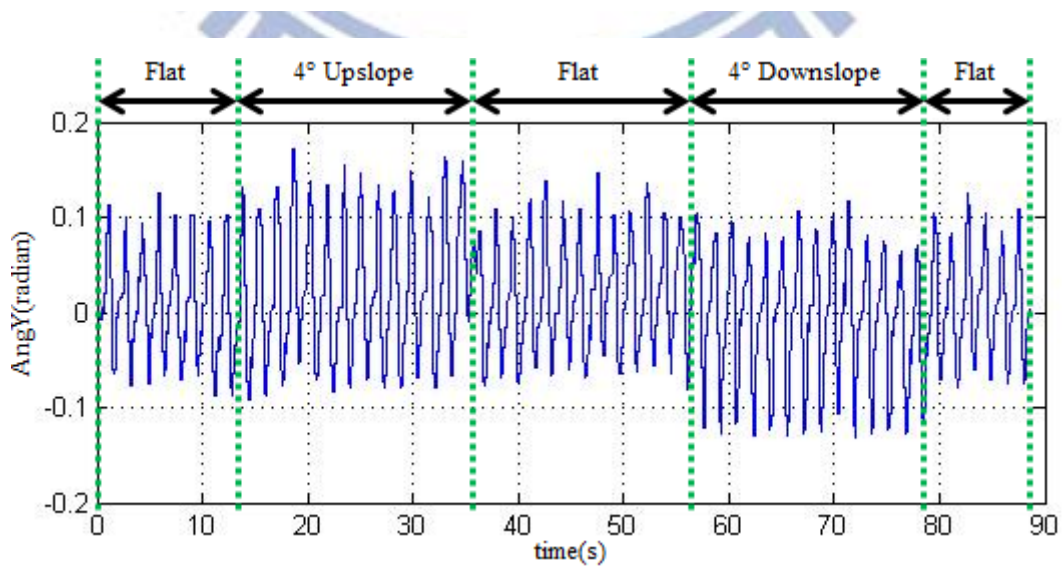


圖 5.2 行走過程中之 Y 方向旋轉角度曲線

如圖 5.3 顯示，預測補償控制器對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，從圖中可以明顯的看出當機器人面臨到地形的變化時都能適時的進行上半身的修正，使得上半身能相對於水平地面持續保持直立，在平地的預測補償值都在理想值 0° 附近；在上坡的預測補償值都在理想值 4° 附近；在輸出曲線受到輸入曲線改變的影響，使得上半身在走到上坡有多前傾(正值變大)少後傾(負值變小)，且在下坡時多後傾少前傾的狀況發生，如圖 5.4 所示。

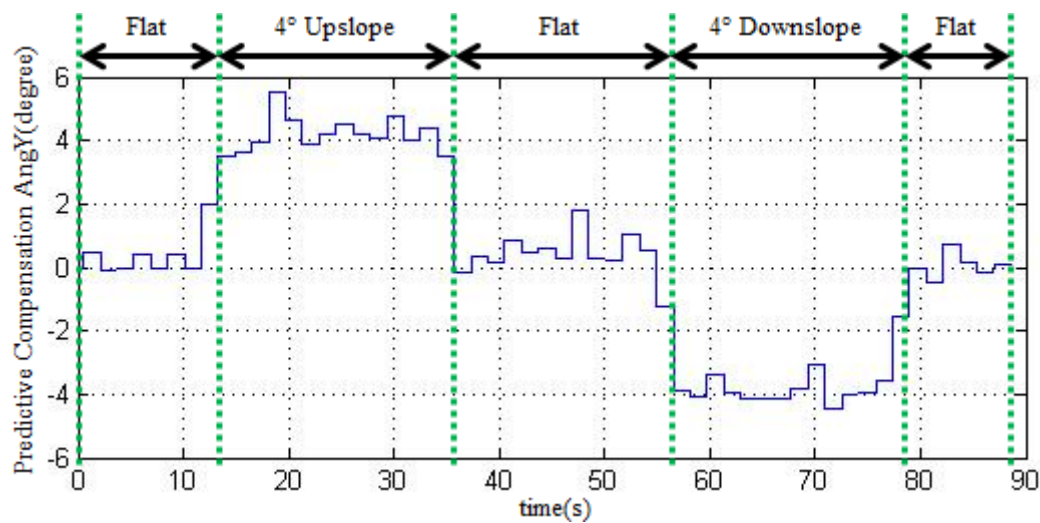


圖 5.3 直接補償於關節 Hip-Pitch 之預測補償角度值

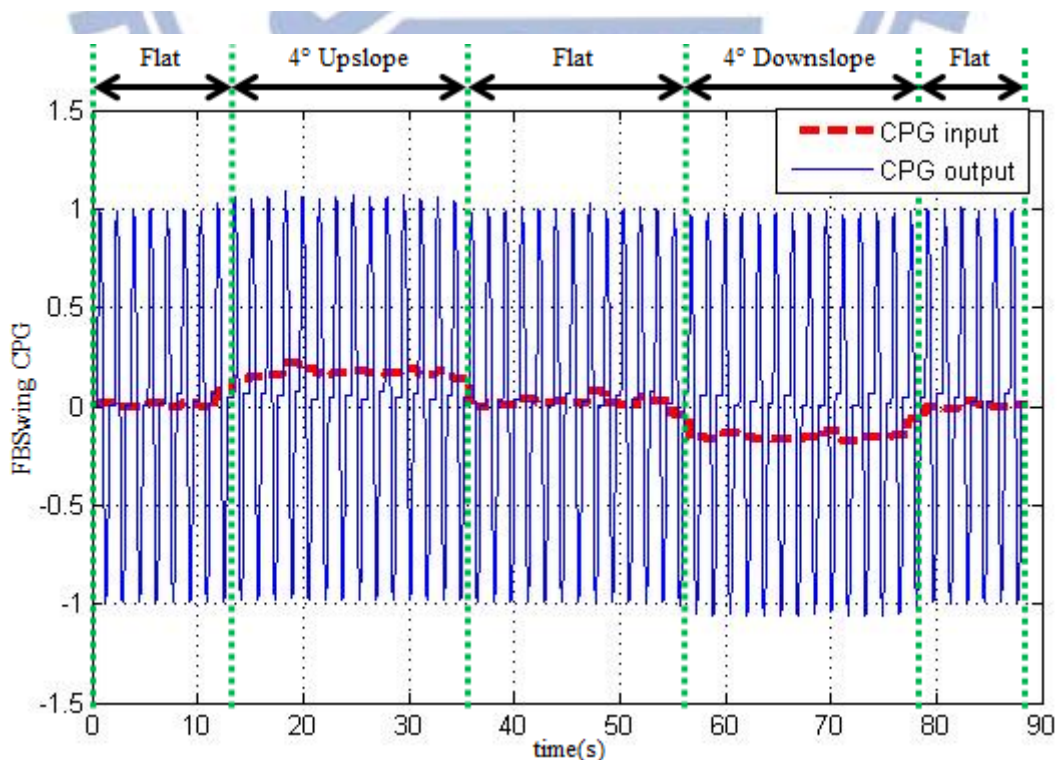


圖 5.4 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

如圖 5.5 顯示，在即時補償控制器中，對於補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，為了克服雙腳支撐轉換成單腳支撐時的不平衡，而產生行走曲線的偏移，在圖中可以看出當走在平地時前後補償值平均；在走到上坡時，補償值會先將上半身重心往前(負值)旋轉傾斜，克服抬腳時的重心往後傾，而為了使得曲線符合在落腳時上半身往後擺盪的曲線，所以每步的最後再把重心往後(正值)旋轉傾斜；而走到下坡時為了使得上半身不會因為抬腳時的慣性衝力和重心對於地形的分力影響往前傾，補償值會將上半身重心往後(正值)旋轉傾斜，達到遵守行走曲線的行走。

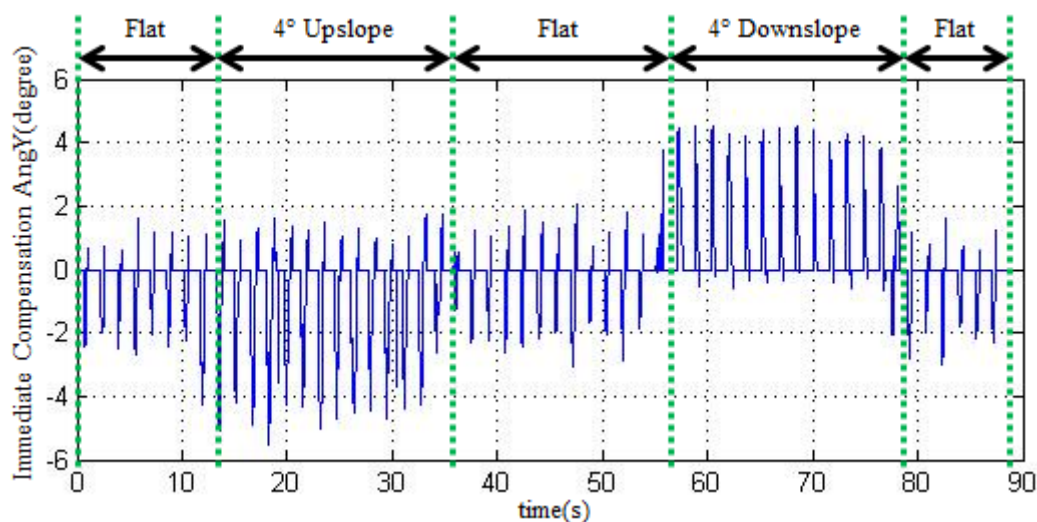


圖 5.5 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

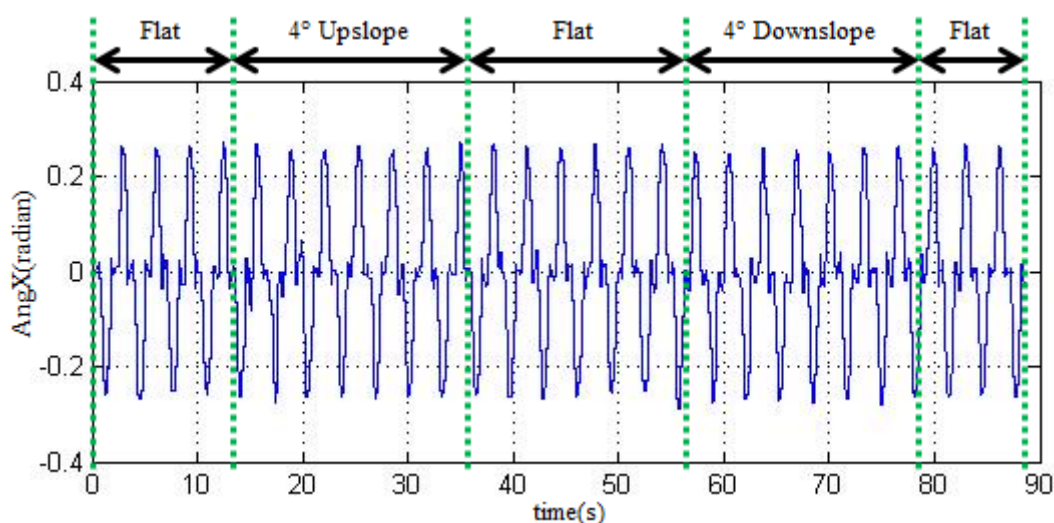


圖 5.6 行走過程中之 X 方向旋轉角度曲線

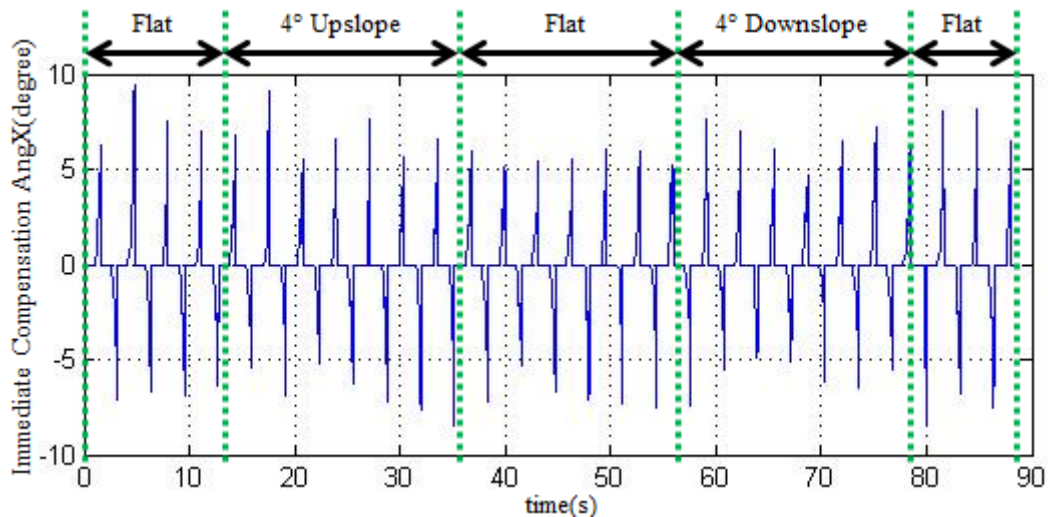


圖 5.7 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

行走時所偵測到在 X 方向的旋轉角度 AngX，如圖 5.6 所示，此感測值 AngX 隨上半身的左右擺盪呈現很規律的左右擺盪，由於抬腳時的重心影響和我們設計即時補償到關節 Hip-Roll 的即時補償控制器，即是希望當抬腳成功之後重心不會太往支撐腳方向偏，因此雖然我們設定左右搖擺的曲線為在 $\pm 19.5^\circ$ 之間，但感測值最多不超過 $\pm 16^\circ$ 。其中對於即時補償到關節 Hip-Roll 的即時補償控制值，如圖 5.7 所示。

5.2.2 平衡行走於 5° 斜坡之模擬結果

平衡行走於 5° 斜坡之模擬過程如圖 5.8(a)~(o) 所示，此時對於地形偵測的情況與補償控制器的作用效果說明如下：

行走時所偵測到在 Y 方向的旋轉角度 AngY，如圖 5.9 所示，此感測值 AngY 隨上半身的前後擺盪呈現擺盪的曲線，從整體變化上來看，在上坡時上半身有稍微的往前傾，另外，在下坡時上半身的傾斜角度仍有多向後傾斜，這些感測值表示補償控制器的作用導致上半身重心的擺動行為，能讓行走持續保持平穩。

如圖 5.10 顯示，預測補償控制器中，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，從圖中可以明顯的看出當機器人面臨到地形的變化時都能適時的進行上半身的修正，使得上半身能相對於水平地面持續保持直立，

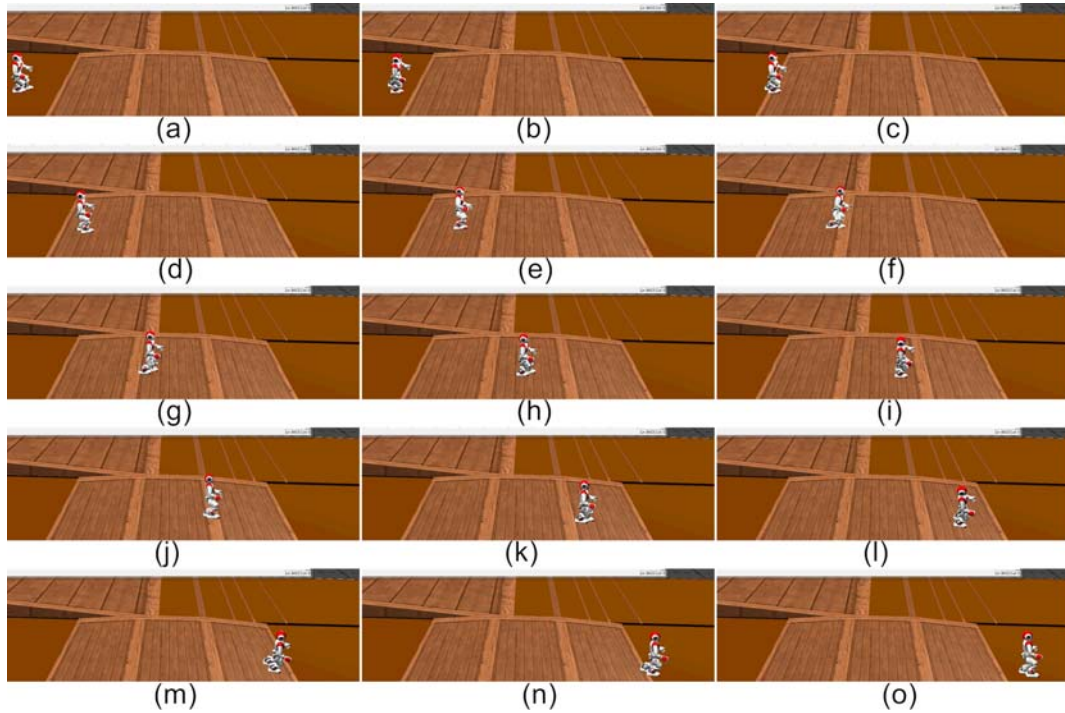


圖 5.8 平衡行走於 5° 斜坡之模擬過程

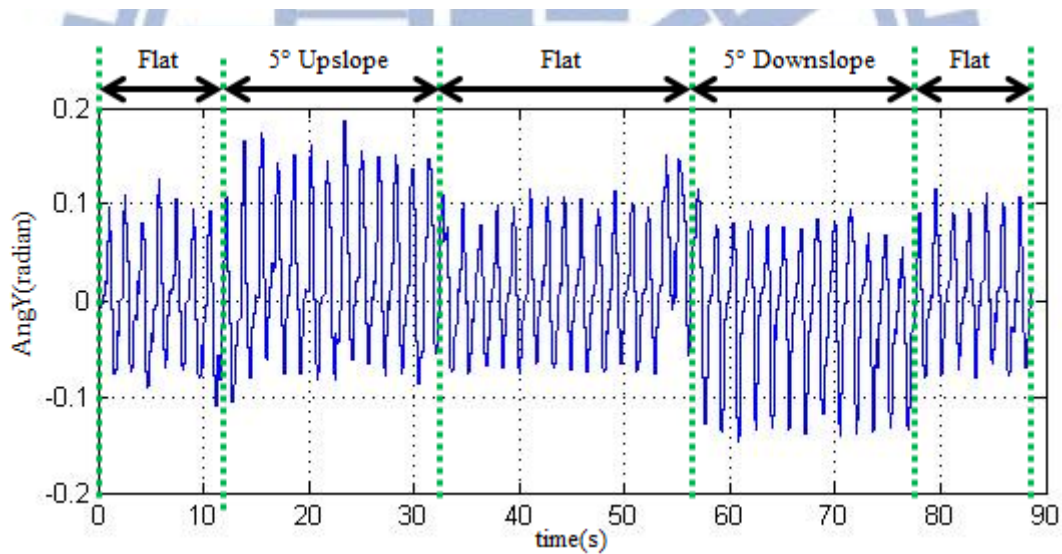


圖 5.9 行走過程中之 Y 方向旋轉角度曲線

在平地的預測補償值都在理想值 0° 附近；在上坡的預測補償值都在理想值 5° 附近；在下坡的預測補償值都在理想值 -5° 附近。

如圖 5.11 顯示，影響機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變的影響，曲線調整效果與行走在 4° 斜坡相同，皆為讓上半身在走到上坡有多前傾(正值變大)少後傾(負值變小)，且在下坡時多後傾少前傾的狀況發生。

如圖 5.12 顯示，在即時補償控制器中，對於補償到關節 Hip-Pitch 的即時角

度補償值，此補償器的效果與行走在 4° 斜坡相同，在平地產生小幅度的前後即時增加補償，在上下坡時則皆是往會讓行走更為穩定的擺盪方向進行即時補償。

行走時所偵測到在 X 方向的旋轉角度 $AngX$ ，如圖 5.13 所示，此感測值 $AngX$ 曲線的變化與行走在 4° 的斜坡相同。其中對於即時補償到關節 Hip-Roll 的即時補償控制值，如圖 5.14 所示，補償器效果也與行走在 4° 斜坡相同。

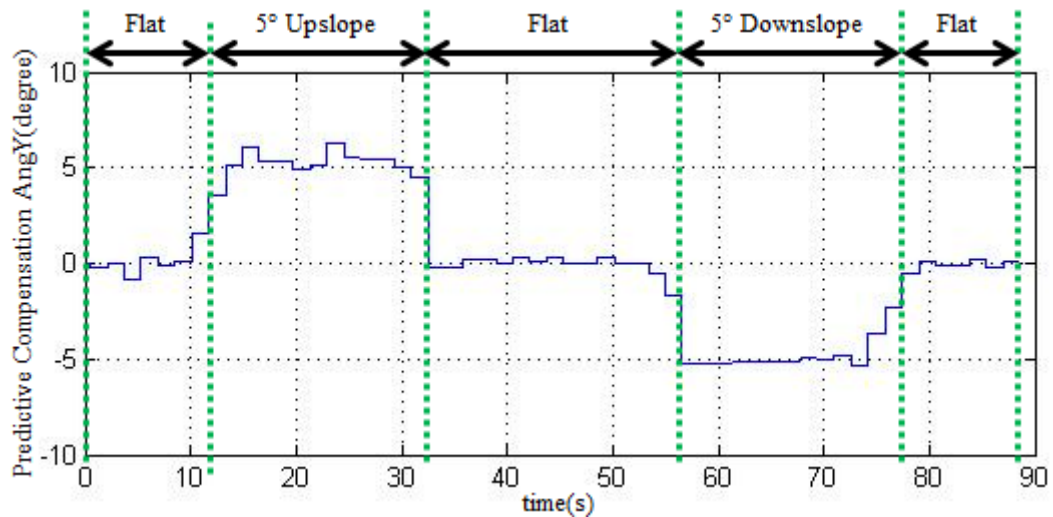


圖 5.10 直接補償於關節 Hip-Pitch 之預測補償角度值

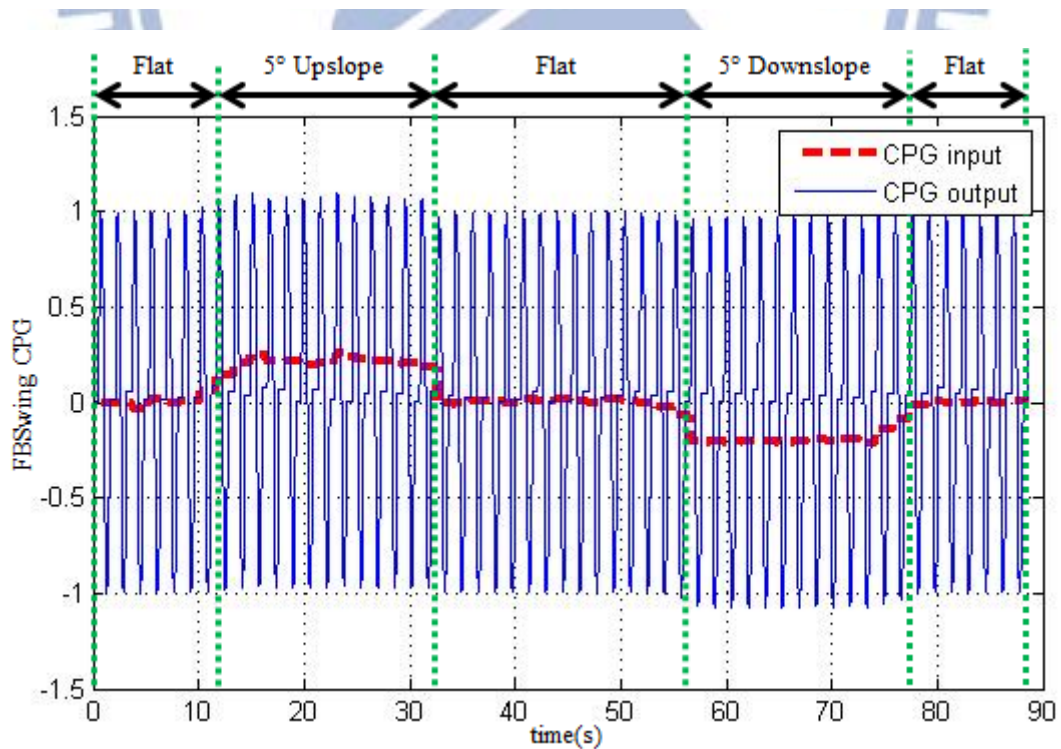


圖 5.11 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

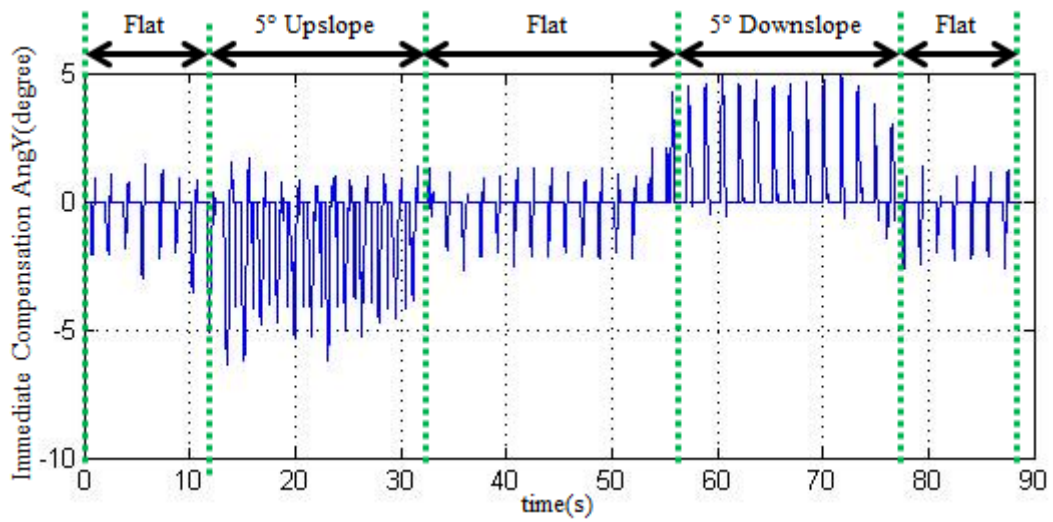


圖 5.12 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

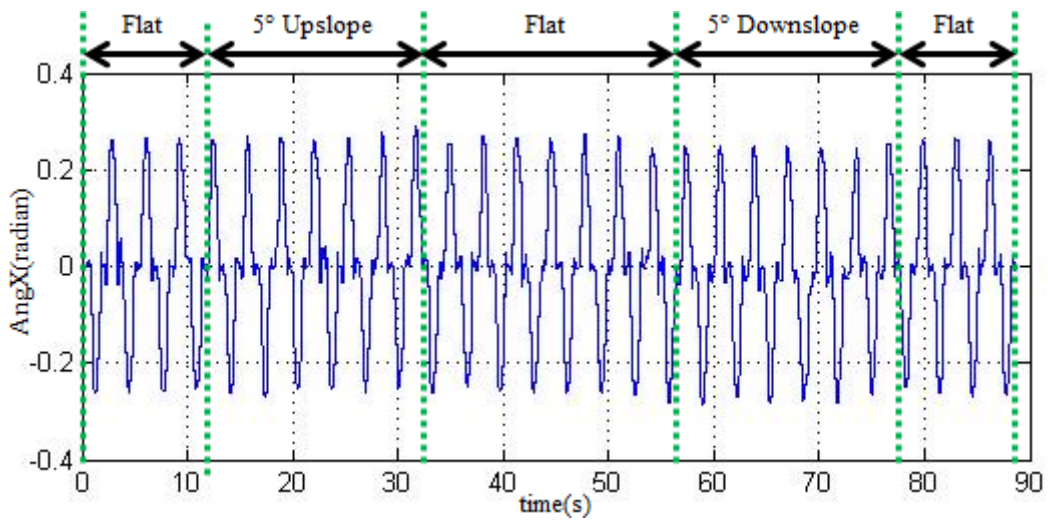


圖 5.13 行走過程中之 X 方向旋轉角度曲線

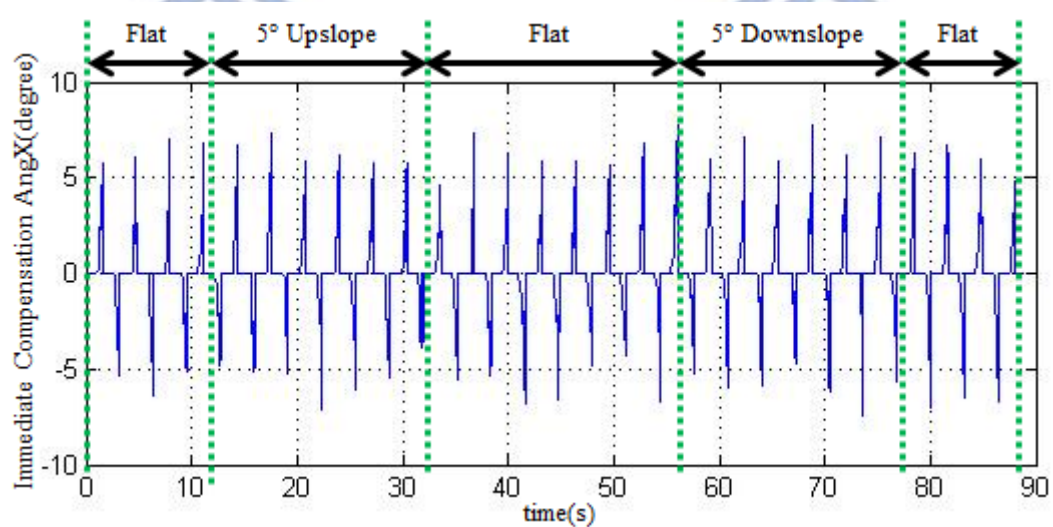


圖 5.14 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

5.2.3 平衡行走於 7° 斜坡之模擬結果

平衡行走於 7° 斜坡之模擬過程如圖 5.15 所示，此時對於地形偵測的情況與補償控制器的作用效果說明如下：

行走時所偵測到在 Y 方向的旋轉角度 AngY，如圖 5.16 所示，此感測值 AngY 曲線的變化與行走於 4° 的斜坡相同，且此圖顯示出控制器的作用效果更佳的明顯。

預測補償控制器中，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.17 所示，此補償效果與行走於 4° 和 5° 斜坡皆相同，在平地的預測補償值都在理想值 7° 附近；在上坡的預測補償值都在理想值 7° 附近；在下坡的預測補償值都在理想值 -7° 附近。在此圖中有一紅圈處為唯一預測補償值，補償了較理想值 7° 大了 2° 以上之值，在後面我們將搭配其他的圖進行說明，說明為何會發生預測誤差變大和即時補償器修正此錯誤之效果顯示。

如圖 5.18 顯示，控制機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變的影響，曲線調整效果與行走於 4° 斜坡相同，皆為讓上半身在走到上坡有多前傾(正值變大)少後傾(負值變小)，且在下坡時多後傾少前傾的狀況發生。

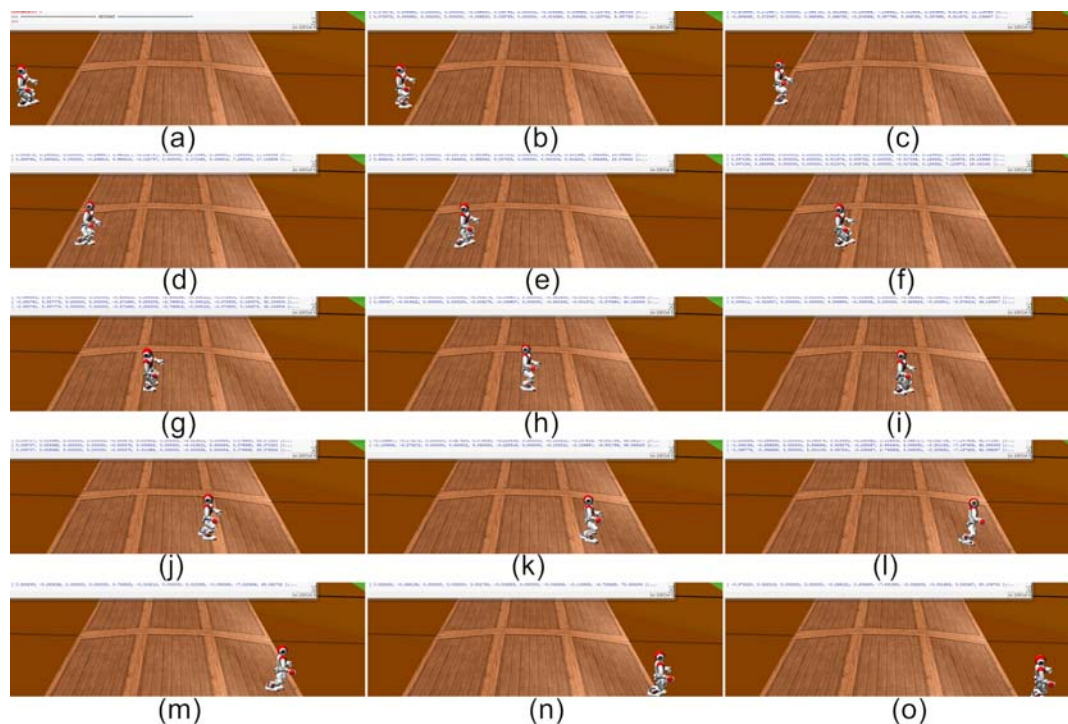


圖 5.15 平衡行走於 7° 斜坡之模擬過程

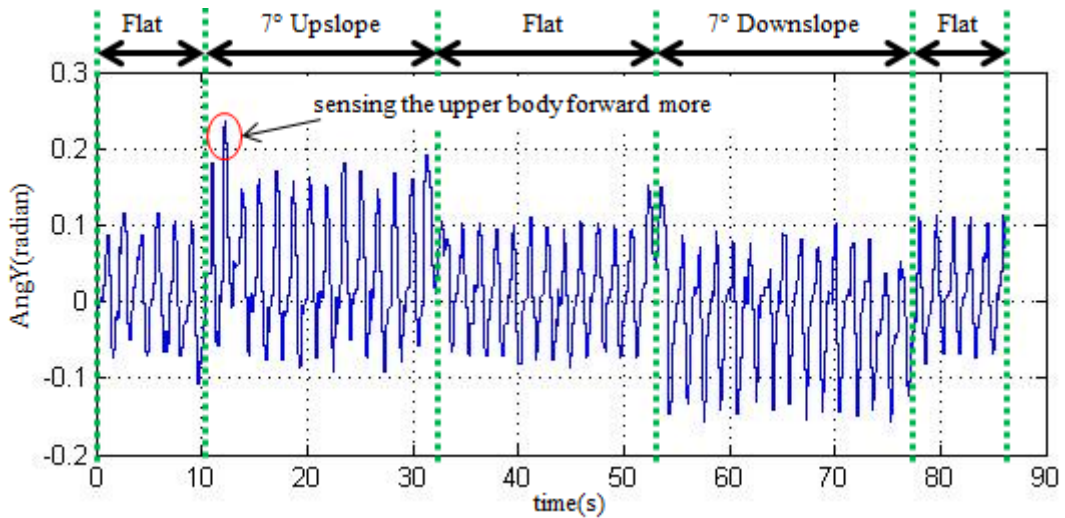


圖 5.16 行走過程中之 Y 方向旋轉角度曲線

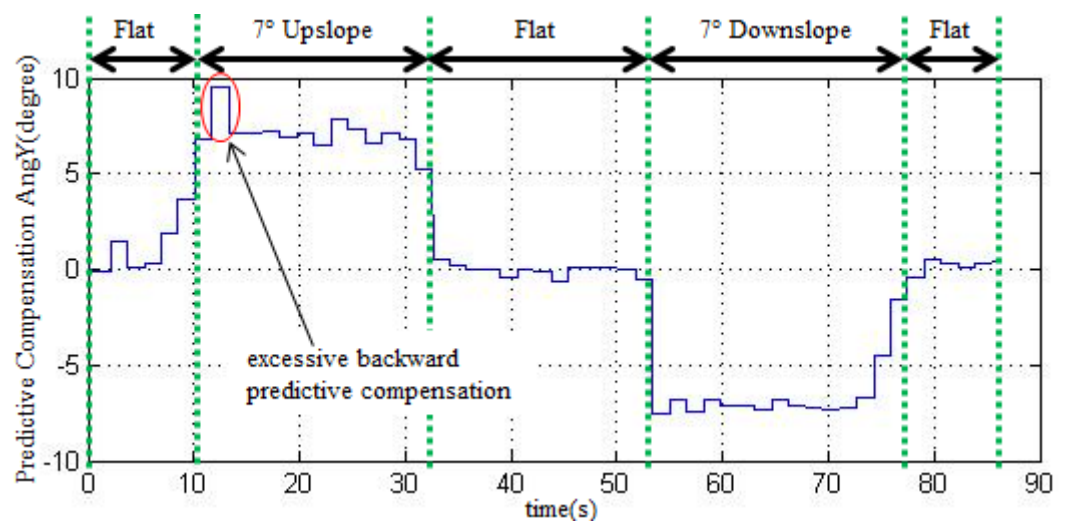


圖 5.17 直接補償於關節 Hip-Pitch 之預測補償角度值

如圖 5.19 顯示，在即時補償控制器中，對於補償到關節 Hip-Pitch 的角度補償值，此補償器效果大致上與行走於 4° 和 5° 斜坡相同，在平地產生小幅度的前後即時增加補償，在上下坡時則皆是往會使行走更為穩定的擺盪方向進行即時補償。但從圖 5.19 中的左上小紅圈顯示，在踏上 7° 上坡之第一步的後期，此時向後傾斜補償過多，因此導致預測補償角度計算出大於 7° 有 2° 以上的補償值(如圖 5.17 紅圈處顯示)，而此補償值也直接影響到 CPG 的輸出導致上半身擺盪更加往前傾(如圖 5.18 紅圈處顯示)、下一步的即時補償角度往前補償(負值)更多(如圖 5.19 中的左下小紅圈顯示)，這也導致行走時所偵測到在 Y 方向的感測資訊旋轉角度

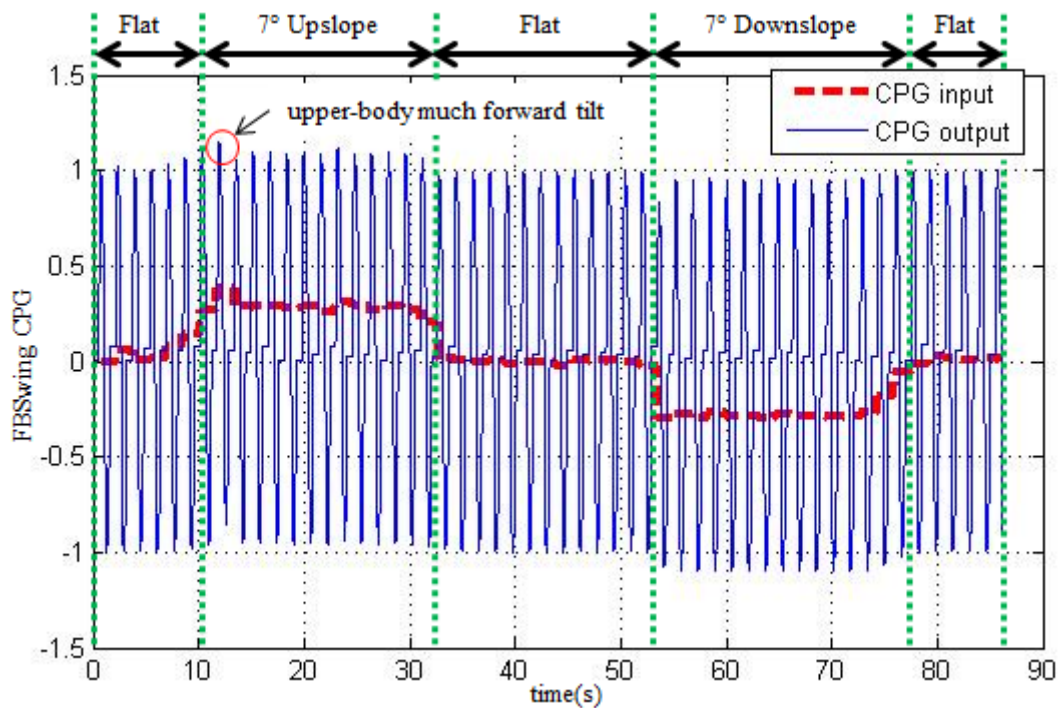


圖 5.18 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

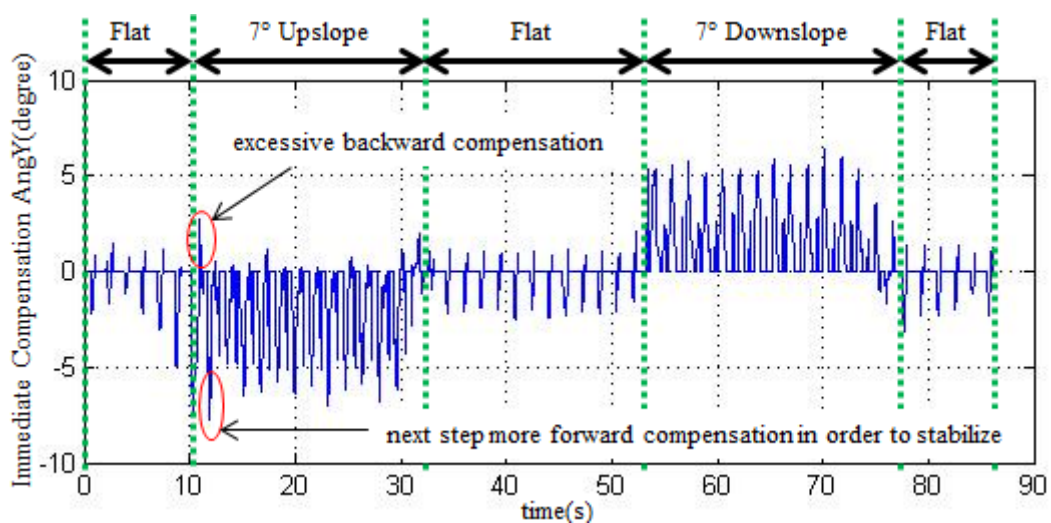


圖 5.19 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

AngY，偵測出較大的向前傾角度(如圖 5.16 紅圈處顯示)。

行走時所偵測到在 X 方向的旋轉角度 AngX，如圖 5.20 所示，此感測值 AngX 曲線的變化與行走於在 4° 的斜坡相同。其中對於即時補償到關節 Hip-Roll 的即時補償控制值，如圖 5.21 所示，補償器效果也與行走於 4° 斜坡相同。

從圖 5.15 中能看出當機器人行走於 7° 斜坡時，在 Y 方向機器人有往右偏的情形，由圖 5.20 也能顯示出機器人行走向右偏的情形，因為在上坡路段 AngX

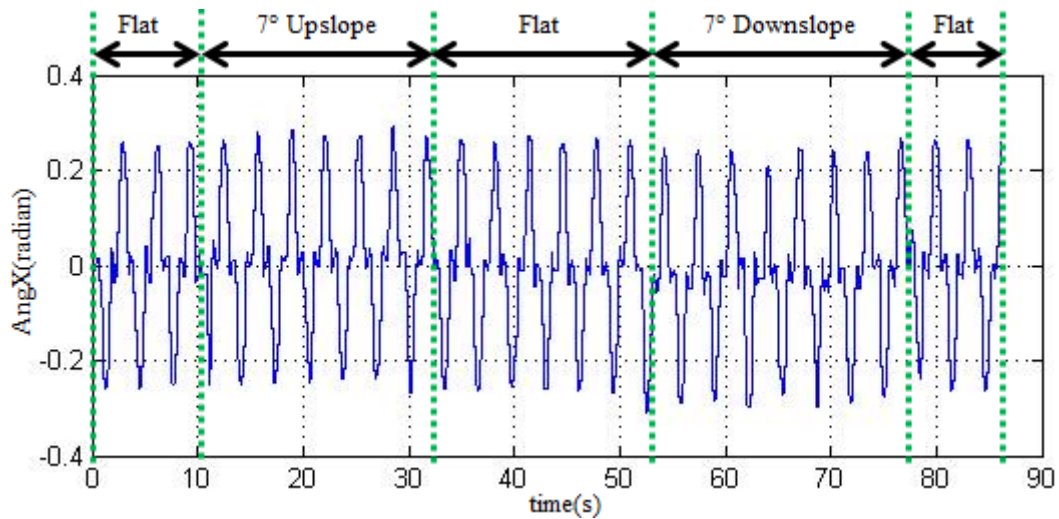


圖 5.20 行走過程中之 X 方向旋轉角度曲線

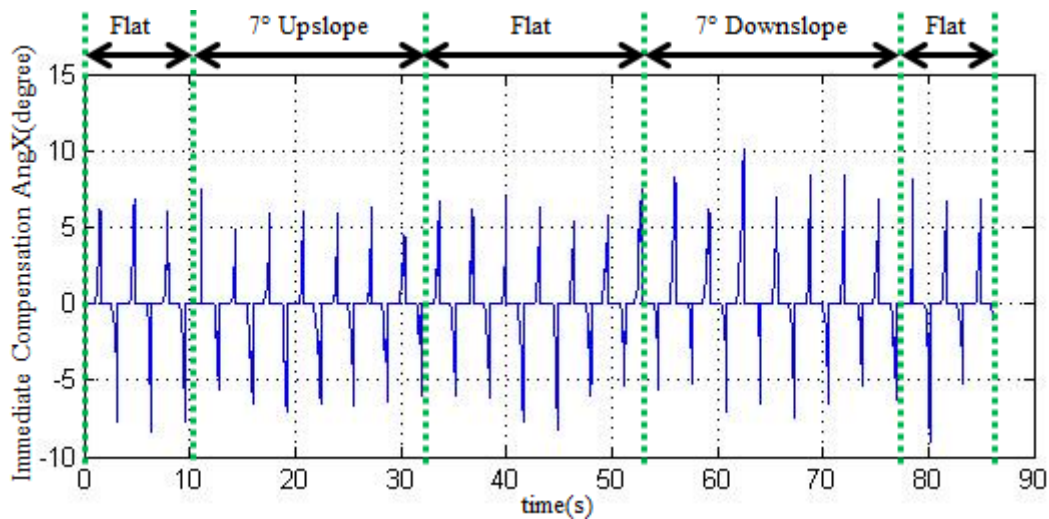


圖 5.21 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

正值變大負值變小(左高右低)、下坡路段 AngX 負值變大正值變小(右高左低)。

對於此行走偏移的情形，而導致走到上下坡時的左右腳高度的偏差，我們在 3.3 節提及仍然是希望透過即時補償於關節 Hip-Roll 之即時補償角度值來修正，由圖 5.21 顯示，此即時補償控制器有進行修正，在上坡路段還不明顯，但在下坡路段時能明顯的看出補償值面對到左腳為支撐相時往左擺盪的幅度被較大的減小(正值變大)，而在右腳為支撐相時往右擺盪的幅度被減小的值反而減少(負值變小)，顯示出當走偏時，重心分量對於左右方向的分量影響是不一樣的，而即時補償控制器有達成應有的補償控制。

5.2.4 模擬結果之比較

由圖 5.3、5.10 和 5.17 顯示，當雙足機器人面對斜坡角度變化時，偵測的預測補償值皆有快速的收斂到理想預期補償角度的附近，符合地形斜度。

接著對於預測補償值的偵測誤差上，如表 5.4 顯示，預測補償控制器在 4° 、 5° 和 7° 的斜坡環境中所計算出的預測補償值之平均值，皆與個別此地形所對應的理想值相差不到 0.5° ；而其中大於理想值最多為走在 7° 上坡的 9.523975° ，其餘之地形大於的理想值的差異值皆小於 2° ；另外，小於理想值最多為走在 7° 上坡的 5.224797° ，所有之地形小於理想值的差異值皆小於 2° 。由此得證，預測補償控制器透過上半身的姿態對地形進行斜度的判斷有良好的準確性，使得我們可以對上半身姿態進行預測的控制，藉此保持行走時的穩定度。

5.3 平衡行走之實驗結果

我們建立 4° 和 7° 的斜坡環境，如圖 5.22 所示，讓雙足機器人 NAO 完成在平地與兩種坡度的上坡和下坡之間行走的實驗，此實驗目的為驗證行走的步態規劃是符合靜態步行的設計，行走時的 ZMP 軌跡皆在 ZMP 的行走穩定區間中，以及驗證我們設計的控制方法在面對較複雜且非理想的環境因素及機器人的物理特性時，仍然有能力達成面對地形上的角度變化能得到適當的補償控制修正。

表 5.4 模擬結果之預測補償值比較

Simulation N-Terrain	Max Value	Min Value	Average Value
Simulation1- Flat	1.804273°	-1.256°	0.248313°
Simulation2- Flat	0.330695°	-0.85868°	-0.02962°
Simulation3- Flat	1.510835°	-0.57641°	0.110832°
Simulation1- 4° Upslope	5.496289°	3.487605°	4.197829°
Simulation2- 5° Upslope	6.311095°	3.575244°	5.207364°
Simulation3- 7° Upslope	9.523975°	5.224797°	7.089494°
Simulation1- 4° Downslope	-3.05464°	-4.4251°	-3.88774°
Simulation2- 5° Downslope	-4.77113°	-5.35198°	-5.11362°
Simulation3- 7° Downslope	-6.70269°	-7.52893°	-7.131°



(a)

(b)

圖 5.22 (a)4°之斜坡環境(b)7°之斜坡環境



圖 5.23 穩定行走之步態切片圖

5.3.1 步態驗證之平地行走

我們讓機器人 NAO 在無感測資訊回授進行平衡控制下，只利用到第二章之步態規劃的行走方式，在平地上進行穩定行走的測試，如圖 5.23(a)-(u)所示，圖

中為左右腳各走一步，行走一步的時間為 1.5 秒，其中每一小圖是以 0.15 秒為間隔切割出 21 張切片圖，順序為由圖 5.23(a)到圖 5.23(u)。理想步態規劃上我們讓機器人在單腳跨步時的跨步長度為 10 公分、跨步高度為 1.5 公分，在實際行走中，走完 10 步之後的長度為 49.2 公分，與理想長度的 50 公分只差了 0.8 公分，顯示出對於我們設計的跨步長度為準的，只是會受到滑步的影響產生少許誤差；另外，因為加入上半身左右搖擺作用，因此抬腳高度會因為臀關節的旋轉而有抬腳高度較為設定值高的情況產生。接著，對於行走時，單腳支撐相中四個 CPG 個別所產生出的曲線，如圖 5.24 所示，在圖中明顯的看出透過表 5.1 四個 CPG 的參數設計，四個 CPG 曲線皆符合達成跨步行走之控制要求，第一、三個 CPG 輸出之控制跨步長度 q_x 、跨步高度 q_z 和 Hip-Roll 擺盪角度 q_{hip_roll} 的 CPG 輸出週期為 Hip-Pitch 擺盪角度 q_{hip_pitch} 的 CPG 輸出週期的 0.5 倍；第二、因應跨步行走所需產生之 CPG 輸出曲線之間的相位差。另外，在行走步態中包含了雙腳支撐相，因此在圖中對於週期性的 CPG 曲線皆會有一段時間曲線暫時擺盪，讓雙腳支撐相時可以進行上半身之重心移動，而在行走中 CPG 每次暫停擺盪的值皆

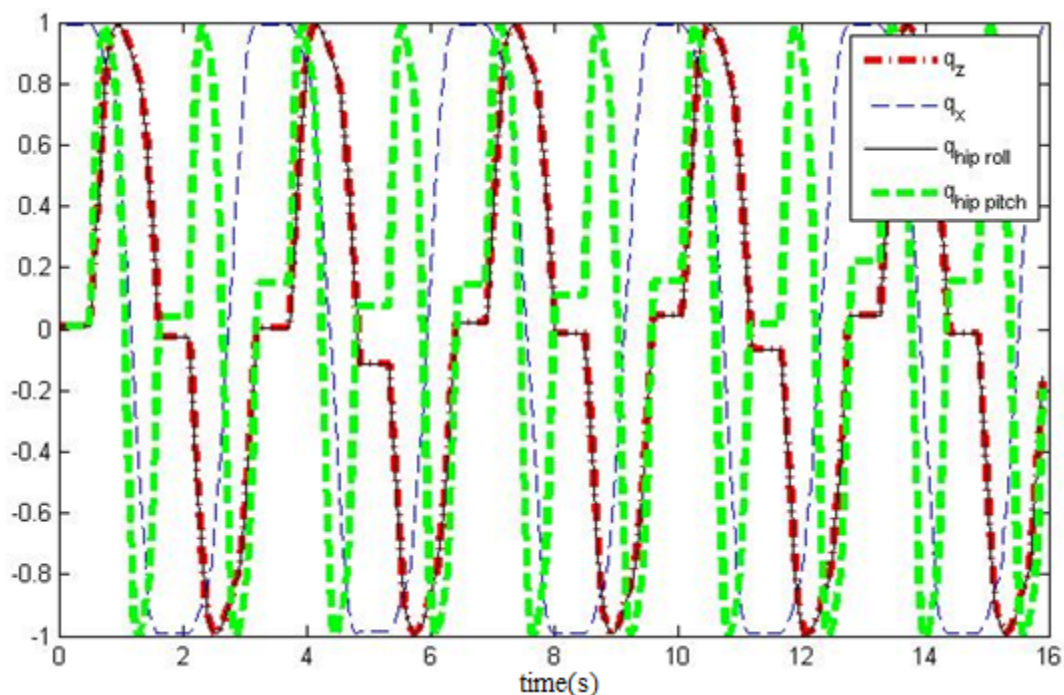


圖 5.24 步態行走時，四個 CPG 之個別振盪圖

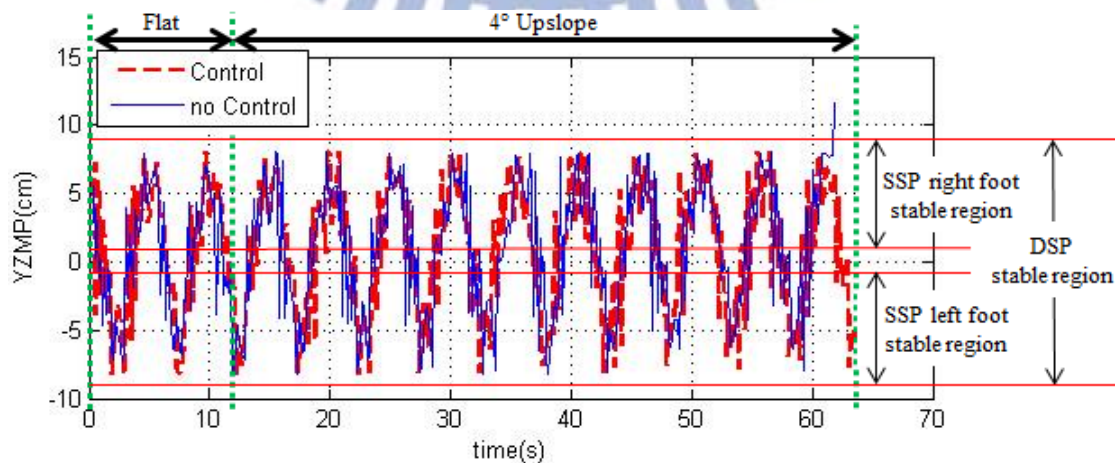
不一樣，雖然會讓雙腳支撐相時的姿態為非理想之直立姿勢，但並不影響行走之平衡，且每個 CPG 的週期皆不會因而改變，因此對於設計在週期性的單腳支撐相中加入非週期性的雙腳支撐相的整體步態規劃是可行的。

5.3.2 斜坡行走實驗結果之 ZMP 驗證

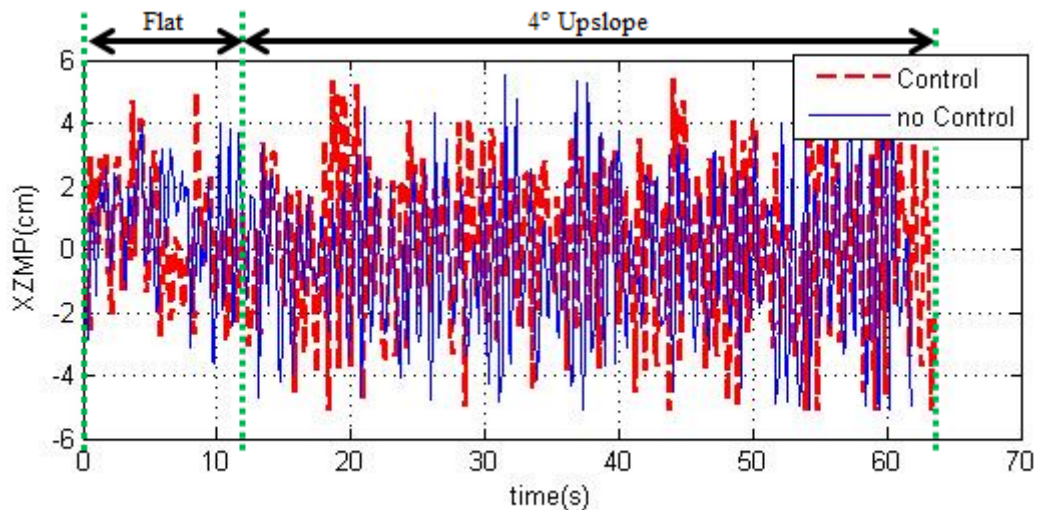
為了驗證行走的步態規劃是符合靜態步行的設計，機器人的重心投影必須全落在腳部的支撐多邊形區域內，因此在機器人平衡行走時，我們將再利用八個腳底壓力感測計(FSRs)，測量機器人與地面之間的作用力，經由計算來得到 ZMP 軌跡，藉此得知機器人行走時的 ZMP 軌跡曲線皆有在 ZMP 的行走穩定區間中。

5.3.2.1 平地→4°上坡之 ZMP 驗證

對於 NAO 從平地走上 4°上坡的實驗，我們做了單純利用步態規劃的行走和有加入補償控制器進行即時平衡控制兩組實驗來進行比較。首先，取得八個腳底壓力感測計(FSRs)行走時的即時更新之值，來計算出 ZMP 的軌跡，如圖 5.25 所示，其中(a)為 Y 方向(左右)的 ZMP 軌跡(b)為 X 方向(前後)的 ZMP 軌跡，不同相位 ZMP 的穩定區間變化在(a)圖中有標明，從(a)圖中看出沒利用控制器的行走較早跌出穩定區間失去平衡，(b)圖的 ZMP 軌跡都在穩定區間中，從此圖可看出當行走在上坡時，ZMP 軌跡大體上有多往前移動，表示有因上半身的控制前傾導致重心往前。



(a) ZMP trajectory of Y-direction



(b) ZMP trajectory of X-direction

圖 5.25 ZMP 軌跡

行走時上半身所偵測到在 Y 方向的旋轉角度 $AngY$ ，如圖 5.26 所示，由此圖可以看出上半身在控制器有作用時，在上坡路段時上半身的擺盪有控制為基於絕對於平地直立姿態的狀態在 0° 附近擺盪，讓上半身不會往後傾斜過多。

行走時上半身所偵測到在 X 方向的旋轉角度 $AngX$ ，如圖 5.27 所示，由此圖可以看出左右擺盪的幅度在有無控制中幾乎不變，但在最後無控制的明顯的往右邊跌倒，與 YZMP 軌跡離開穩定區間有相呼應。

在有用到補償控制器中，其中預測補償控制器，首先，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.28 所示，從圖中可以明顯

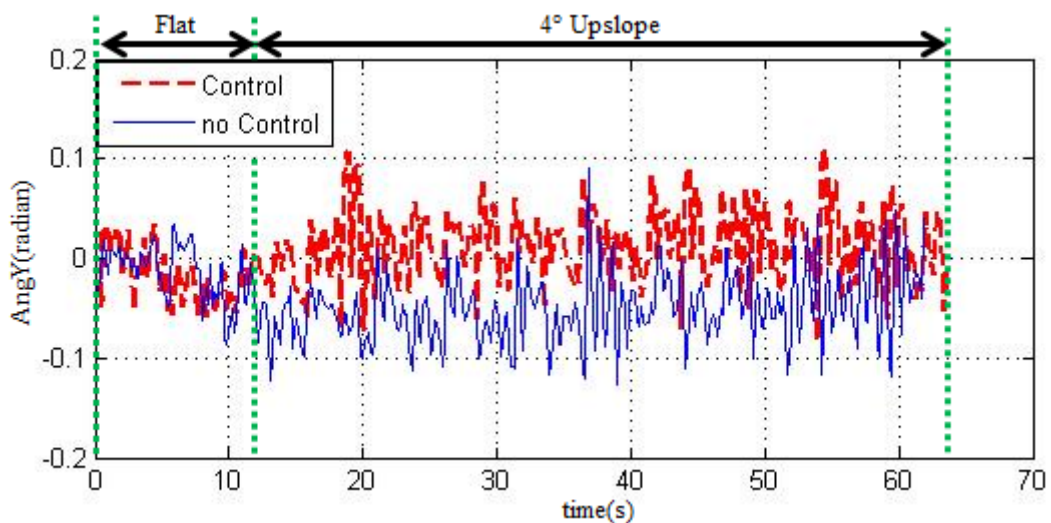


圖 5.26 行走過程中之 Y 方向旋轉角度曲線

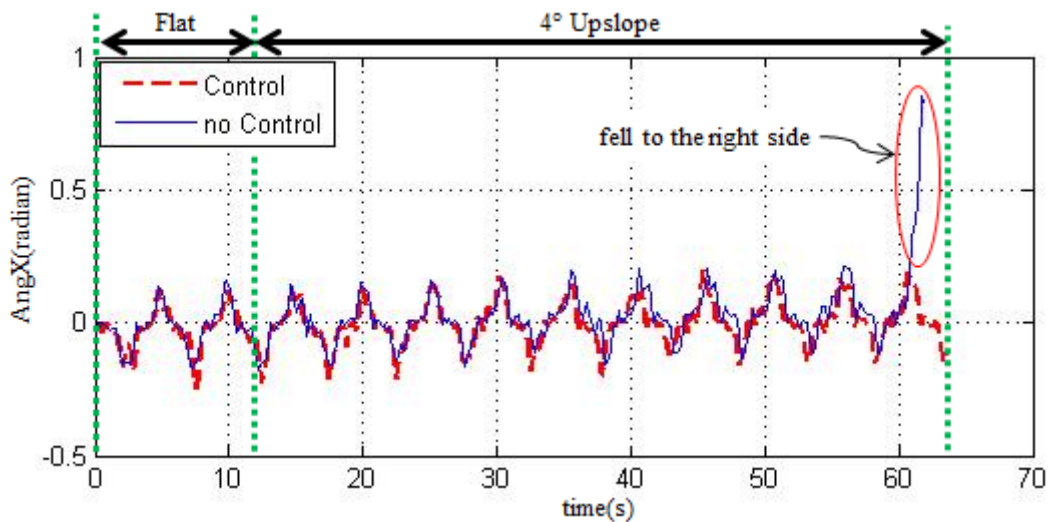


圖 5.27 行走過程中之 X 方向旋轉角度曲線

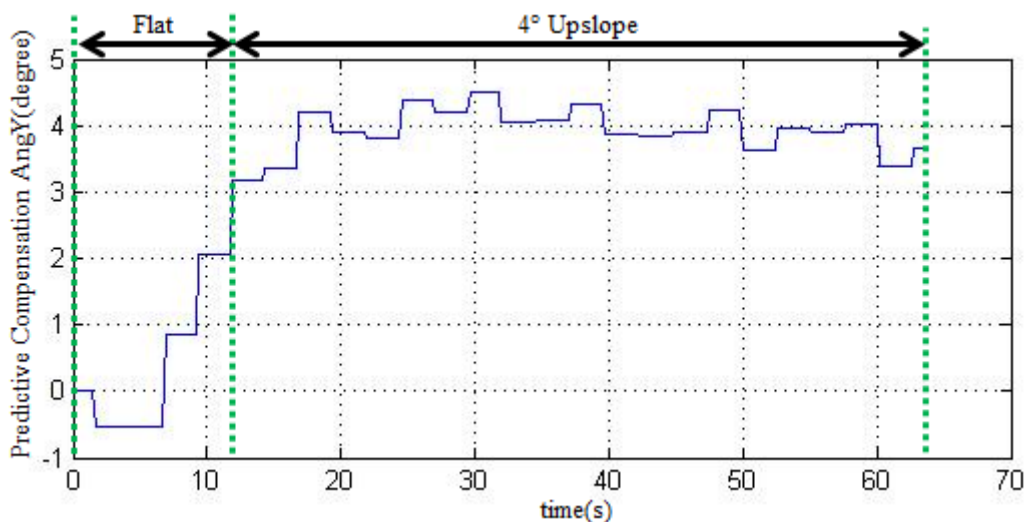


圖 5.28 直接補償於關節 Hip-Pitch 之預測補償角度值

的看出當機器人整個走上 4° 上坡時，感測出要補償的值與理想值 4° 只差了1度多，之後補償值即在 4° 保持穩定，使得上半身相對於水平地面會多前傾一些。接著，機器人上半身前後擺盪的CPG輸出曲線受到輸入曲線改變影響的圖，如圖5.29所示，其中CPG輸出曲線由於加了取得8個FSRs的感測值，增加了即時的運算時間，導致在控制上半身前後擺盪的CPG曲線計算上，由於運算時間加長無法產生出完整的CPG曲線，導致前後擺盪控制上的效能會大大的降低。而在控制行走的另外三個CPG曲線則因為週期為上半身前後擺盪CPG的兩倍，對於此運算時間的加長，仍然可以產生出完整的CPG曲線，較不受影響。

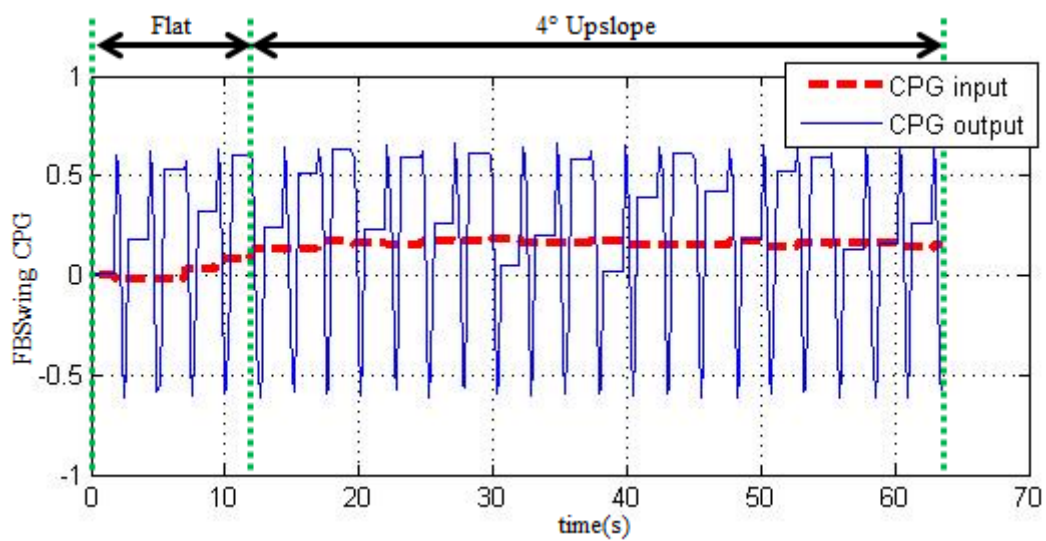


圖 5.29 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

接著，即時補償控制器的補償效果，對於即時補償到關節 Hip-Pitch 的角度補償值，如圖 5.30 所示，對於即時補償到關節 Hip-Roll 的角度補償值，如圖 5.31 所示，從兩圖中可以明顯地看出補償的值大部分均小於 1 度，甚至沒有補償產生，即時補償的效果不顯著。

從此實驗結果的數據來看，有加入控制器的能讓機器人行走的更加平穩，但因為加了 8 個 FSRs 的感測值取得，增加了即時的運算時間，導致補償控制會因為延遲的感測資訊導致補償的效果不佳或是誤差產生，因此接下來在以 ZMP 驗證平衡為主的的不同地形行走實驗，將不會進行補償控制器詳細的效果說明。

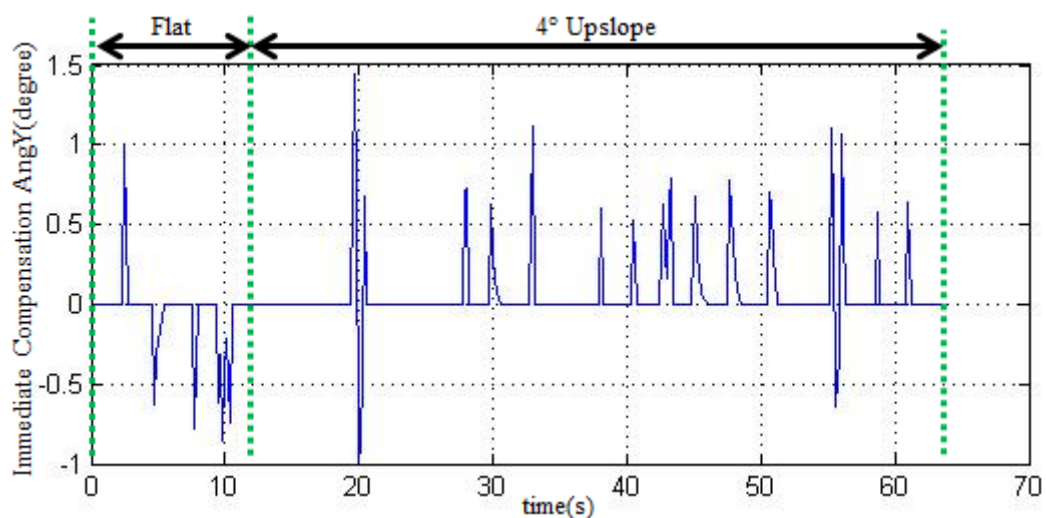


圖 5.30 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

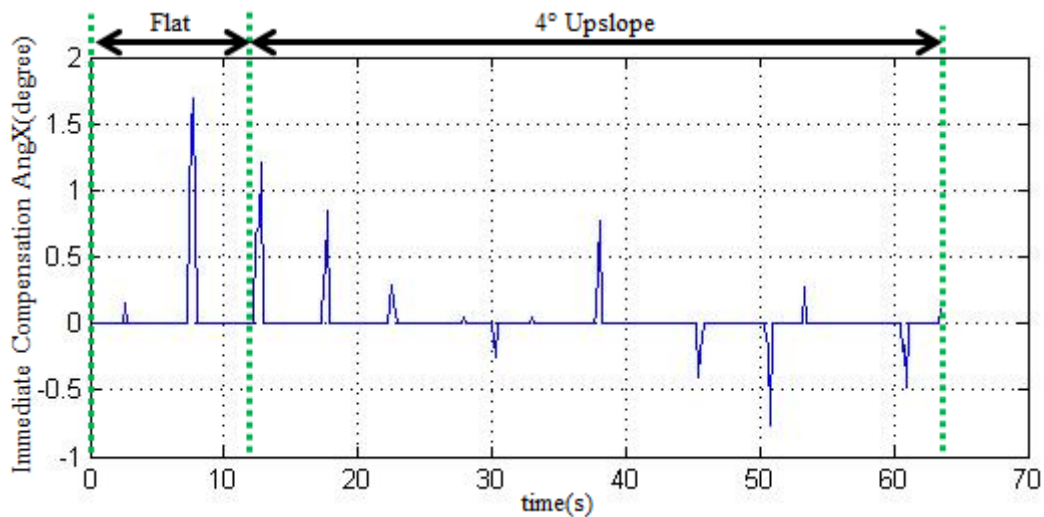
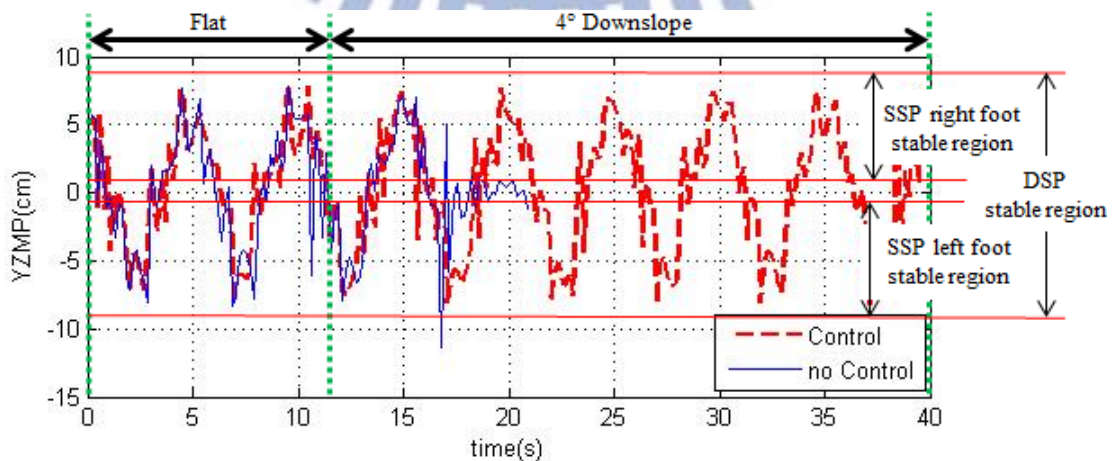


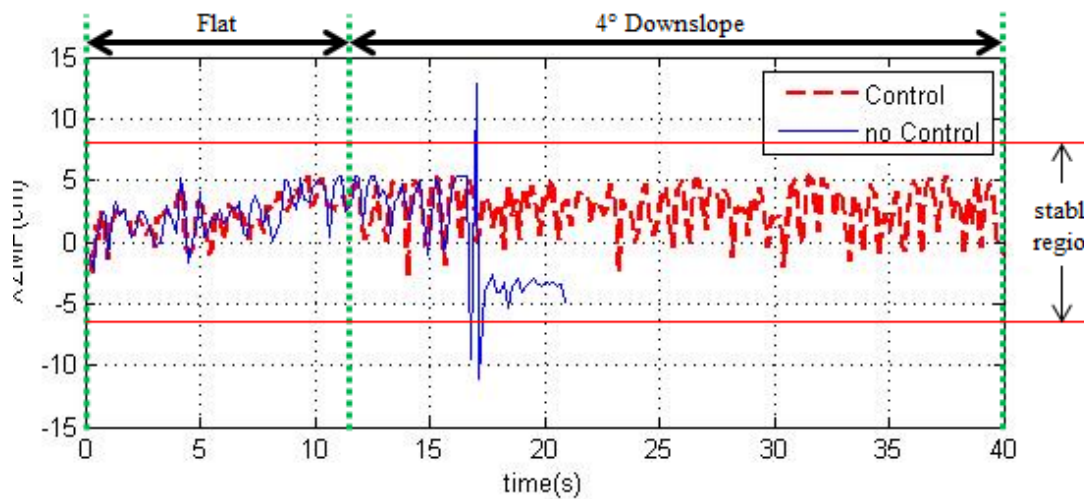
圖 5.31 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

5.3.2.2 平地→4°下坡之 ZMP 驗證

對於 NAO 從平地走下 4°下坡的實驗，我們仍然做了單純利用步態規劃的行走和有加入補償控制器進行即時平衡控制兩組實驗來進行比較。行走時 ZMP 的軌跡，如圖 5.32 所示，其中(a)為 Y 方向的 ZMP 軌跡(b)為 X 方向的 ZMP 軌跡，兩圖中的紅色橫線之間為 ZMP 的穩定區間，從(a)圖中可以看出沒利用控制器的行走進入下坡後的第三步 SSP 左腳站時就跌出 ZMP 的穩定區間(左右晃)失去平衡，此時從(b)圖看出同一時間 X 方向也跌出 ZMP 的穩定區間，且可看出當行走在下坡時，ZMP 軌跡往後曲線有較多，表示有因上半身的控制往後傾導致重心往後。



(a) ZMP trajectory of Y-direction



(b) ZMP trajectory of X-direction

圖 5.32 ZMP 軌跡

行走時上半身所偵測到在 Y 方向的旋轉角度 $AngY$ ，如圖 5.33 所示，由此圖可以看出，無控制的情況走至下坡時往前跌倒，與 XZMP 軌跡離開穩定區間有相呼應。上半身在控制器有作用時，在下坡路段時上半身的擺盪有控制為基於絕對於平地直立姿態的狀態在較接近 0° 附近擺盪，讓上半身不會往前傾斜過多。

行走時上半身所偵測到在 X 方向的旋轉角度 $AngX$ ，如圖 5.34 所示，由此圖可以看出左右擺盪的幅度皆在 0° 擺盪，而無控制的在下坡後往右邊跌倒，與 YZMP 軌跡離開穩定區間有相呼應。

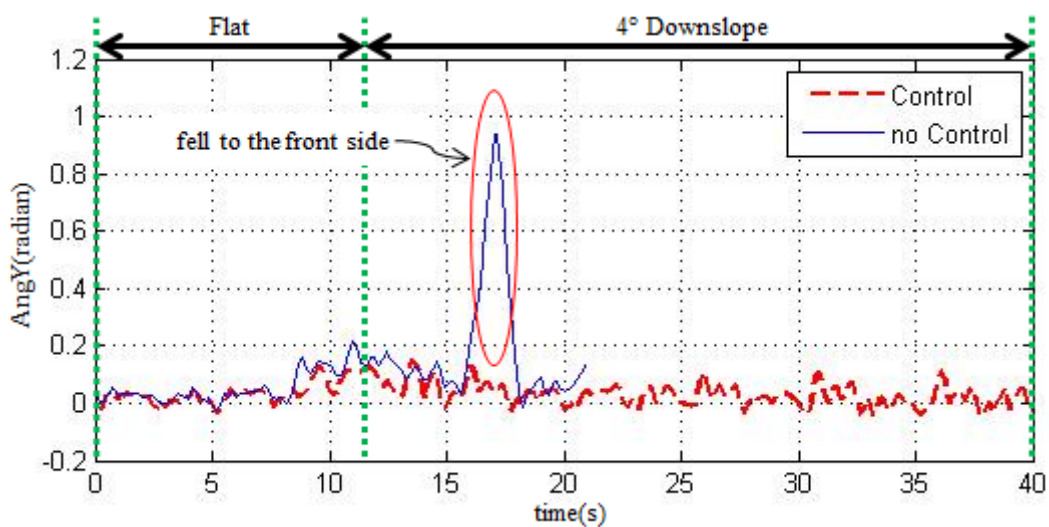


圖 5.33 行走過程中之 Y 方向旋轉角度曲線

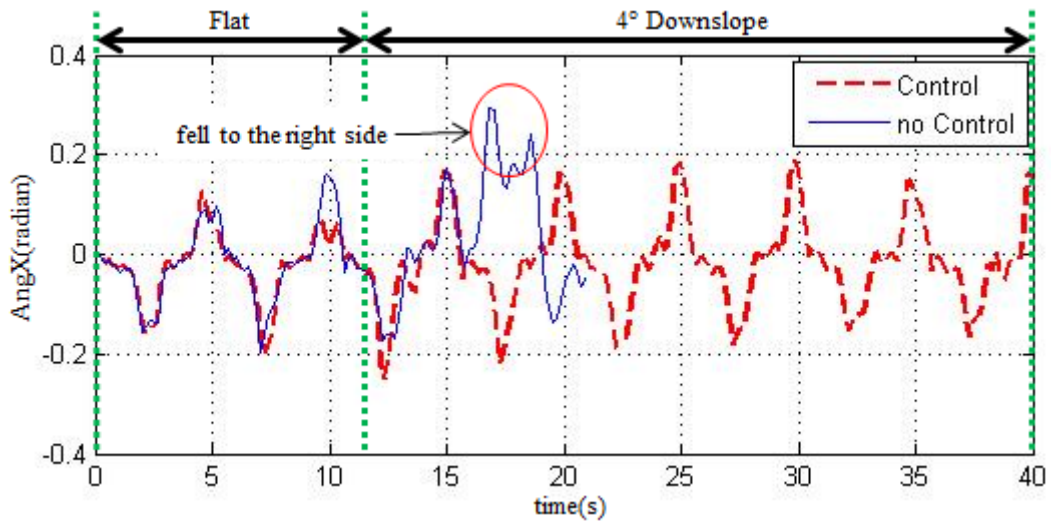
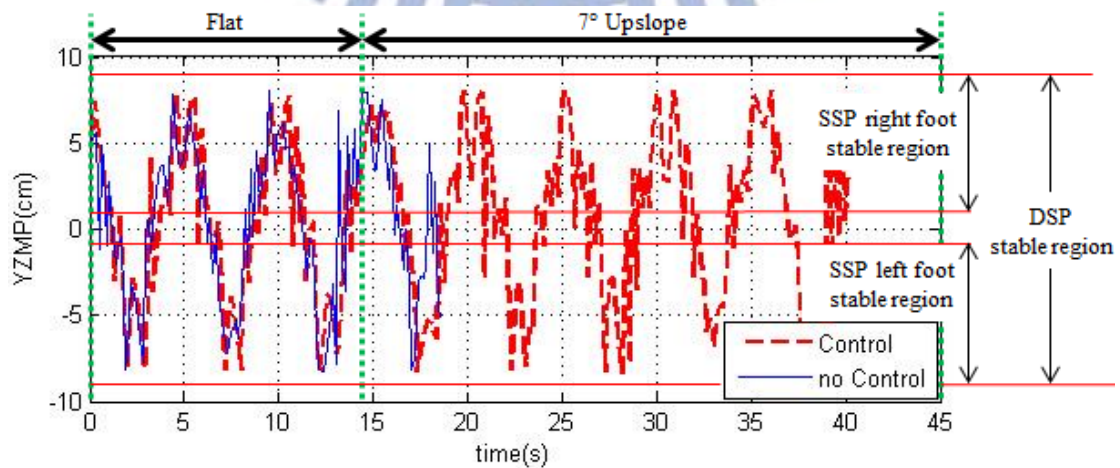


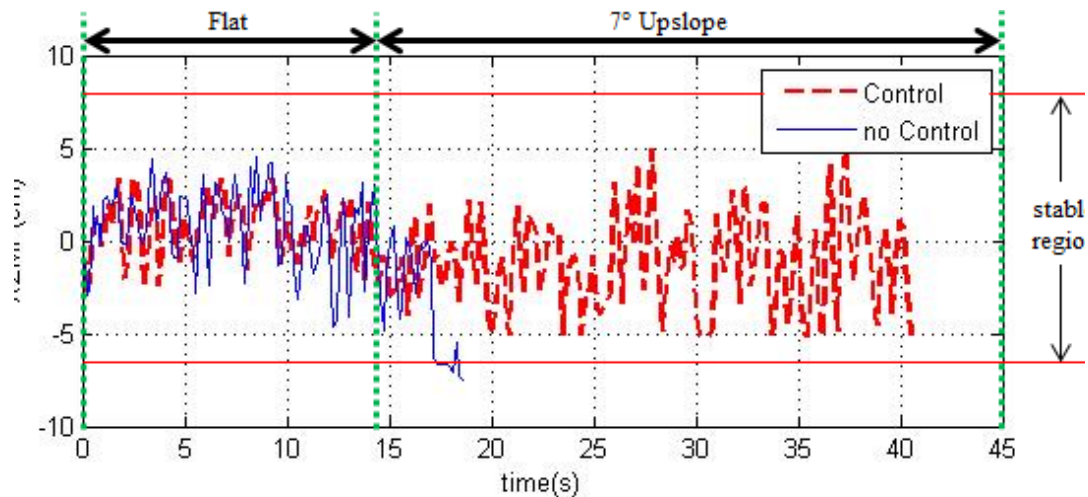
圖 5.34 行走過程中之 X 方向旋轉角度曲線

5.3.2.3 平地→7°上坡之 ZMP 驗證

對於 NAO 從平地走上 7° 上坡的實驗，我們仍然做了單純利用步態規劃的行走和有加入補償控制器進行即時平衡控制兩組實驗來進行比較。行走時 ZMP 的軌跡，如圖 5.35 所示，其中(a)為 Y 方向的 ZMP 軌跡(b)為 X 方向的 ZMP 軌跡，兩圖中的紅色橫線之間為 ZMP 的穩定區間，從(a)圖中可以看出沒利用控制器的行走進入上坡後的第二步 SSP 左腳站時就跌出 ZMP 的穩定區間(往右倒)失去平衡，此時從(b)圖看出同一時間 X 方向也跌出 ZMP 的穩定區間，且可看出當行走在上坡時，ZMP 軌跡沒有因上坡往後移動，表示有因上半身的控制前傾導致重心往前。



(a) ZMP trajectory of Y-direction



(b) ZMP trajectory of X-direction

圖 5.35 ZMP 軌跡

行走時上半身所偵測到在 Y 方向的旋轉角度 $AngY$ ，如圖 5.36 所示，由此圖可以看出，無控制的情況走至上坡時往後跌倒，與 XZMP 軌跡離開穩定區間有相呼應。上半身在控制器有作用時，在上坡路段時上半身的擺盪有控制為基於絕對於平地直立姿態的狀態在較接近 0° 附近擺盪，讓上半身不會往後傾斜過多。

行走時上半身所偵測到在 X 方向的旋轉角度 $AngX$ ，如圖 5.37 所示，由此圖可以看出左右擺盪的幅度皆在 0° 擺盪，而無控制的在上坡後往右邊跌倒，與 YZMP 軌跡離開穩定區間有相呼應。

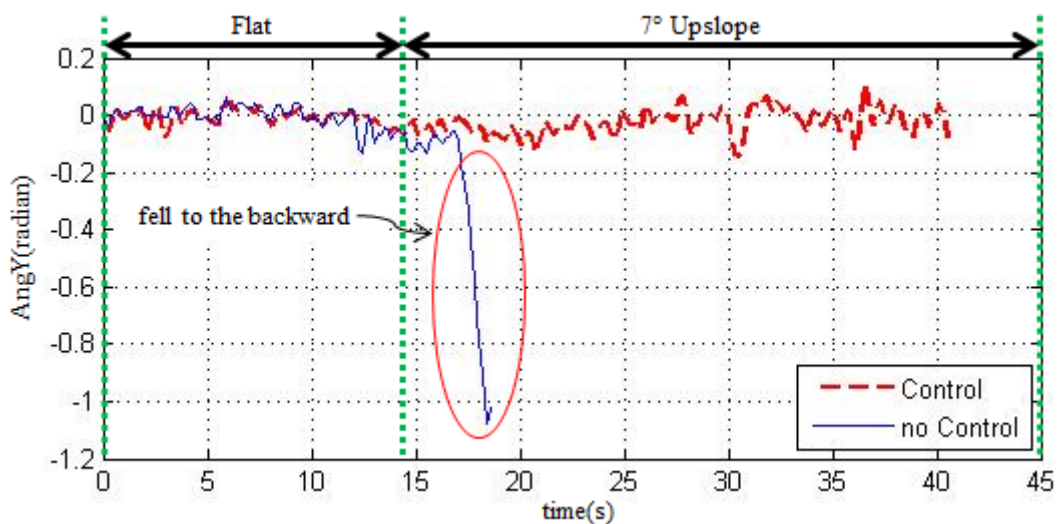


圖 5.36 行走過程中之 Y 方向旋轉角度曲線

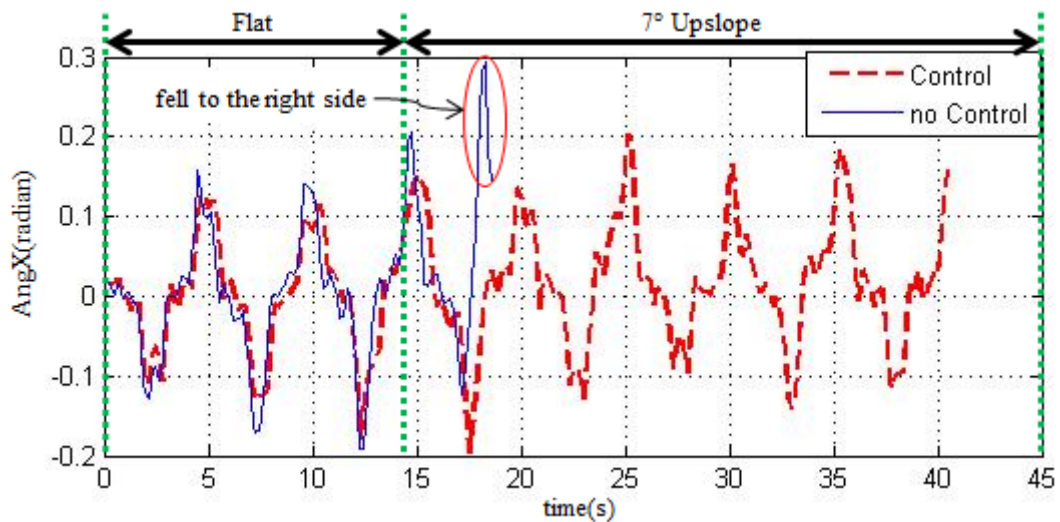
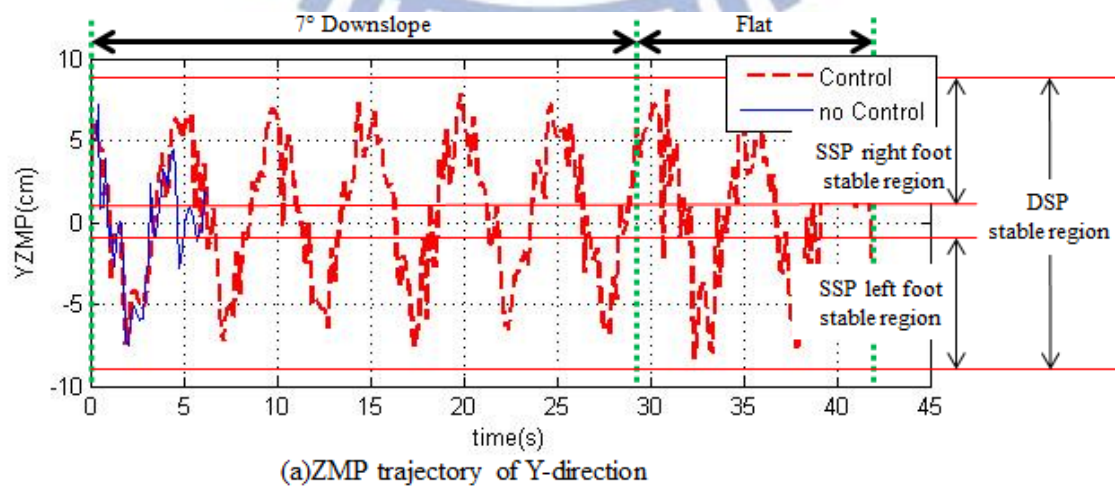
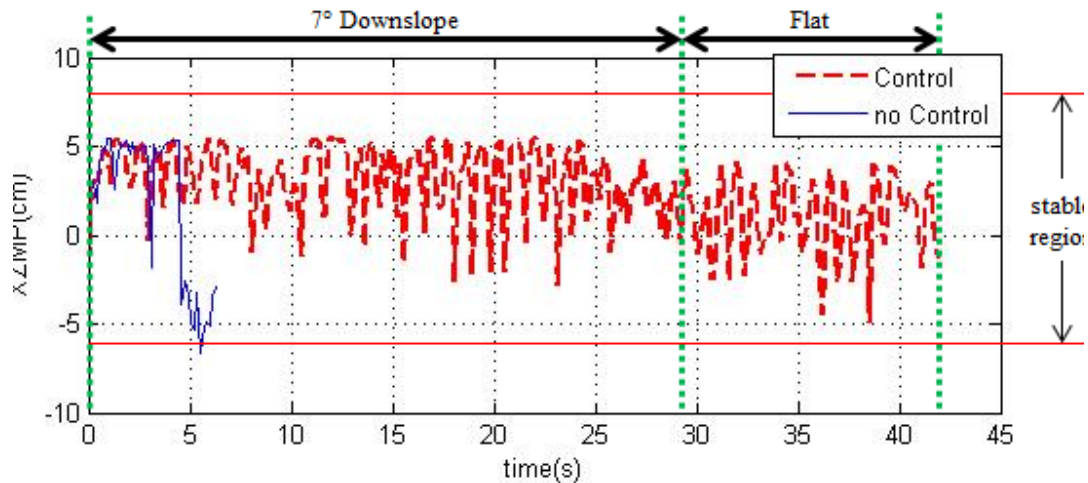


圖 5.37 行走過程中之 X 方向旋轉角度曲線

5.3.2.4 7°下坡→平地之 ZMP 驗證

對於 NAO 從 7°下坡走到平地的實驗，我們仍然做了單純利用步態規劃的行走和有加入補償控制器進行即時平衡控制兩組實驗來進行比較。行走時 ZMP 的軌跡，如圖 5.38 所示，其中(a)為 Y 方向的 ZMP 軌跡(b)為 X 方向的 ZMP 軌跡，兩圖中的紅色橫線之間為 ZMP 的穩定區間，從(a)圖中可以看出沒利用控制器的行走下坡開始的第二步 SSP 右腳站時就跌出 ZMP 的穩定區間(往左倒)失去平衡，此時從(b)圖看出同一時間 X 方向也跌出 ZMP 的穩定區間，另外可看出當行走在 7°下坡時，ZMP 軌跡與 4°下坡相比沒有較往前，表示有因上半身控制後傾導致重心保持在與 4°下坡差不多的位置。





(b) ZMP trajectory of X-direction

圖 5.38 ZMP 軌跡

行走時上半身所偵測到在 Y 方向的旋轉角度 $AngY$ ，如圖 5.39 所示，由此圖可以看出，無控制的情況下坡一開始就往前跌倒，與 XZMP 軌跡離開穩定區間有相呼應。上半身在控制器有作用時，在下坡路段時上半身的擺盪有控制為基於絕對於平地直立姿態的狀態在較接近 0° 附近擺盪，讓上半身不會前傾過多。

行走時上半身所偵測到在 X 方向的旋轉角度 $AngX$ ，如圖 5.40 所示，由此圖可以看出左右擺盪的幅度皆在 0° 擺盪，而無控制的在上坡後往左邊跌倒，與 YZMP 軌跡離開穩定區間有相呼應。

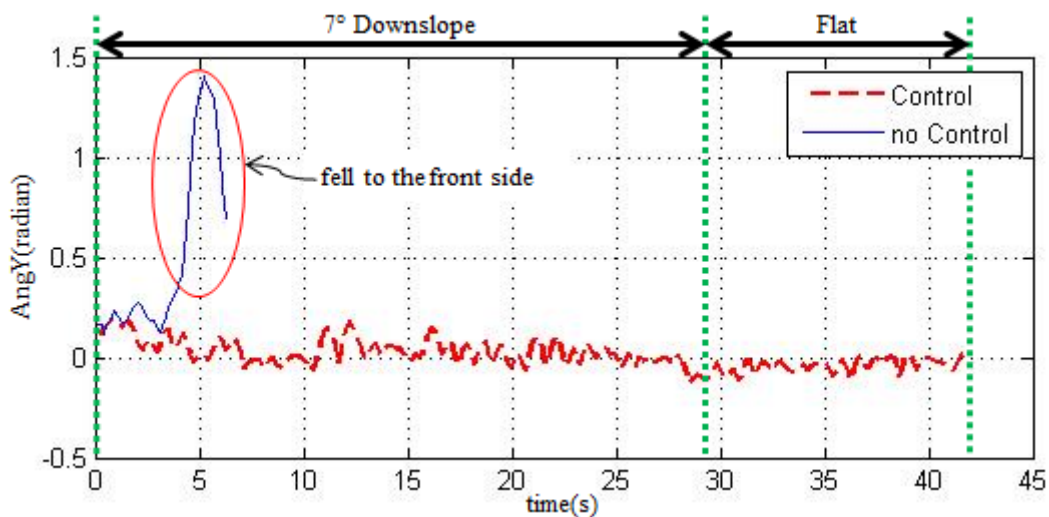


圖 5.39 行走過程中之 Y 方向旋轉角度曲線

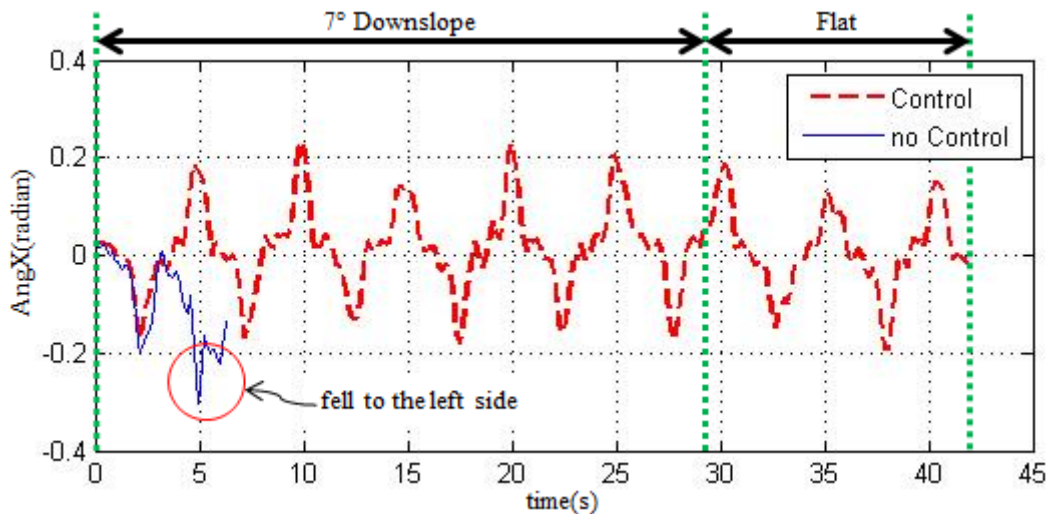


圖 5.40 行走過程中之 X 方向旋轉角度曲線

5.3.3 斜坡行走實驗結果之控制器效果驗證

由於在 5.3.2 節中已驗證了行走步態的設計能讓機器人在行走時 ZMP 都能保持在穩定的區間當中，因此在此節當中我們將不再即時取得 8 個 FSRs 的感測值，藉此讓即時的運算時間加快，而從實驗數據中得知，確實每個運算週期平均快了 0.09s。因此整體控制器的效果會顯著許多，機器人行走因而更加平穩可以在更多的地形變化中進行行走，所以在接下來的實驗中，我們讓機器人在平地→4°上坡→平地、平地→4°下坡→平地、平地→7°上坡和 7°下坡→平地，四種地形之間變化的情況進行四次實驗，以驗證控制器效果及在不同坡度中補償值的比較。

5.3.3.1 平地→4°上坡→平地之行走實驗結果

平地→4°上坡→平地之行走實驗過程如圖 5.41 所示，此時對於地形偵測的情況與補償控制器的作用效果說明如下：

行走時所偵測到在 Y 方向的旋轉角度 AngY，如圖 5.42 所示，此感測值 AngY 隨上半身的前後擺盪呈現擺盪的曲線，從整體變化上來看，在上坡路段上半身有往前傾維持穩定。但在最後的平地路段，一開始上半身產生向前傾的振動，使得機器人不穩定(圖 5.42 的紅圈處)，此時透過即時補償控制器，即時補償在關節 Hip-Pitch 的即時補償值，讓上半身重心有即時較大往後(正值)旋轉傾斜(圖 5.45

的紅圈處)，使得上半身產生即時修正，藉此讓機器人行走能回歸平穩。

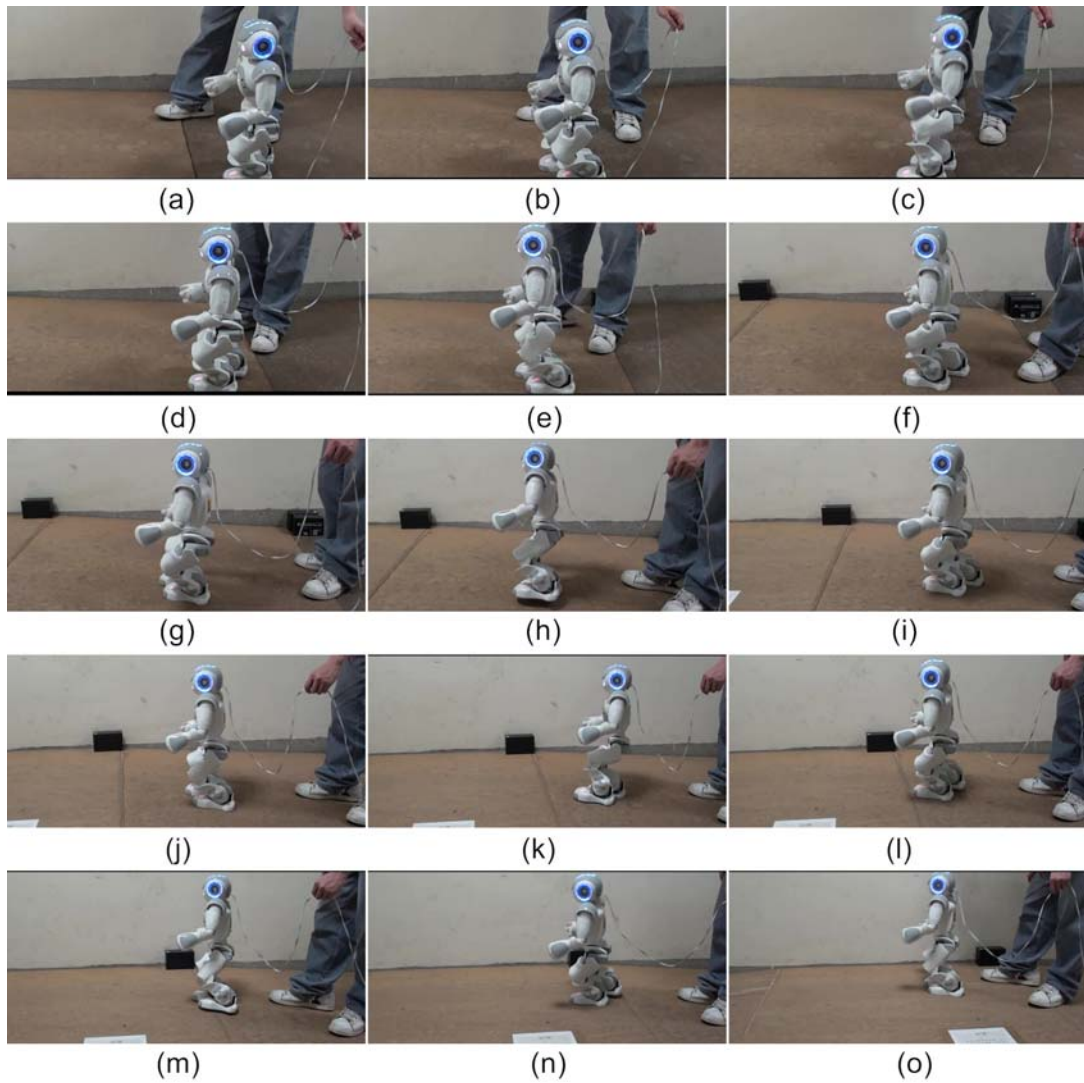


圖 5.41 平地→4°上坡→平地之行走實驗過程

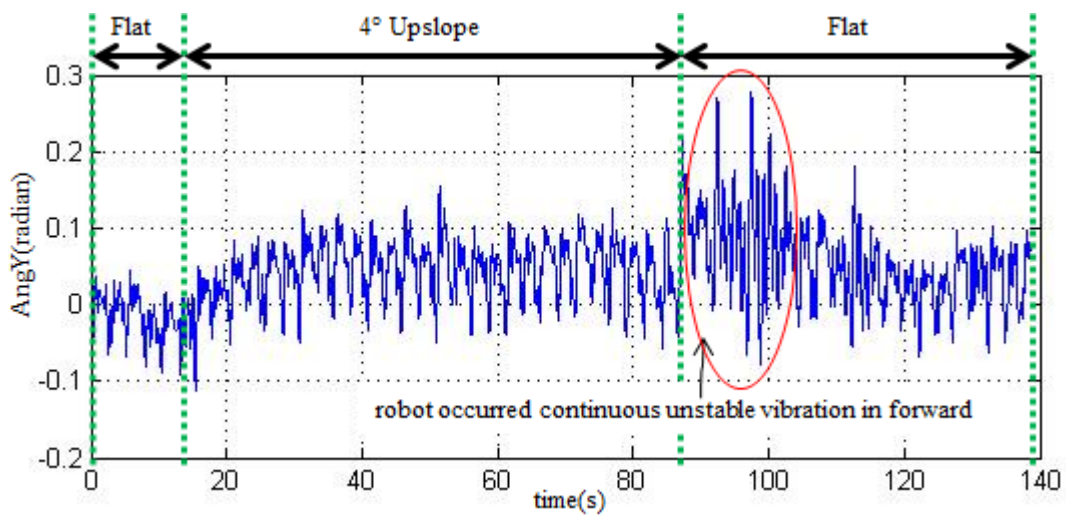


圖 5.42 行走過程中之 Y 方向旋轉角度曲線

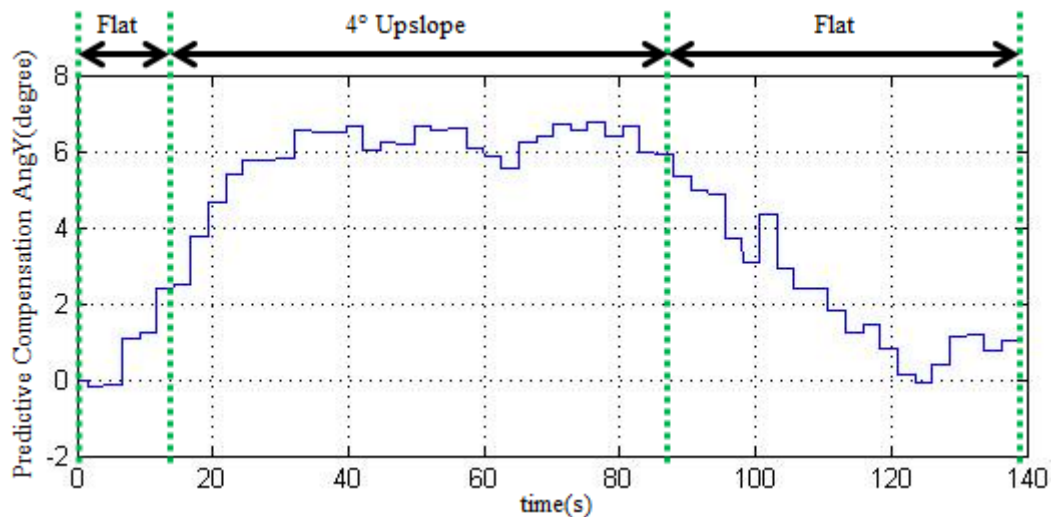


圖 5.43 直接補償於關節 Hip-Pitch 之預測補償角度值

預測補償控制器中，首先，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.43 所示，從圖中可以明顯的看出當機器人整個走上 4° 上坡時，感測出要補償的值與理想值 4° 只差了 1 度多，但是補償值在之後的幾步一直增加到 6.5 度才保持穩定，使得上半身相對於水平地面會多前傾一些，如圖 5.42 所示；而當再度走到平地時，從圖 5.43 中可看出補償值降到 0 度的收斂速度較為緩慢，但最後仍然能降到 1 度附近保持穩定。接著，影響機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變影響，使得上半身在走到上坡有多前傾(正值變大)少後傾(負值變小)，如圖 5.44 所示，圖中在 CPG 曲線之間的橫線為雙腳支撐相，由圖顯示此時的真實中之上半身會比模擬的情況更加的前傾。

在即時補償控制器中，對於即時補償到關節 Hip-Pitch 的角度補償值，如圖 5.45 所示，在此圖可以明顯地顯示出當走到上坡時，在預測補償值小於理想補償值 4 度時，前兩步補償值仍然有將上半身重心往前(負值)旋轉傾斜，而當預測補償值大於 4 度且成長到平均為 6 度時，即時補償值也持續補償了 1.75 度，使得上半身重心往後(正值)旋轉傾斜，使得整體補償理想值仍然接近於 4 度；而當再度走到平地時，預測補償值下降的較為緩慢，從即時補償值來看，此時依然是有將上半身重心往後(正值)旋轉傾斜，使得整體補償理想值更趨近於 0 度，達到遵守行走曲線的行走。

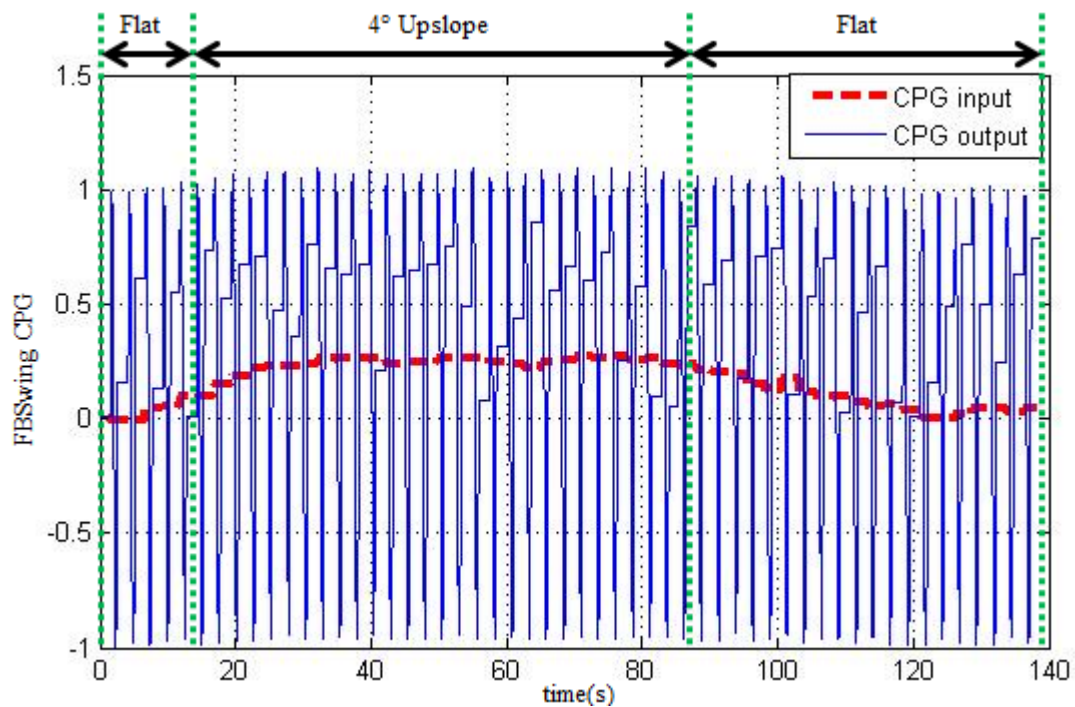


圖 5.44 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

如圖 5.46 顯示，對於行走時所偵測到在 X 方向的旋轉角度 $AngX$ ，此感測值 $AngX$ 隨上半身的左右擺盪呈現很規律的左右擺盪，由於抬腳時的重心影響和我們設計即時補償到關節 Hip-Roll 的即時補償控制器，即是希望當抬腳成功之後，重心不會太往支撐腳方向偏，但顯然在真實環境中我們設定左右搖擺的曲線為在 $\pm 9.5^\circ$ 之間，不過感測值顯示基本上都超出了 1 或 2 度，表示若是缺少了此即時

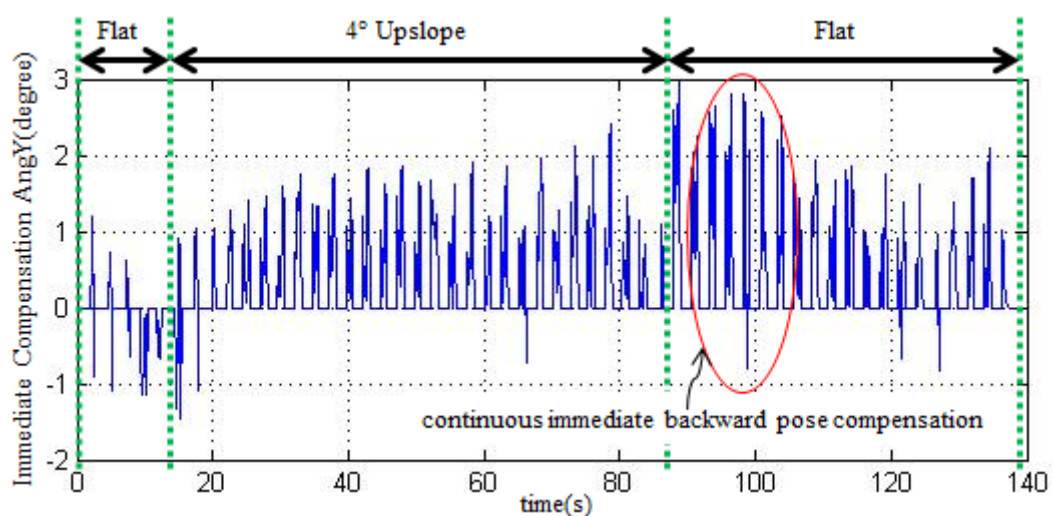


圖 5.45 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

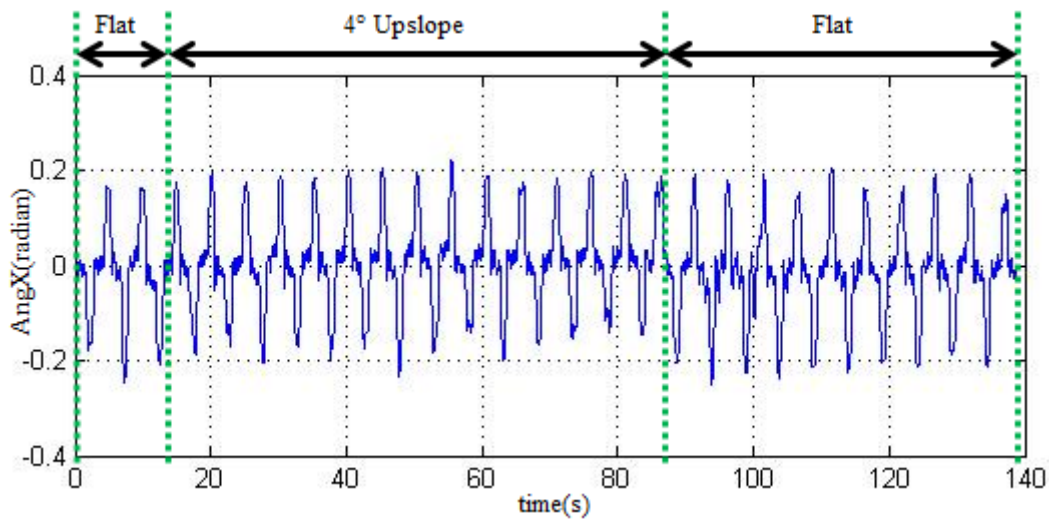


圖 5.46 行走過程中之 X 方向旋轉角度曲線

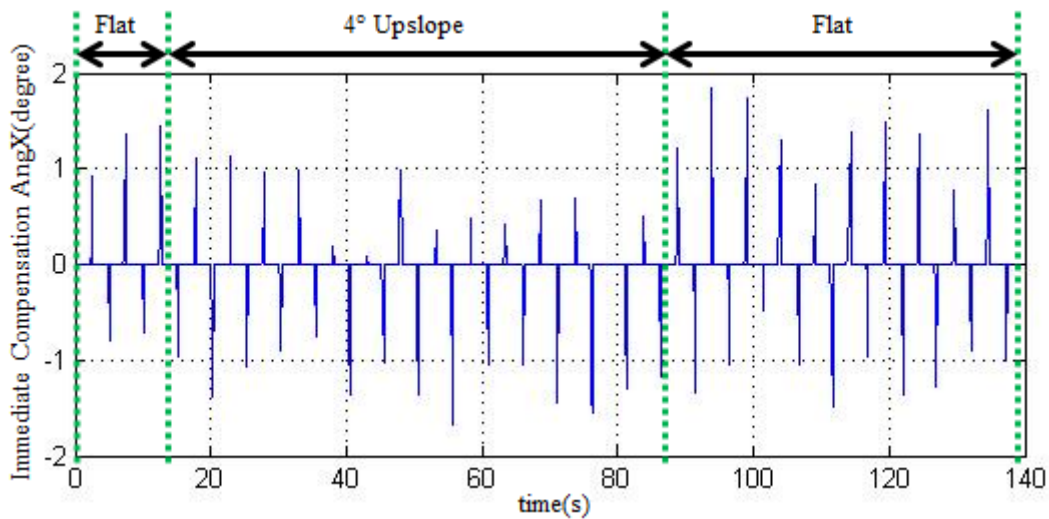


圖 5.47 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

補償控制器可能就會發生往支撐腳方向傾倒的情況發生，但由於此控制器導致機器人可以保持平穩的行走。其中對於即時補償到關節 Hip-Roll 的即時補償控制值，如圖 5.47 所示，且由圖 5.41 中可以看出機器人走在上坡後段時，開始較明顯的往右偏，因此上半身重力向後分力會有部分轉換為向右分力，讓機器人的左右擺盪因重心的關係呈現出不平衡，從圖 5.47 中我們可以看出在上坡後段對於向左擺盪的縮減補償變少，向右擺盪的縮減補償明顯的增加，來達到左右擺盪幅度的平衡保持，從圖 5.46 中可看出在上坡後段上半身的左右搖擺並不受重心影響。

5.3.3.2 平地→4°下坡→平地之行走實驗結果

平地→4°下坡→平地之行走實驗過程如圖 5.48(a)~(o)所示。

此時對於地形偵測的情況與補償控制器的作用效果說明如下：

如圖 5.49 顯示，行走時所偵測到在 Y 方向的旋轉角度 $AngY$ ，此感測值 $AngY$ 隨上半身的前後擺盪呈現擺盪的曲線，從整體變化上來看，在下坡路段上半身與走在平地的姿態沒有多大的改變。在最後的平地路段中，也是處於平穩的狀態，只有在從平地走到下坡時的瞬間，上半身產生忽然的前傾使得機器人忽然不穩定(圖 5.49 的紅圈處)，而此時透過即時補償控制器中，即時補償在關節 Hip-Pitch 的角度補償控制器，讓上半身重心有即時很大的往後(正值)旋轉傾斜(圖 5.52 的紅圈處)，使得上半身即時的修正回來避免跌倒，致使機器人行走能回歸平穩。

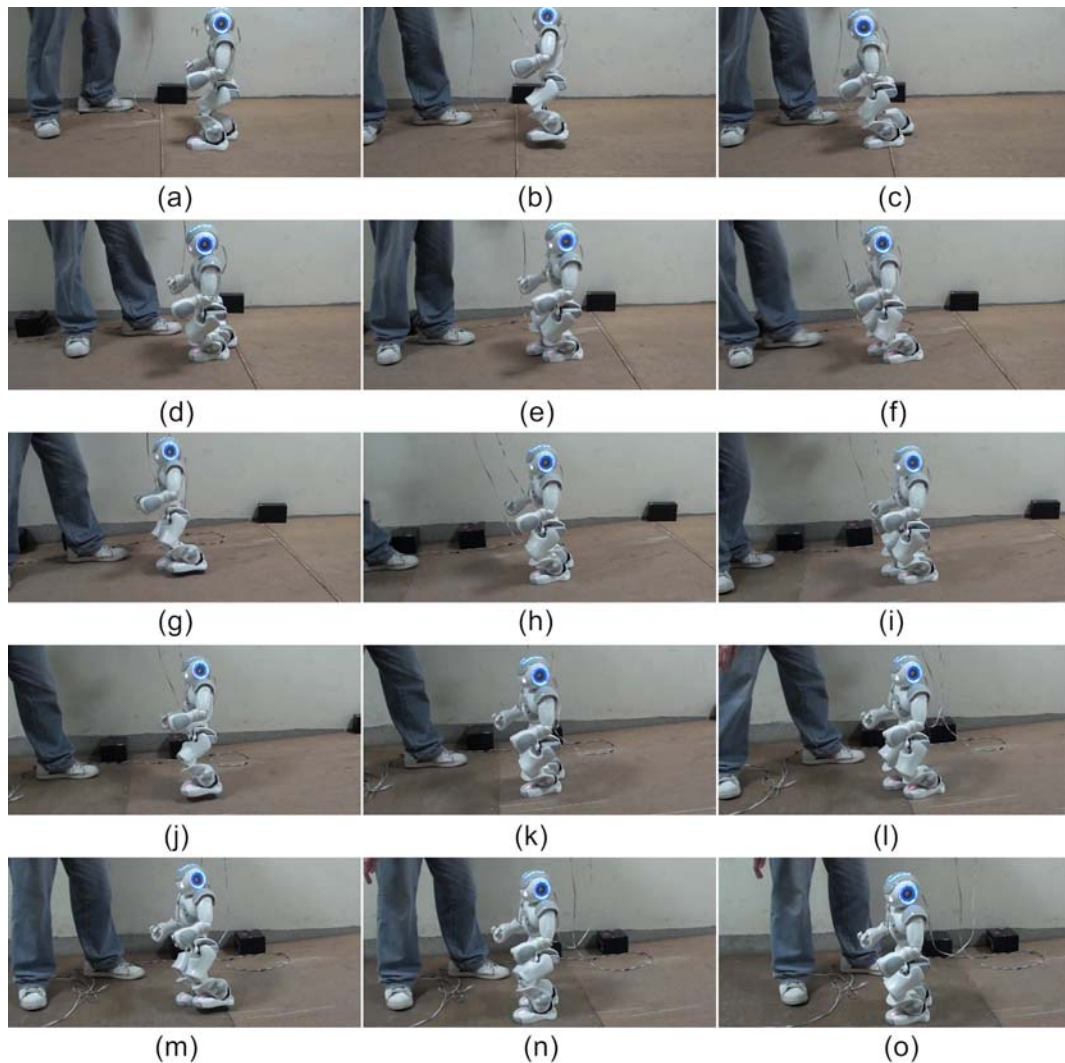


圖 5.48 平地→4°下坡→平地之行走實驗過程

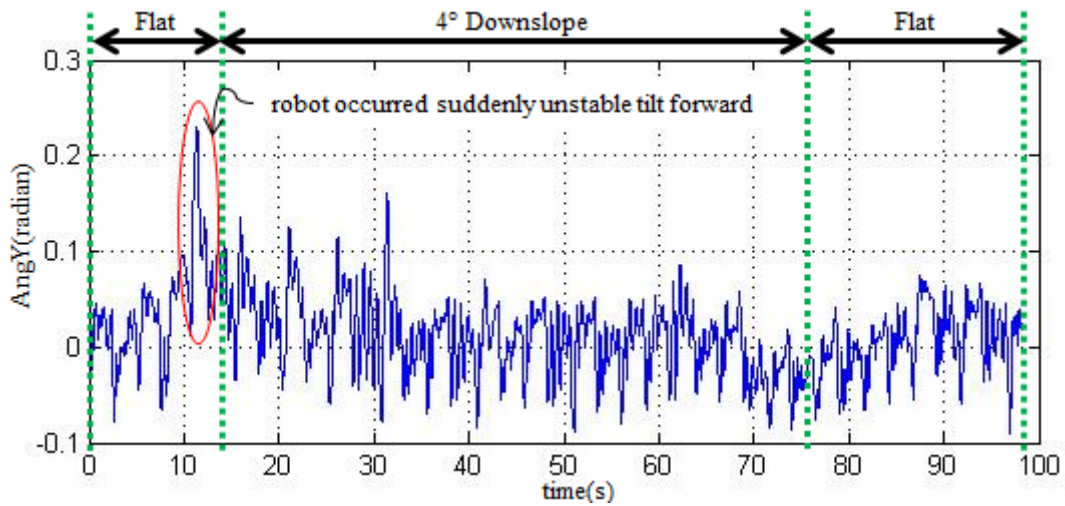


圖 5.49 行走過程中之 Y 方向旋轉角度曲線

預測補償控制器中，首先，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.50 所示，從圖中可以明顯的看出當機器人整個走下 4° 下坡之後，感測出要補償的值迅速的到達與理想值 -4° 只差了 ±1 點多度，補償值處於在 -4.9 度附近保持穩定；而當再度走到平地時，與 5.3.3.1 節的上坡實驗相比，算是較迅速的昇到 -1 點多度，且在幾步之後在 0 度附近保持穩定。接著，影響機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變的影響，使得上半身的前後擺盪在走到下坡有多後傾(負值變大)少前傾(正值變小)，如圖 5.51 所示。

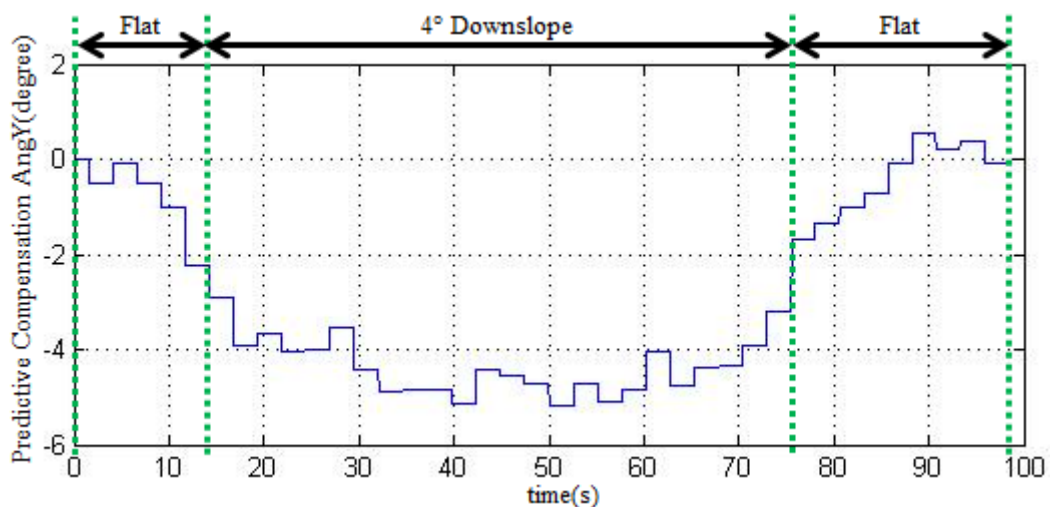


圖 5.50 直接補償於關節 Hip-Pitch 之預測補償角度值

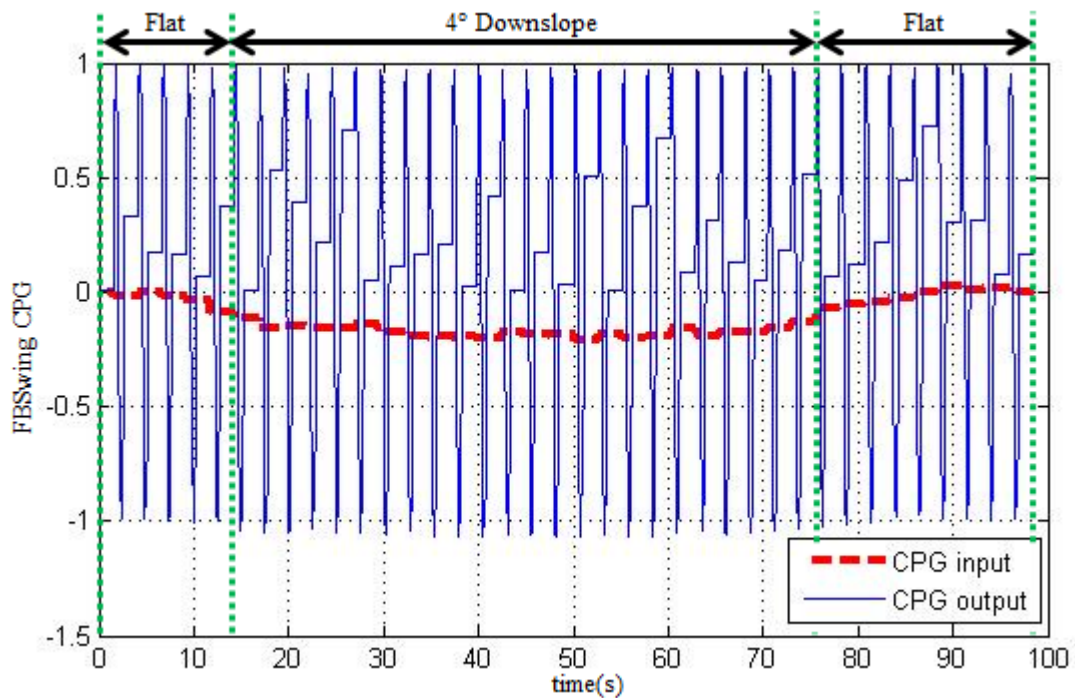


圖 5.51 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

在即時補償控制器中，對於即時補償到關節 Hip-Pitch 的角度補償值，如圖 5.52 所示，在此圖可以明顯地顯示出當走到下坡時，即時補償值有持續的將上半身重心往後(正值 1.4 度)旋轉傾斜，使得整體補償理想值接近於 -6.5 度，但我們認為是下坡路段上半身重心有往前傾的狀況，讓感測資訊 AngY 仍然與走在平地的資訊差不多；當再度走到平地時，預測補償值在 -1 點多度時，即時補償值有將上半身重心往前(負值)旋轉傾斜，而預測補償值在正 0 度附近振盪時，即時補償

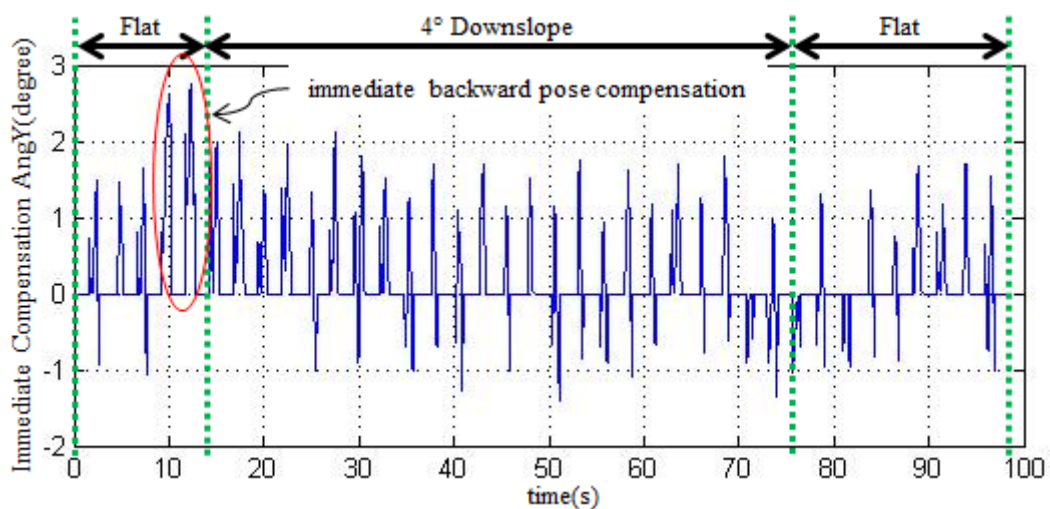


圖 5.52 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

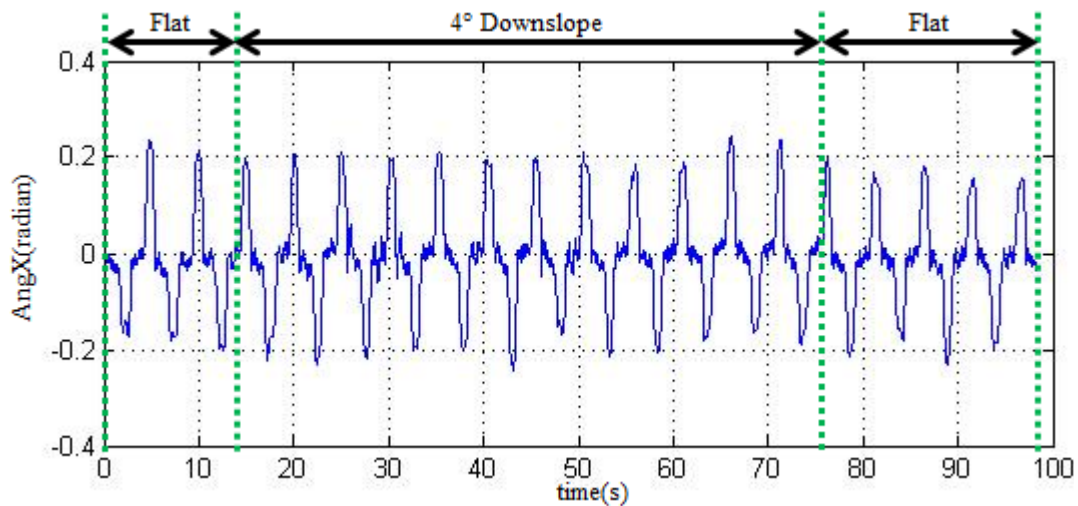


圖 5.53 行走過程中之 X 方向旋轉角度曲線

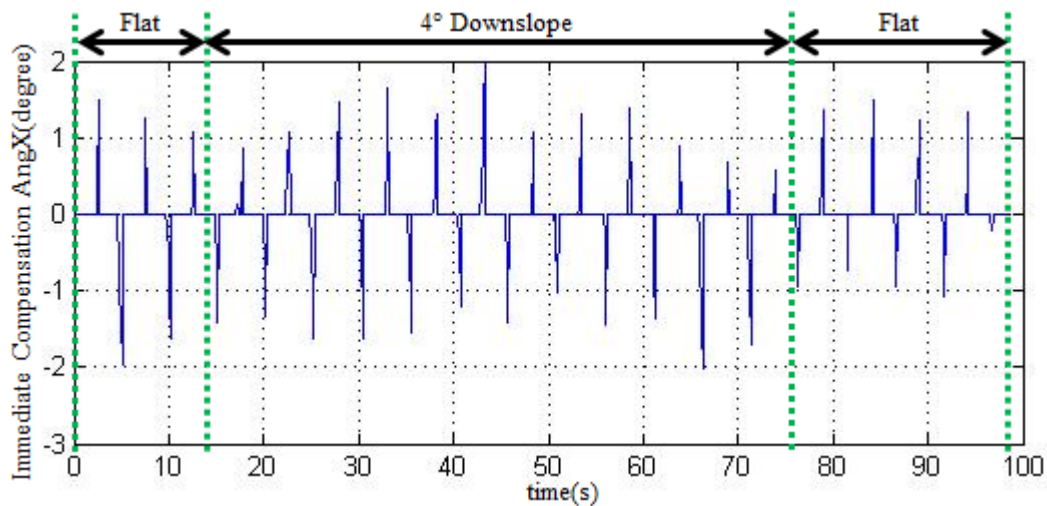


圖 5.54 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

值有將上半身重心往後(正值)旋轉傾斜，藉此達到遵守行走曲線的行走。

如圖 5.53 顯示，對於行走時偵測到在 X 方向的旋轉角度 $AngX$ ，此感測值 $AngX$ 隨上半身的左右擺盪呈現很規律的左右擺盪，但由圖 5.48 看出在下坡後段機器人行走向左偏，因此 $AngX$ 的曲線有向上移(向右擺盪較多)的現象。圖 5.54 為即時補償到關節 Hip-Roll 的即時補償控制值的曲線圖。因為行走向左偏，上半身重力向前分力會有部分轉換為向右分力，讓機器人的左右擺盪因重心的關係呈現出不平衡，從圖 5.54 中我們可以看出在上坡後段對於向左擺盪的縮減補償變少，向右擺盪的縮減補償明顯的增加，來達到左右擺盪幅度的平衡保持。

5.3.3.3 平地→7°上坡之行走實驗結果

在平地→7°上坡行走時地形偵測的情況與補償控制器的作用效果說明如下：

如圖 5.55 顯示，行走時所偵測到在 Y 方向的旋轉角度 $AngY$ ，此感測值 $AngY$ 隨上半身的前後擺盪呈現擺盪的曲線，當走到上坡的一開始時，上半身在前兩步時仍受到地形變化往後傾斜(圖 5.55 的紅圈處)，但在後面路段上半身就與走在平地的姿態(上半身保持與平地為垂直姿態的振盪)沒有多大的改變，而此時透過即時補償控制器中，即時補償在關節 Hip-Pitch 的角度補償控制器，讓上半身重心有即時很大的往前(負值)旋轉傾斜(圖 5.58 的紅圈處)，使得上半身即時的修正回來避免跌倒，致使機器人行走能平穩。

預測補償控制器中，首先，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.56 所示，從圖中可以看出當機器人走到 7° 上坡時，感測出要補償的值在五步之後才到達理想值 7°，此時尚未滿足預測補償理想值 7° 的上半身傾斜，從圖 5.58 可看出，已經利用了即時補償的往前傾來進行修正，使機器人平衡，到五步之後預測補償值皆處於在 7 度附近保持穩定，符合了預測補償值所需補償的理想值 7°。接著，影響機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變的影響，使得上半身的前後擺盪在走到上坡有多前傾(正值變大)少後傾(負值變小)，如圖 5.57 所示。

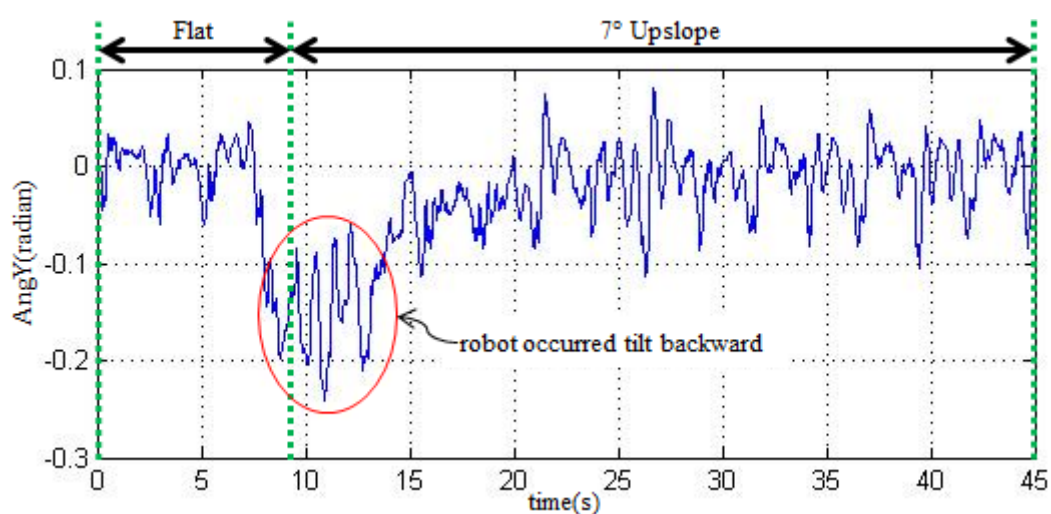


圖 5.55 行走過程中之 Y 方向旋轉角度曲線

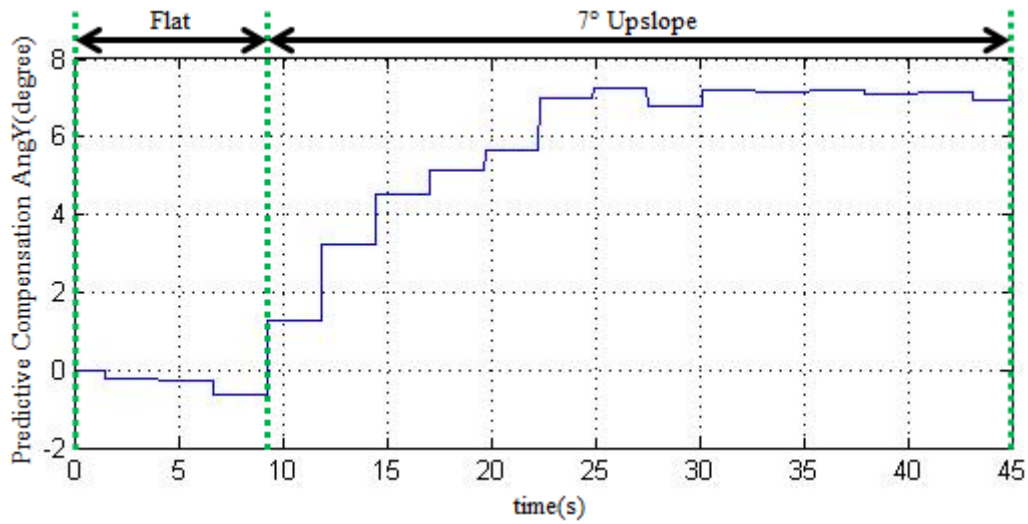


圖 5.56 直接補償於關節 Hip-Pitch 之預測補償角度值

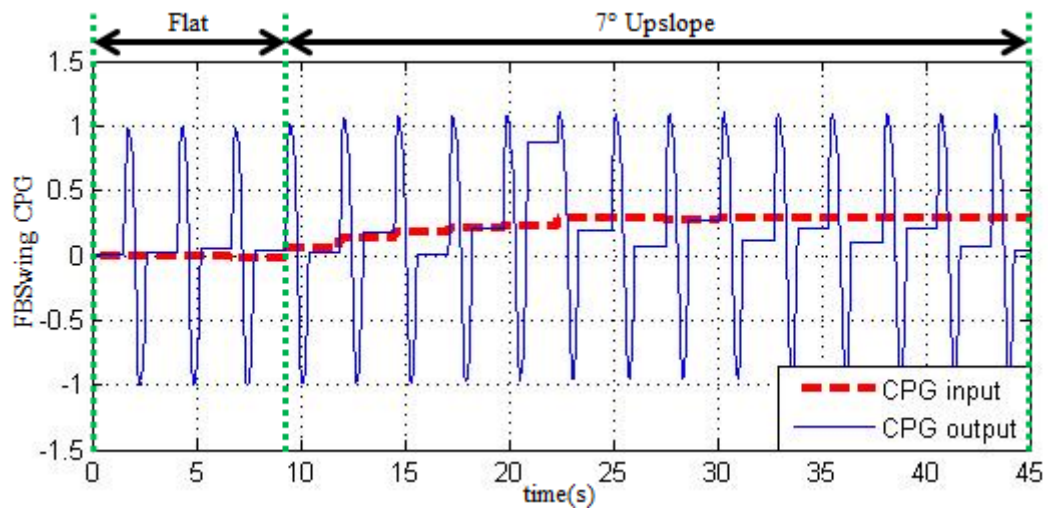


圖 5.57 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

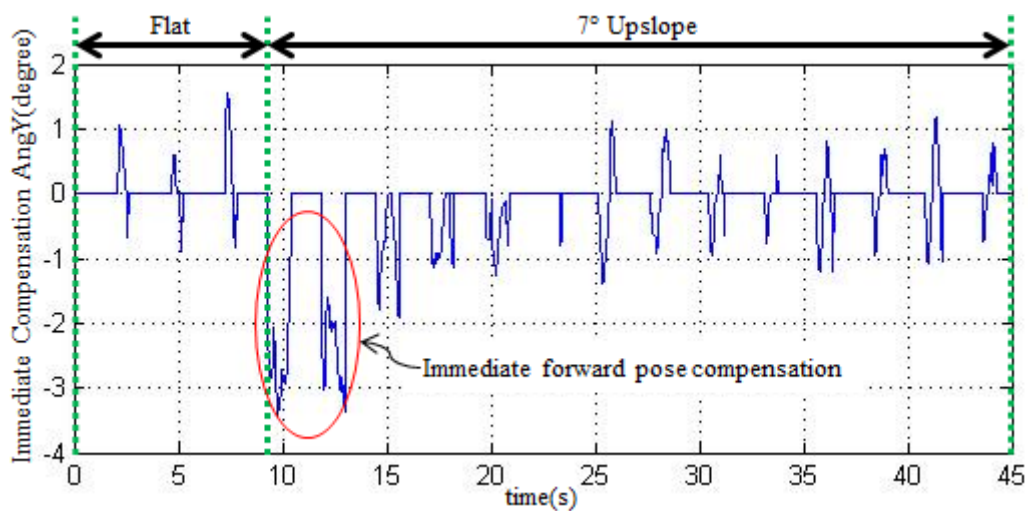


圖 5.58 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

在即時補償控制器中，對於即時補償到關節 Hip-Pitch 的角度補償值，如圖 5.58 所示，在此圖可以明顯地顯示出當走到上坡的一開始，預測補償值尚未計算出滿足地形的斜坡角度(如圖 5.56 所示)時，即時補償值產生較大的負值將上半身重心往前旋轉傾斜，使得整體補償值能變大較符合理想值，增加行走的平穩，而在最後當預測補償值計算出之值符合理想值時，即時補償值的補償角度也是恢復成與平地時的即時補償相同，只有進行些微的上半身前後旋轉的角度修正，讓機器人可以符合所需的上半身姿態使得行走的更加平穩。

對於行走時偵測到在 X 方向的旋轉角度 AngX，如圖 5.59 顯示，此感測值

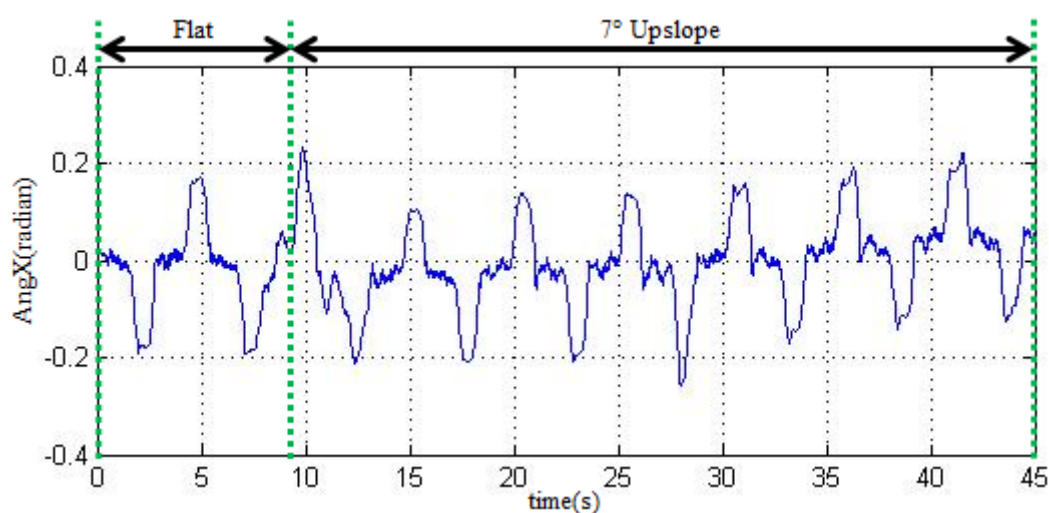


圖 5.59 行走過程中之 X 方向旋轉角度曲線

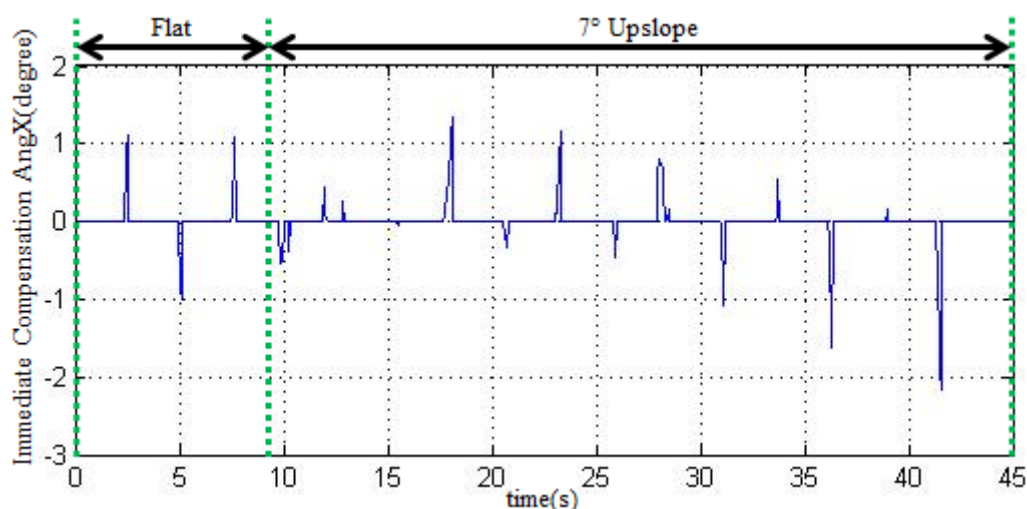


圖 5.60 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

AngX 隨上半身的左右擺盪呈現規律的左右擺盪，但由圖 5.59 看出在上坡後段機器人行走向右偏，因此 AngX 的曲線有向上移(向右擺盪較多)的現象。圖 5.60 為即時補償到關節 Hip-Roll 的即時補償控制值的曲線圖。因為行走向右偏，上半身重力向後分力會有部分轉換為向右分力，讓機器人的左右擺盪因重心的關係呈現出不平衡，從圖 5.60 中我們可以看出在上坡後段對於向左擺盪的縮減補償(正值)變少，向右擺盪的縮減補償(負值)明顯的增加，來達到左右擺盪幅度的平衡保持。

5.3.3.4 7°下坡→平地之行走實驗結果

對於 7°下坡→平地行走時，地形偵測的情況與補償控制器的作用效果說明如下：

如圖 5.61 顯示，行走時所偵測到在 Y 方向的旋轉角度 AngY，此感測值 AngY 隨上半身的前後擺盪呈現擺盪的曲線，由於機器人是放在 7°下坡上直接開始進行行走，因此一開始上半身的姿態為絕對於平地的垂直姿態，所以對於 7°下坡為往前傾 7°(正值)，隨著控制補償的調整才恢復成行走於平地的姿態，在 0 附近進行前後擺盪。

預測補償控制器中，首先，對於直接補償到關節 Hip-Pitch 的角度補償值在每一步上的改變，如圖 5.62 所示，從圖中可以明顯的看出當機器人走在 7°下坡時，第一次感測出要補償的值只到-2 點多度，會發生此原因為為了減少行走到後期機器人的擾動影響到補償值偏差太大，因此我們在計算預測補償值上進行一比例的縮減，而此舉動也影響到一開始對於地形變化的立即預測補償的效能，但是透過與即時補償的搭配，也讓機器人達繼續達到平穩的行走，並且在四步之後補償值也在-7°附近保持穩定；而當再走到平地時，預測補償值只有昇到-2 點多度保持穩定，但是透過與即時補償的搭配，讓機器人能持續平穩的行走。接著，影響機器人上半身前後擺盪的 CPG 輸出曲線受到輸入曲線改變影響，使得上半身前後擺盪在走下坡時多後傾(負值變大)少前傾(正值變小)，如圖 5.63 所示。

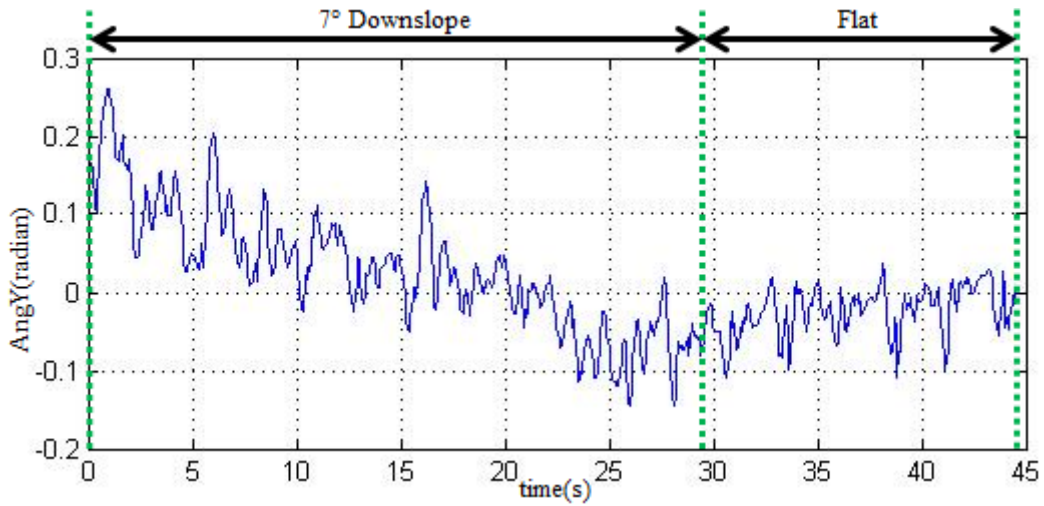


圖 5.61 行走過程中之 Y 方向旋轉角度曲線

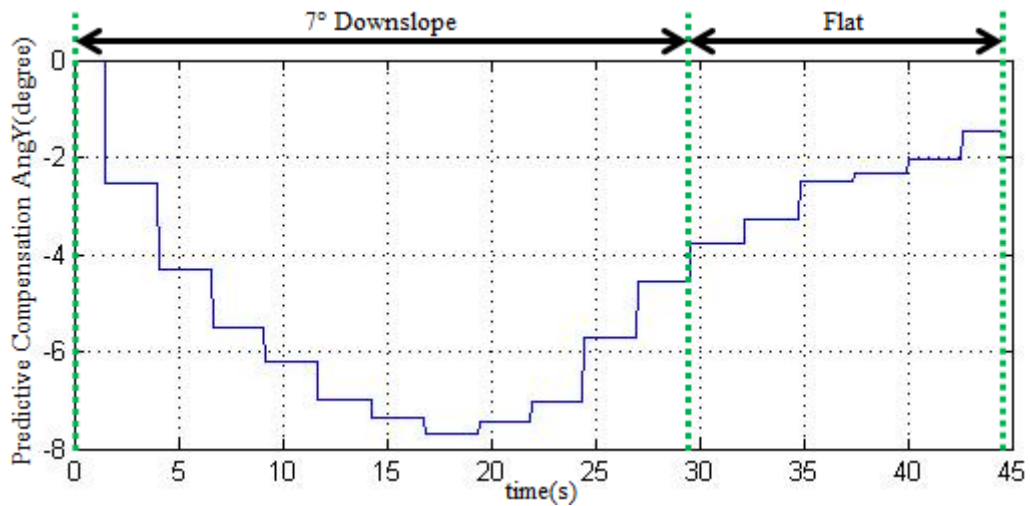


圖 5.62 直接補償於關節 Hip-Pitch 之預測補償角度值

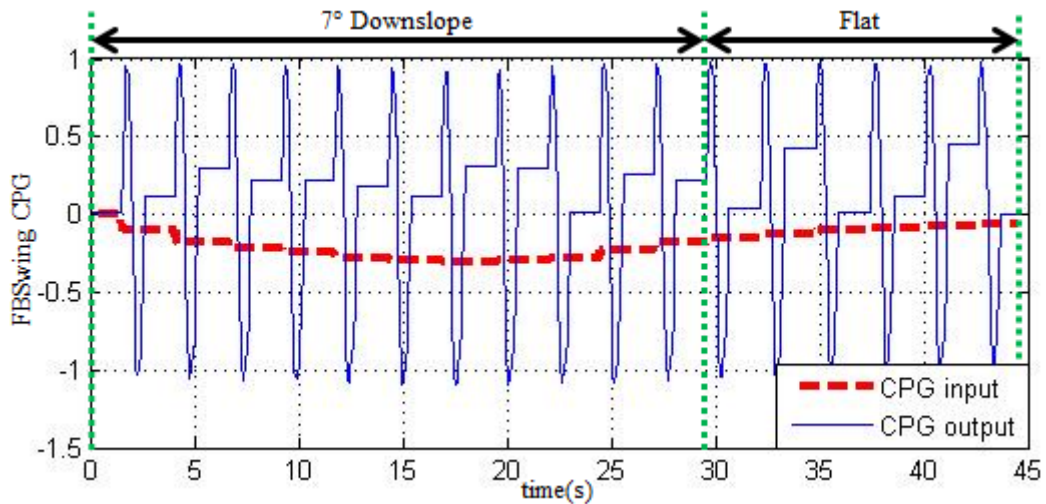


圖 5.63 控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係

在即時補償控制器中，對於即時補償到關節 Hip-Pitch 的角度補償值，如圖 5.64 所示，在此圖可以明顯地看出走在下坡一開始，預測補償值尚未計算出滿足地形的斜坡角度(如圖 5.62 所示)時，即時補償值產生很大的正值將上半身重心往後旋轉傾斜，使得整體補償值能變大使得較符合理想值，增加行走的平穩，而在之後走到平地時，當預測補償值計算出之值小於行走平地的理想值時，即時補償值也在每一步的一開始產生負值將上半身重心往前旋轉傾斜，使得整體補償值能變大使得較符合理想值，之後在前後的修正來調整上半身姿態。

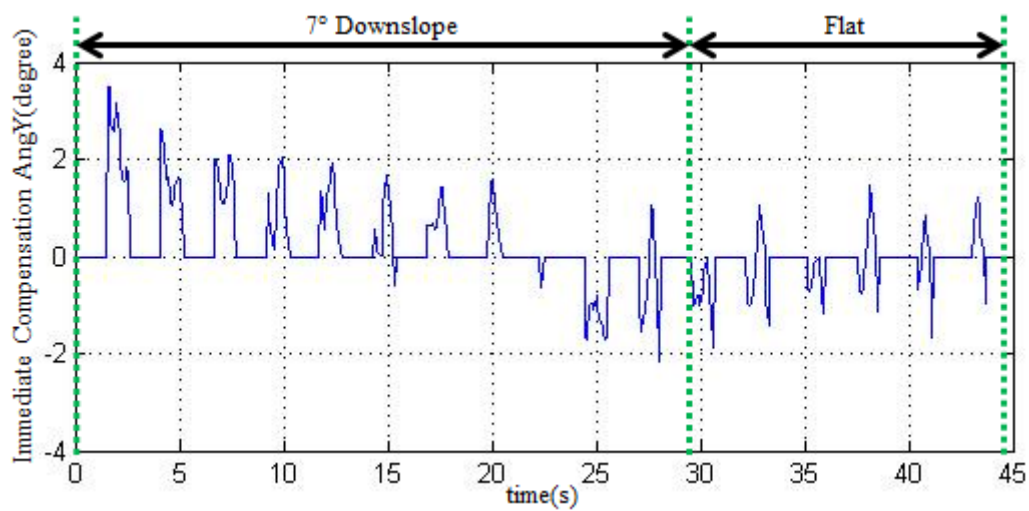


圖 5.64 單腳支撐相時補償於關節 Hip-Pitch 之即時補償角度值

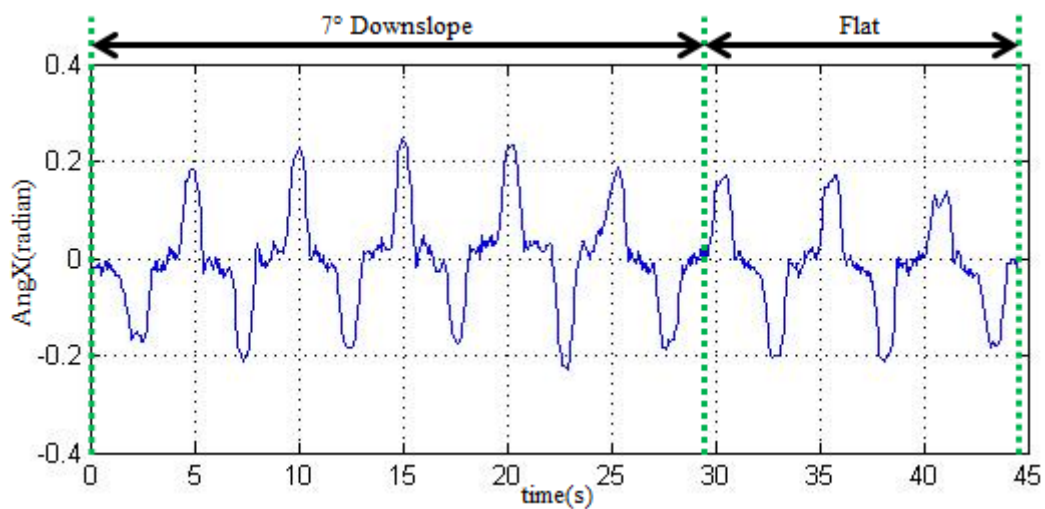


圖 5.65 行走過程中之 X 方向旋轉角度曲線

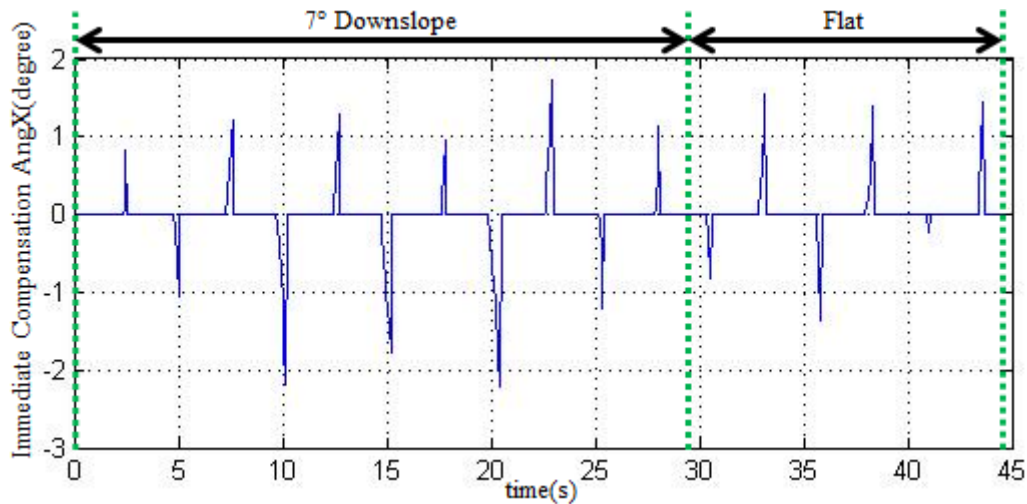


圖 5.66 單腳支撐相時補償於關節 Hip-Roll 之即時補償角度值

如圖 5.65 顯示，對於行走時偵測到在 X 方向的旋轉角度 AngX，此感測值 AngX 隨上半身的左右擺盪呈現很規律的左右擺盪，但從圖中看出在下坡中段機器人行走向左偏，因此 AngX 的曲線有向上移(向右擺盪較多)的現象。圖 5.66 為即時補償到關節 Hip-Roll 的即時補償控制值的曲線圖。因為行走向左偏，上半身重力向前分力會有部分轉換為向右分力，讓機器人的左右擺盪因重心的關係呈現出不平衡，從圖 5.66 中我們可以看出在上坡後段對於向左擺盪的縮減補償變少，向右擺盪的縮減補償明顯的增加，來達到左右擺盪幅度的平衡保持。

5.3.4 實驗結果之討論

由於真實 NAO 行走時，我們是透過 NAO 的內建運算法達到對於每個關節角度在一段時間內，點到點的位置控制，因此當行走時會因馬達轉動時的停止啟動和加減速產生震動，讓機器人整體會有抖動產生，在此論文中只有盡量的透過對於運算法的參數調整來降低此狀況，並無法完全的杜絕。

5.4 討論

對於行走步態在 DSP 和 SSP 之間的切換不會有交疊的情況產生，為了防止交疊的情況，我們在 DSP 和 SSP 轉換之間的控制機制為，若在計算中要到達的控制位置超過設定值時，我們會進行位置移動的限制，使得步態繼續保持理想，

因此整個週期的時間會比理想值有非常少許的增加，但不影響整體的行走步態。

對於行走步態的設計，由於我們所設計的平衡補償的控制方法為即時的修正，且無針對不同的步態進行補償控制的學習或訓練，因此對於不同步態的應用，理想中能得到符合預期的補償控制效果。但實際上若是改變其行走速度仍然會有所缺點產生，當考慮到機器人馬達運行限制及重心移動慣性力影響之下，首先對於雙腳支撐相的時間縮短，我們已從實驗的測試得知當速度快時會大幅的影響感測資訊的精確度，因此在實驗中我們增加雙腳支撐相的時間來增加感測資訊的精確度，另外對於單腳支撐相的時間縮短上，由於我們希望是利用較大的抬腳高度與步伐來進行實驗，因此在相同的步伐下當把時間縮短，機器人在移動時會因為各個馬達本身快速轉動產生出的抖動，導致機器人會有持續自我擾動的情況產生，並且由於速度的加快機器人移動時的慣性力增大，也讓機器人會有不穩的情況產生，因此在相同的行走參數中，只減少單腳支撐相時間要達成跨步是困難的，必須相對應的調整上半身擺盪的控制及步伐長度及高度上的縮減，才使得機器人可以在行走速度增快的情況下，達到平衡穩定的平地行走並且讓感測器提供準確的資訊使得補償控制器能加以應用，達到機器人能在未知斜面上進行行走的控制。

透過模擬和實驗，從控制上半身前後擺盪的 CPG 的輸入與輸出之間的變化關係圖得知，基於 CPG 振盪器所設計的步態規劃，在雙腳支撐相時，上半身的擺盪角度值不一定為理想值 0，且左右腳在 Z 方向的高度也不一定為等高，此情況在模擬中因程式計算較快差距甚小，對於整體行走幾乎不受影響。在實驗中則程式計算上較慢，會有較大的差距產生(如圖 5.28、5.35 所示)，但由於雙腳支撐相的設計產生較穩定的區間，因此仍然可以保持平穩的行走，對於預測補償控制器的偵測，由於我們有將此偏差值考慮進去，因此在補償控制上也將不受影響。

在模擬或實驗中，感測資訊在整體的曲線移動上皆有符合我們預期和控制器所產生之效果，但仍然會有一些不符合預期的突波出現，我們認為是忽然不穩定的擾動震盪產生或補償控制過多的補償產生，但由於補償控制器會持續的進行平衡控制，因此這些突波並不會影響整體的平穩行走。

首先，由 5.3.1 節中三個模擬得知，利用補償控制器行走的機器人，在對於較不考慮衝擊力影響的模擬環境下，能成功的在“平地→上坡→平地→下坡→平地”之間行走，且此時對於預測和即時補償控制器皆有顯著的好效果產生。

接著，由 5.3.2 節中四個實驗得知，沒有利用補償控制器行走皆比有利用補償控制器行走使得 ZMP 更快的離開穩定區間，上半身傾斜而跌倒，且當測試的斜坡愈斜時，愈快發生 ZMP 離開穩定區間，上半身傾斜而跌倒，由此可證明當有利用補償控制器，讓上半身進行姿態調整控制，行走是可以更穩定的；另外，從實驗中的上半身姿態感測值比較可以看出，有利用補償控制器的上半身與無補償控制的上半身相比皆有明顯的在較靠近 0 的區間進行晃動，即是在較穩定的區間進行晃動。

最後，由 5.3.3 節中四個實驗得知，在利用了補償控制器行走的機器人，對於衝擊力影響較大的實際環境下，當補償控制時間夠快時，機器人仍然可以在環境中進行長時間的平穩行走，且預測和即時補償控制器仍然有好的效果產生。

所以對於預測補償控制器，對於下一步所面臨的地形，針對上半身的姿態調整成與在平地時行走的姿態相同，以及即時補償控制器，對於單腳支撐相時，上半身在前後及左右方向因應平穩條件的即時補償控制，都因而得到對於平穩行走有很好的控制效果之證明。

第六章 結論與未來展望

6.1 結論

本論文發展出使雙足機器人能自主偵測並平穩進行行走於不同坡度的設計方法。在本論文中，首先對 Central Pattern Generator(CPG)振盪原理與特性進行探討，使得我們可以基於 CPG 之振盪原理提出一種步態規劃，使機器人得以產生連續的行走步態，在平地上進行平穩的行走。接著，在對於地形環境在斜度上發生改變時，本論文著重於藉由上半身姿態之傾斜角度的調整，達成持續平穩行走的控制器設計。在此設計中，我們利用了上半身在 X、Y 方向的旋轉角度感測值，使機器人得知目前相對於地形情況的上半身姿態角度值，並且利用預測補償控制器先進行下一步為支撐腳時的補償控制，而此預測補償控制除了直接上半身的角度補償控制之外，同時也透過 CPG 的特性，對上半身前後擺盪的曲線進行修正，使得擺盪的效果能讓機器人更加穩定；並在單腳跨步時，再使用即時補償控制器針對在單腳支撐相時的不穩定因素進行上半身的直接角度補償修正。藉此完成整體的平衡行走控制系統，使機器人能平穩地行走在不同的地形坡度之間。

此平衡行走控制系統是透過商用機器人 NAO 得到成功的驗證，模擬和實驗的驗證結果顯示，本論文所提出的行走步態是可行的，且整體的補償控制器皆能達到我們在設計理念中所要達成之效果。模擬結果顯示，在不改變參數之下，機器人對於不同角度的坡度達成的上下坡行走，顯示出我們的控制方法不會侷限在單一角度上。另外，在真實環境下實驗顯示，雖然有較多的雜訊，但透過對步態和控制參數的調整，仍然可以在平地與上下坡之間達成穩定的行走，藉此證明我們的控制方法能實現在真實機器人行走在環境中，產生即時的控制效果。而從模擬和實驗中也驗證出，此行走步態和補償控制器的應用對於未知斜面的行走，NAO 至少可以在 $-7^{\circ}\sim 7^{\circ}$ 之間進行行走，當然若用在不同機器人上，角度範圍限制也會不相同，所以此套平衡行走方法能讓一般化的機器人進行使用，無須利用到特製化機構即可達成。

6.2 未來展望

在未來研究展望方面，首先是對於補償控制時影響到感測資訊的變化，我們希望建立出一套互相作用的關係，使得補償控制可以進一步利用感測輸入影響 CPG 輸出的關係，達到直接修正步態規劃曲線來完成行走的平穩控制；或是透過補償控制對感測資訊的變化，修改跨步參數的調整，調整跨步的高度和長度，以及修改控制上半身擺盪的參數，調整上半身擺盪的曲線，使得行走的步態及擺盪曲線更符合地形的變化，而並非全用 CPG 的參數調整來達到，但目的皆是希望可以直接改善行走曲線，而非加入額外的補償控制器來達到平穩控制。

另外，本論文在感測資訊使用方面，只使用上半身的姿態資訊達成平衡控制，無法偵測到當擺盪腳落地時所產生使機器人不穩定的衝擊力，而此問題對於本論文也是一大變因，在本論文中目前是以增長雙腳支撐相的時間來抵銷此作用力，但未來仍希望透過對於腳底壓力感測器的感測資訊，建立一套對於衝擊力的控制方法，消除此不穩定因素，使得機器人可以行走的更平穩且行走週期更為加快。

参考文献

- [1] J. Duysens, Van de Crommert and Henry. WAA., “Neural Control of Locomotion. Part 1: the Central Pattern Generator from Cats to Humans,” *Gait and Posture*, Vol.7, No.3, pp.131–141, 1998.
- [2] A. J. Ijspeert, “Central Pattern Generators for Locomotion Control in Animals and Robots: a Review,” *Neural Networks*, Vol.21, pp.642-653, 2008.
- [3] M. Wang, L. Sun, P. Yuan and Q. Meng, “Periodicity Locomotion Control Based on Central Pattern Generator,” in *Proc. of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 21-23, 2006, pp.3144-3148.
- [4] Larsen, Jorgen Christian, “Central Pattern Generators in Modern Science.”
- [5] L. Righetti and A. J. Ijspeert, “Programmable Central Pattern Generators: an Application to Biped Locomotion Control,” in *Proc. of IEEE International Conference on Robotics and Automation*, Orlando, Florida, May, 2006, pp.1585-1590.
- [6] L. Righetti and A. J. Ijspeert, “Pattern Generators with Sensory Feedback for the Control of Quadruped Locomotion,” in *Proc. of IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, May 19-23, 2008, pp.819-824.
- [7] K. Watanabe, G. L. Liu, M. K. Habib and K. Izumi, “Neural Oscillators with a Sigmoidal Function for the CPG of Biped Robot Walking,” in *the 33rd Annual Conference of the IEEE Industrial Electronics Society(IECON)*, Taipei, Taiwan, Nov. 5-8, 2007, pp.128-133.
- [8] K. Matsuoka, “Sustained Oscillations Generated by Mutually Inhibiting Neurons with Adaptation,” *Biological Cybernetics*, Vol. 52, pp. 367–376, 1985.
- [9] K. Matsuoka, “Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators,” *Biological Cybernetics*, Vol. 56, pp. 345–353, 1987.
- [10] M. Williamson, “Neural Control of Rhythmic Arm Movements,” *Neural Networks*, Vol. 11, No. 7-8, pp. 1379-1394, 1998.
- [11] Y. Fukuoka, H. Kimura and Avis H. Cohen, “Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts,” *The International Journal of Robotics Research*, Vol. 22, No. 3-4, pp. 187-202, 2003.
- [12] S. J. Yi, B. T. Zhang and Daniel D. Lee, “Online Learning of Uneven Terrain for Humanoid Bipedal Walking,” in *Proc. of International Conference on Association for the Advancement of Artificial Intelligence*, Atlanta, Georgia, USA, pp. 1639-1644, 2010.
- [13] Miomir Vukobratovi'c and Branislav Borovac, “Zero-Moment Point — Thirty Five Years Of Its Life,” *International Journal of Humanoid Robotics*, 2004, Vol. 1, No. 1, pp. 157-173.

- [14] F. Ali, A. C. Amran and A. Kawamura, "Bipedal Robot Walking Strategy on Inclined Surfaces using Position and Orientation based Inverse Kinematics Algorithm," in *Proc. of IEEE International Conference on Control, Automation, Robotics and Vision*, Singapore, December 7-10, 2010, pp.181-186.
- [15] M. Ogino, H. Toyama and M. Asada, "Stabilizing Biped Walking on Rough Terrain based on the Compliance Control," in *Proc. of IEEE/RSJ International Conference on Intelligent Robot and Systems*, San Diego, CA, USA, Oct 29- Nov 2, 2007, pp.4047-4052.
- [16] Jian Li and Weidong Chen, "Modeling and Control for a Biped Robot on Uneven Surfaces," in *Proc. of IEEE International Conference on Decision and Control*, Shanghai, China, 2009, pp.2960-2965.
- [17] H. Inada and K. Ishii, "Behavior Generation of Bipedal Robot Using Central Pattern Generator(CPG)(1st Report: CPG Parameters Searching Method by Genetic Algorithm)," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October. 2003, Las Vegas, USA, pp. 2179–2184.
- [18] G. Endo, J. Nakanishi, J. Morimoto, and G. Cheng, "Experimental Studies of a Neural Oscillator for Biped Locomotion with QRIO," in *IEEE International Conference on Robotics and Automation*, April. 2005, Barcelona, Spain, pp. 596–602.
- [19] V. H. Dau, C. M. Chew and A. N. Poo, "Planning Bipedal Walking Gait Using Augmented Linear Inverted Pendulum Model," in *IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, 2010, pp. 575–580.
- [20] Mei Shuai, Chenglong Fu and Ken Chen "A Real-time Control Method for Humanoid Robot to Walk Stably on Uneven Ground," in *Proc. of IEEE Conference on Robotics, Automation and Mechatronics*, Beijing, China, 2006, pp.1~6.
- [21] Barkan Ugurlu and Atsuo Kawamura, "Eulerian ZMP Resolution based Bipedal Walking: Discussions on the Intrinsic Angular Momentum Rate Change about Center of Mass," in *Proc. of IEEE International Conference on Robotics and Automation*, Alaska, USA, 2010, pp. 4218~4223.
- [22] 許煥坤, "雙足機器人之步態補償控制設計," 國立交通大學電機與控制工程學系碩士論文, 2008.
- [23] Krister Wolff, Jimmy Pettersson, Almir Heralic' and Mattias Wahde, "Structural Evolution of Central Pattern Generators for Bipedal Walking in 3D Simulation," in *Proc, of IEEE International Conference on Systems, Man, and Cybernetics*, Taipei, Taiwan, 2006, pp.227-234.

- [24] N. Sadati and K. A. Hamed, "Neural Control of a Fully Actuated Biped Robot," in *Proc. of IEEE International Conference on Robotics and Biomimetics*, December 17-20. 2006, Kunming, China, pp. 1299–1304.
- [25] H. Kimura, Y. Fukuoka and A. H. Cohen, "Biologically Inspired Adaptive Dynamic Walking of a Quadruped Robot," in *8th International Conference on the Simulation of Adaptive Behavior*, Jul. 2004, LA, USA, pp. 201–210.
- [26] G. Endo, J. Nakanishi, J. Morimoto and G. Cheng, "An Empirical Exploration of a Neural Oscillator for Biped Locomotion Control," in *IEEE International Conference on Robotics and Automation*, April. 2004, New Orleans, LA, USA, pp. 3036–3042.
- [27] G. Endo, J. Nakanishi, T. Matsubara, J. Morimoto and G. Cheng, "Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 213-228, 2008.
- [28] Yuta Hoshino, Chenglong Fu and Ken Chen, "A Passive Walking Strategy for a Biped Robot with a Large Mass Torso by a Spring and a Damper," in *Proc. of IEEE Conference on Mechatronics and Automation*, Beijing, China, 2011, pp.1269-1274.
- [29] NAO Website: <http://www.aldebaran-robotics.com/>
- [30] F.J.O. Corpuz, B.C.Y. Lafoteza, R.A.L. Broas and M. Ramos Jr, "Design and Implementation of a Closed-Loop Static Balance System for the YICAL Leg 2 Biped," in *IEEE TENCON*, 2009, pp. 1-6.
- [31] Y. Gao, X. Ruan, H. Song and J. Chen, "Research on Filtering Prople in Inertial Sensors for a Two-Wheeled Self-Balanced Robot," *Chinese Journal of Sensors and Actuators*, Vol. 23, No. 5, pp. 696-700, 2010.
- [32] C.L. Hwang, H.C. Wu and M.L. Ling, "The Stepping over an Obstacle for the Humanoid Robot with the Consideration of Dynamic Balance," *SICE Annual Conference*, August 18-21. 2010, Taipei, Taiwan, pp. 2260-2268.
- [33] NAO Simulation Website: <http://www.cyberbotics.com/>